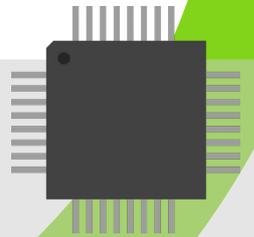
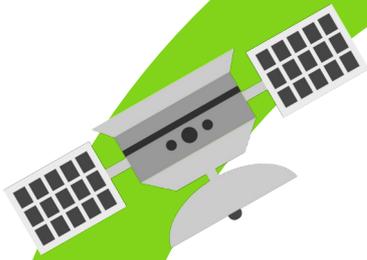


Congreso Argentino de Sistemas Embebidos



2022

18 y 19 de agosto



Libro de trabajos

AGENCIA
NACIONAL DE PROMOCION
CIENTIFICA Y TECNOLÓGICA



CONICET



UNIVERSIDAD
NACIONAL
DE LA PLATA



CASE 2022

Libro de Trabajos

Modalidades Artículo, Foro Tecnológico y Reporte

Congreso Argentino
de
Sistemas Embebidos

18 y 19 de agosto de 2022

CONICET



AGENCIA

NACIONAL DE PROMOCION
CIENTIFICA Y TECNOLOGICA



UNIVERSIDAD
NACIONAL
DE LA PLATA



ISBN 978-987-46297-9-1



9 789874 629791

Congreso Argentino de Sistemas Embebidos-CASE 2022 : libro de trabajos /
Diego Javier Brengi, Lutenberg, Ariel / Antonelli, Maximiliano/ Zacchigna, Federico G / Germino, Santiago /
Cebedio, Celeste / - 1a ed. - Ciudad Autónoma de Buenos Aires :
ACSE - Asociación Civil para la investigación, Promoción y Desarrollo de
Sistemas Eléctricos Embebidos, 2022.

Libro digital, PDF

Archivo Digital: descarga y online

ISBN 978-987-46297-9-1

1. Circuitos Electrónicos. 2. Hardware. 3. Software. I. Brengi, Diego Javier.

CDD 621.3907

Fecha de catalogación: 21/10/2022

Libro de Trabajos
Modalidades Artículo, Foro Tecnológico y Reporte
Congreso Argentino de Sistemas Embebidos - CASE 2022

Editores:

Zacchigna, Federico G.	FIUBA
De Micco, Luciana	UNMDP/ICyTE/CONICET
Brengi, Diego	INTI/UNLaM/FIUBA
Antonelli, Maximiliano	UNMDP/IcyTE/CONICET
Lutenberg, Ariel	FIUBA/CONICET
Cebedio, Celeste	UNMDP/IcyTE
Germino ,Santiago	FIUBA/CONICET

Diseño gráfico de tapas:

Diego Brengi

Copyright © 2022

Asociación civil para la investigación, promoción y desarrollo de los sistemas electrónicos embebidos.

Se otorga permiso para copiar y redistribuir este libro de trabajos, siempre que se mantengan los mensajes de copyright y la autoría de la obra y sus partes.

Prefacio

El diseño de sistemas embebidos es un motor clave de la industria y del desarrollo científico y tecnológico, y es un campo que en los últimos años ha crecido notablemente en la Argentina, tanto en la academia como en la industria.

El CASE (Congreso Argentino de Sistemas Embebidos) fomenta la presentación formal de trabajos usando el modelo de revisión de pares para garantizar la calidad en la presentación de los mismos. Este congreso forma parte del SASE (Simposio Argentino de Sistemas Embebidos), un evento anual que reúne a la comunidad académica y a la industria en torno a los sistemas embebidos, buscando fomentar esta temática.

El CASE 2022 se realizó el 18 y 19 de agosto, en el Centro de Postgrado Sergio Karakachoff de la Universidad Nacional de La Plata en modalidad presencial, transmitiendo además por streaming.

Los objetivos que persigue el congreso son:

- Ofrecer un lugar de encuentro para investigadores y becarios de todo el país, fomentando la colaboración.
- Difundir en el medio académico los adelantos científicos y tecnológicos producidos a nivel mundial.
- Propiciar la presentación y discusión de trabajos de investigación desarrollados en Argentina.
- Estimular en los estudiantes universitarios avanzados el interés por la investigación en el área de los sistemas embebidos.
- Coordinar y actualizar los contenidos de sistemas embebidos de los programas de grado y posgrado de las universidades argentinas.

Las áreas temáticas del CASE se organizan de la siguiente manera:

- Arquitectura de procesadores
- Bioingeniería
- Procesamiento Digital de Señales (Digital Signal Processing, DSP)
- FPGAs, HDLs y ASICs
- Implementación de Sistemas Embebidos
- Internet de las Cosas (Internet of Things, IoT)
- Linux Embebido
- Open Source Hardware
- Protocolos y Comunicaciones
- Robótica

- Sistemas Operativos en Tiempo Real (Real-Time Operating Systems, RTOS)
- Software Embebido
- Inteligencia Artificial

Dentro de cada una de estas áreas se permiten las modalidades Artículo, Foro Tecnológico y Reporte, según el tipo de trabajo.

Los trabajos presentados al CASE fueron sometidos a un proceso de revisión por pares doble ciego, corrección y *rebuttal*. De este modo fueron seleccionados 15 trabajos en la modalidad Artículo, 31 en modalidad Foro Tecnológico y 12 en la modalidad Reporte, de un total de 58 trabajos presentados.

Esta publicación se encuentra también disponible en forma *online* en la página web: www.sase.com.ar/case/

Esperamos que los trabajos recopilados en esta memoria sean de su interés y contamos con su participación en futuras ediciones del evento.

Atentamente,

Comité Organizador CASE

Entidad organizadora

- ACSE (Asociación Civil para la Investigación, Promoción y Desarrollo de los Sistemas Electrónicos Embebidos)
- UNLP (Universidad Nacional de La Plata)

Instituciones a la que pertenecen los organizadores

- CONICET (Consejo Nacional de Investigaciones Científicas y Técnicas)
- FIUBA (Facultad de Ingeniería Universidad de Buenos Aires)
- INTI (Instituto Nacional de Tecnología Industrial)
- UNLaM (Universidad Nacional de La Matanza)
- UNMDP (Universidad Nacional de Mar del Plata)
- UTN-FRH (Universidad Tecnológica Nacional - Facultad Regional Haedo)

Comité organizador estable

- Mg. Diego Brengi (INTI/UNLaM/FIUBA)
- Dr. Ariel Lutenberg (FIUBA/CONICET)
- Dra. Luciana De Micco (UNMDP/ICyTE/CONICET)
- Dr. Maximiliano Antonelli (UNMDP/ICyTE/CONICET)
- Ing. Federico G. Zacchigna (FIUBA)
- Mg. Santiago Germino - FIUBA/CONICET
- Ing. Celeste Cebedio - UNMDP

Comité organizador local

- Ing. Gerardo Sager (UNLP)

Chairs temáticos

- | | |
|--|-----------------------------------|
| ● Bioingeniería: | Ing. Juan Manuel Reta (UNER) |
| ● Comunicaciones y protocolos: | Ing. Ignacio Zaradnik (UNLaM) |
| ● FPGA, HDL y ASIC: | Ing. Salvador Tropea (INTI) |
| ● Implementación de
Sistemas Embebidos: | Dr. Julio Dondo (UNSL) |
| ● Inteligencia Artificial: | Ing. Yoel López (FIUBA) |
| ● IoT: | Ing. Lorenzo H. Tabares (CEADEN) |
| ● Open Source Hardware: | Mg. Ing. Sergio Burgos (UTN-FRP) |
| ● DSPs: | Ing. Alejandro Furfaro (UTN-FRBA) |
| ● Robótica: | Ing. Claudio Verrastro (CNEA) |
| ● Software Embebido: | Dr. Ricardo Cayssials (UNS) |

Revisores

Alanis, Arnulfo
Alessandrini, Gustavo
Alvarez, Nicolas
Amado, Martín Hugo
Antonelli, Maximiliano
Arizaga, Juan Antonio
Arnone, Leonardo
Barrangú, Juan Pablo
Bouchet, Agustina
Bouza, Magdalena
Brengi, Diego
Brizuela, Jose
Burgos, Enrique Sergio
Carbonetto, Sebastián
Carnaghi, Marco
Cayssials, Ricardo
Cazarez-Castro, Nohe Ramon
Cebedio, Maria Celeste
Comas, Diego Sebastián
Comas, Edgardo
De Micco, Luciana
Dondo, Julio
Filomena, Eduardo
Furfaro, Alejandro
Gallina, Sergio
Garcia Inza, Mariano
Gayoso, Carlos Arturo
Germino, Santiago
Gutiérrez, Marcelo
Heredia, Martín Alejandro
Hernandez Tabares, Lorenzo
Hidalgo, Roberto
Leiva, Lucas
Lopez, Yoel
Lopresti, Raúl Eduardo
Lozada Castillo, Norma
Lozano, Clevis
Lugo, Jorge
Lutenberg, Ariel
Luviano-Juarez, Alberto
Marchi, Edgardo
Monte, Gustavo
Oliva, Rafael
Pastafiglia, Daniel
Perez, Santiago
Perez-Paina, Gonzalo
Pulido-Luna, Jesus R.
Rabioglio, Lucas Andrés
Reta, Juan Manuel
Ridolfi, Pablo
Rodriguez-Sanchez,
M.Cristina
Sanca, Gabriel Andrés
Soria, Carlos
Tropea, Salvador
Uriz, Alejandro J.
Vaca, Gonzalo Nahuel
Verrastro, Claudio
Wassinger, Nicolas
Zacchigna, Federico G.
Zaradnik, Ignacio
Zecchin, Danilo

Índice de trabajos

Introducción	I
Índice de trabajos	VIII
Artículo	1
Bioingeniería	1
Preliminary Results of the Design and Prototyping of a Mass producible Low-cost Portable Mechanical Ventilator for Patients with respiratory failure , <i>Clevis Lozano, Jay Molino, Asdrual Rojas, Alfredo Lescher and Svetlana de Tristan</i>	2
Algoritmo para estimación de modelos pulmonares de pacientes bajo respiracion asistida , <i>Diego Alejandro Riva, Carolina Alejandra Evangelista and Paul Federico Puleston</i>	3
FPGAs, HDLs y ASICs	4
LabOSat: nine missions in a decade , <i>Gabriel Andrés Sanca, Mariano Barella, Fernando Gomez-Marlasca, Pablo Levy and Federico Golmar</i>	5
Open and Configurable Channelization for Future Software-Defined Radio , <i>Ignacio Milani, Guillermo Riva and Carlos Zerbinì</i>	6
Conteo visual automático de huevos de parásitos en rumiantes utilizando Xilinx DPU , <i>Lucas Leiva, Bruno Eduardo Nicolás Constanzo, Martín Vázquez and Juan Toloza</i>	7
Estrategia de modulación alternativa con fines de balance energético implementada en FPGA , <i>Yesenia Reyes Severiano, Jesus Aguayo Alquicira, Susana Estefany De León Aldaco and Luis Mauricio Carrillo Santos</i>	8
Diseño e implementación de generador de señal BPSK con parámetros regulables , <i>Mariano Morel, Raul Lopresti, Maximiliano Antonelli, Lucas Rabioglio and Luciana De Micco</i>	9
Implementación de Sistemas Embebidos	12
LabOSat-02: Development of an on-board computer for small satellites , <i>Leandro Gagliardi</i>	13
Inteligencia Artificial	14
Corn plant counting with point supervision , <i>Alfonso Rafel, Juan I. Cornet and Lucas C. Uzal</i>	15
Algoritmo de optimización por lobos grises para aplicaciones de filtrado adaptativo , <i>Guillermo Salinas, Eduardo Pichardo, Ángel Vázquez, Juan Avalos and Giovanny Sánchez</i>	18
Red neuronal aplicada a la predicción de caudales , <i>Rodolfo Frabotta and Ricardo Veiga</i>	19
Internet de las Cosas (Internet of Things, IoT)	22

Comunicación de datos para dispositivos IoT de bajos recursos: RS485 sobre la red eléctrica, <i>Guillermo R. Friedrich and Guillermo H. Reggiani</i>	23
Procesamiento Digital de Señales (Digital Signal Processing, DSP)	24
Image descriptors inspired in the human retina, <i>Pablo A. Salvadeo and Emmanuel Bello</i>	25
Protocolos y Comunicaciones	26
Codificación segura a nivel de hardware, <i>Raúl Eduardo Lopresti, Maximiliano Antonelli, Jorge Castiñeira Moreira and Luciana De Micco</i>	27
Software Embebido	28
Respiratory rate estimation on embedded system, <i>Isabel Morales, Leonardo Martinez Hornak, Alfredo Solari and Julián Oreggioni</i>	29
Foro Tecnológico	30
Bioingeniería	30
Evaluación de un algoritmo de compresión de imágenes médicas basado en la Transformada Wavelet y Redes Neuronales de Kohonen, <i>Daniel Max Saldarriaga and Ricardo Alfredo Veiga</i>	31
Implementación de placa adquisidora para el estudio de la técnica de Time Interleaving aleatorio, <i>Matias Medina, Raul Lopresti, Gustavo Zabaleta, Lucas Rabioglio and Luciana De Micco</i>	34
Sistema para el Registro Continuo de Variables Cardiovasculares durante Pruebas de Basculación, <i>Luis Antonio Noa Llorens, Yamel Jesús Almeida Pichardo, Angel Regueiro Gómez, Carmenchu Regueiro Busoch and Carmen Brígida Busoch Morlán</i>	37
FPGAs, HDLs y ASICs	40
Implementation of CIC filters for decimation of 532nm atmospheric LIDAR signals on FPGA., <i>Martín Alejandro Heredia, Facundo Larosa and Héctor Alberto Lacomi</i>	41
Aceleración de Simulación de Circuitos Cuánticos Parametrizados en SoC FPGA, <i>Agustin Silva, Claudio González and Omar Gustavo Zabaleta</i>	44
Stable type-2 fuzzy logic controller hardware implementation, <i>Yazmin Maldonado-Robles and Nohe Ramon Cazarez-Castro</i>	48
Adaptación de un stack de Aceleración por Hardware sobre FPGA, <i>Pedro Martos</i>	51
Migración de una plataforma de hardware no certificada hacia una certificada, <i>Luis David Diaz-Charris, Martín N. Menéndez, Santiago Germino, Ariel Lutenberg and Ramiro Ghignone</i>	54
Implementación de Sistemas Embebidos	57
Sistema de ensayo de relés ferroviarios de seguridad, <i>Gaspar Santamarina, Adrián Laiuppa, Gustavo Ramoscelli and Ariel Lutenberg</i>	58
Sistema de Comunicaciones Acústicas Subacuáticas aplicando SDUA, a partir de un SDR, <i>Pablo Martín, Maria Celeste Cebedio and David Petruzzi</i>	61
Optimización de la detección de rayos X para microdensitometría de muestras de madera utilizando un sensor de imagen CMOS COTS, <i>Damian Leonel Corzi, Jose Lipovetzky, Martín Pérez, Fabricio Alcalde Bessia, Guillermina Dalla-Salda, Anne Sophie Sergent, Alejandro Martínez Meier and Mariano Gómez Berisso</i>	64

Implementación de sistema embebido para adquisición de señales XBT , <i>Patricio Bos, Mariano Cinquini, Christian Galasso and Walter Correa</i>	67
Wearable device prototype for vital signs monitoring , <i>Leonardo Martinez Hornak, Isabel Morales, Alfredo Solari and Julián Oreggioni</i>	70
Detector de cajas calientes en material rodante , <i>Fabian Leonardo Sarmiento, Gustavo Ramoscelli and Ariel Lutenberg</i>	73
Diseño e implementación de data logger open hardware para estación meteorológica. , <i>Lázaro Andrés O’Farrill and Alex Manuel Rivera</i>	76
Sistema de Adquisición de Datos utilizando una CIAA-Safety , <i>Juan Pablo Rumie Vittar, Dario Walter Díaz, Gustavo Rodríguez, Juan Oviedo, Ariel Principi, Damian Primo, Diego Badino, Martin Escobar and Diego Salvador Fusari</i>	79
Herramientas de apoyo al estudio de sistemas embebidos utilizando uModelFactory , <i>Felipe Nirino and Nicolás T. Almaraz</i>	82
Implementación de una maqueta de sistema de control con aplicaciones didácticas , <i>Enrique Sergio Burgos, Francisco Alberto Sala and Carlos Alberto Cappelletti</i>	86
Inteligencia Artificial	89
People Behavior Tracking , <i>Hernán Contigiani and Pasquínell Urbani</i>	90
B-VGG16: Red neuronal de convolución binarizada para el reconocimiento de objetos , <i>Nicolás Urbano Pintos, Hector Alberto Lacomí and Mario Blas Lavorato</i>	93
Genre Classification with Deep Learning Techniques , <i>Marco Carnaghi and María Celeste Cebedio</i>	96
Espectrogramas de registros de Ballenas Barbadas, sintetizados a partir de Autoencoders. , <i>Marco Carnaghi and Maria Celeste Cebedio</i>	99
Deteccion de eventos y estimación de volumen de captura en buques pesqueros mediante visión artificial , <i>Nicolas Horro</i>	102
Computational Creativity through AI modeling , <i>Fernando Furundarena, Daniela López De Luise and Micaela Veiga</i>	105
Síntesis de espectrogramas de sonidos subacuáticos con Autoencoders y Transfer Learning , <i>María Celeste Cebedio and Marco Carnaghi</i>	109
Exploring design choices in Image Classification Convolutional Neural Networks , <i>Nicolas Cecchi and Javier Kreiner</i>	112
Internet de las Cosas (Internet of Things, IoT)	115
IOT device network for remote monitoring , <i>Hernan Rodriguez and Sebastian Marinsek</i>	116
Diseño de una plataforma para servicios IoT con aplicaciones en el sector productivo , <i>Jorge Osio, Facundo Chazarreta, Matias Busum Fradera, Juan Eduardo Salvatore and Daniel Martín Morales</i>	119
Procesamiento Digital de Señales (Digital Signal Processing, DSP)	122
Extended bank of observers for fault detection in hexarotors , <i>Claudio Pose, Leonardo Garberoglio, Ezequiel Pecker Marcosig, Ignacio Mas and Juan Giribet</i>	123
Algoritmo adaptativo NLMS/F basado en conjunto de membresías , <i>Laura Hidalgo, Ángel Vázquez, Xochitl Maya, Juan Avalos and Giovanni Sánchez</i>	127
Protocolos y Comunicaciones	130
Sistema celular alternativo y de emergencia basado en OpenBTS y SDR , <i>Alian Ernesto Matos Rodríguez and Marcelino Sánchez Posada</i>	131

Reporte	134
FPGAs, HDLs y ASICs	134
A Simple Low Voltage, Temperature Compensated MOSFET Dosimeter for embedded applications , <i>Pablo Antonio Petrashin, Walter Lancioni, Agustin Laprovitta, Carlos Dualibe and Juan Castagnola</i>	135
Implementación de técnicas SPWM en FPGA de código abierto , <i>Benjamín Chavarría Domínguez, Jesus Aguayo Alquicira and Susana De Leon Aldaco</i>	138
Assessment of image resolution reduction on position estimation in visual odometry , <i>Carlos Fernandez, Rodrigo Gonzalez and Daniel Patiño</i>	142
Implementación de Sistemas Embebidos	145
Sistema de información visual para pasajeros de trenes argentinos , <i>Carlos German Carreño Romano and Pablo Martín Gomez</i>	146
Diseño de un transmisor ADS-B embebido utilizando tecnología de Radio Definida por Software (SDR) , <i>Francisco Martín Escobar, Damian Hector Primo, Juan Octavio Oviedo, Juan Pablo Rumie Vittar, Dario Walter Diaz and Diego Daniel Badino</i>	150
Sistema embebido escalable para el monitoreo de parámetros en invernaderos mediante IoT , <i>Jorge Osio, Matias Mardegan, Mauro Salina and Marcelo Cappelletti</i>	153
Computación paralela en sistemas de radar mediante Computadoras de Placa Reducida , <i>Alian Ernesto Matos Rodríguez and Leandro Zambrano Méndez</i>	156
Internet de las Cosas (Internet of Things, IoT)	159
Diseño e implementación de una central operativa para el control y monitoreo de los sistemas de seguridad en el material rodante , <i>Fernando Iglesias, Matías Sambrizzi and Ariel Lutenberg</i>	160

Categoría

ARTÍCULO

Bioingeniería

Área Temática

Preliminary Results of the Design and Prototyping of a Mass producible Low-cost Portable Mechanical Ventilator for Patients with respiratory failure

Jay Molino¹ Senior IEEE, Ana Clevis Lozano^{1,2}, Asdrúal Rojas ¹, Alfredo Lescher¹, Svetlana de Tristán¹

Abstract—This paper shows the results of preliminary prototype's design validation of a pressure & volume control, portable, low cost and easy to replicate mechanical ventilator. This product has been designed for the modalities: (a) volume-controlled mandatory ventilation, (b) pressure-controlled mandatory ventilation. A PID controller was implemented on FPGA for its ease of parallel processing.

Index Terms— FPGA, ventilators, low-cost, portable, mechanical.

Algoritmo para estimación de modelos pulmonares de pacientes bajo respiración asistida

1st Diego A. Riva

*Instituto LEICI, Facultad de Ingeniería
UNLP-CONICET*

La Plata, Argentina
diego.riva@ing.unlp.edu.ar

2nd Carolina A. Evangelista

*Instituto LEICI, Facultad de Ingeniería
UNLP-CONICET*

La Plata, Argentina
cae@ing.unlp.edu.ar

3rd Paul F. Puleston

*Instituto LEICI, Facultad de Ingeniería
UNLP-CONICET)*

La Plata, Argentina
puleston@ing.unlp.edu.ar

Resumen—En este trabajo se presenta el algoritmo desarrollado para calcular el valor de los parámetros que determinan los modelos RC lineal y no lineal del sistema pulmonar en pacientes sedados y bajo ventilación mecánica. Además, se ha desarrollado un entorno de usuario que permite, por un lado, guardar los resultados obtenidos para un posterior procesamiento, y por el otro, mostrar al usuario la evolución de los parámetros.

Palabras Clave—Modelado pulmonar, Identificación de sistemas, Algoritmo de estimación.

Categoría

ARTÍCULO

FPGAs, HDLs y ASICs

Área Temática

LabOSat: nine missions in a decade

G. A. Sanca

*Escuela de Ciencia y Tecnología
Universidad Nacional de San Martín
San Martín, Argentina
gsanca@unsam.edu.ar*

M. Barella

*Department of Physics
University of Fribourg
Fribourg, Switzerland
mariano.barella@unifr.ch*

F. Gomez Marlasca

*Depto. Mat. Condensada
Comisión Nacional de Energía Atómica
Buenos Aires, Argentina
marlasca@cnea.gov.ar*

P. Levy

*CAC/CNEA - CONICET
ECyT/UNSAM
Buenos Aires, Argentina
levy@cnea.gov.ar*

F. Golmar

*ICIFI
CONICET-ECyT.UNSAM
San Martín, Argentina
fgolmar@unsam.edu.ar*

Abstract—LabOSat is an applied research group whose main objective is to increase the Technology Readiness Level of electronic devices and systems for their use in satellite missions. Starting in 2013, the Group has participated in 9 Low Earth Orbit missions in collaboration with Satellogic. Its main payload, LabOSat-01, is a Source-Measurement-Unit instrument dedicated to the electrical characterization of electronic devices. In this brief communication, results of in-orbit operation of LabOSat-01 are presented. A summary of the Group activities in the period 2013-2022 is also presented. These results represent almost 10 years of experience in 9 missions hosting different technologies.

Index Terms—Small satellite, satellite payloads, instrumentation, microelectronics

Open and Configurable Channelization for Future Software-Defined Radio

Ignacio Milani, Carlos Zerbini, Guillermo Riva

Grupo de Investigación y Transferencia en Electrónica Avanzada (GInTEA)

Universidad Tecnológica Nacional, Facultad Regional Córdoba (UTN-FRC)

Córdoba, Argentina

ignaciomilani@gmail.com, carloszerbini@gmail.com, guilriva@gmail.com

Abstract—Nowadays, signal demodulation chains used in Software-Defined Radio (SDR) receivers involve analog and digital sections. For economical reasons, the latter tends to gain ground over the analog one, and typically involves a channelization module based on Digital Down-Converters (DDCs). Given the current trend towards all-digital receivers based on direct RF sampling, the channelization architecture will become a critical issue, requiring open and flexible designs to cope with its challenges. As contribution, we evaluate two open alternatives for DDC's implementation on field-programmable gate arrays (FPGAs), one implemented in high-level language and the other one through hardware description language (HDL). Together, they provide flexibility for parameterization, as well as hardware-level optimization.

Index Terms—All-digital receiver, SDR, channelization, DDC, FPGA

Conteo visual automático de huevos de parásitos en rumiantes utilizando Xilinx DPU

Lucas Leiva

*LabSET - INTIA
UNICEN*

Tandil, Argentina
Facultad de Ingeniería
Universidad FASTA
Mar del Plata, Argentina
lleiva@exa.unicen.edu.ar

Bruno Constanzo

InFo-Lab

Universidad FASTA

Mar del Plata, Argentina
bconstanzo@ufasta.edu.ar

Martín Vázquez

LabSET - INTIA

UNICEN

Tandil, Argentina
*Facultad de Ingeniería
Universidad FASTA*
Mar del Plata, Argentina
mvazquez@labset.exa.unicen.edu.ar

Juan Toloza

*LabSET - INTIA
UNICEN*

Tandil, Argentina
UNTREF
Caseros, Argentina
jmtoloza@labset.exa.unicen.edu.ar

Resumen—El control parasitario es de vital importancia en la industria ganadera. El control de las infecciones se basa en el conteo de huevos de parásitos a partir de muestras obtenidas en el campo. Estas muestras son procesadas en laboratorios y el conteo es realizado generalmente de forma manual utilizando un microscopio óptico. En este trabajo se presenta el desarrollo de un algoritmo de conteo automático basado en Deep Learning sobre MPSoC utilizando Xilinx DPU. La propuesta posee la capacidad de implementar una solución portátil. Los resultados obtenidos disminuyen significativamente el tiempo de procesamiento, en comparación a otras soluciones existentes.

Palabras clave— *deep learning, DPU, conteo de huevos de parásitos*

Estrategia de modulación alternativa con fines de balance energético implementada en FPGA

Yesenia Reyes

Centro Nacional de Investigación
y Desarrollo Tecnológico
Tecnológico Nacional de México
Cuernavaca, México

yeseniareyes16e@cenidet.edu.mx

Jesus Aguayo

Centro Nacional de Investigación
y Desarrollo Tecnológico
Tecnológico Nacional de México
Cuernavaca, México

jesus.aa@cenidet.tecnm.mx

Susana De León

Centro Nacional de Investigación
y Desarrollo Tecnológico
Tecnológico Nacional de México
Cuernavaca, México

susana.da@cenidet.tecnm.mx

Luis Carrillo

Centro Nacional de Investigación
y Desarrollo Tecnológico
Tecnológico Nacional de México
Cuernavaca, México

luis carrillo16e@cenidet.edu.mx

Abstract— Actualmente, el estudio del balance energético en las celdas de los inversores multinivel en cascada es conveniente debido al creciente interés de implementar sistemas que minimicen las pérdidas, garanticen una entrega de la mayor cantidad de potencia a la carga, contribuyan a la confiabilidad del sistema y que al mismo tiempo minimicen el uso de recursos para su implementación. Una de las formas de llevar a cabo el balance energético sin recurrir a una topología adicional es mediante la modificación de técnicas de modulación convencionales. En este artículo se lleva a cabo la implementación en FPGA de estrategias de modulación con fines de balance energético. Primeramente, se abordan tres técnicas convencionales, las cuales son: disposición de portadoras con corrimiento de fase, desplazamiento de nivel por ciclo de portadora y desplazamiento de nivel por ciclo de moduladora. Posteriormente, se presenta la implementación de una estrategia de modulación alternativa que lleva a cabo balance energético en inversor multinivel en cascada, minimizando el uso de recursos digitales.

Palabras clave— *Inversor multinivel en cascada, balance de energía, técnica de modulación, FPGA, distorsión armónica total*

Diseño e implementación de generador de señal BPSK con parámetros regulables

Mariano Morel, Lucas Rabioglio, Raul Lopresti, M. Antonelli y Luciana De Micco
Instituto de Investigaciones Científicas y Tecnológicas en Electrónica (ICYTE)
Facultad de Ingeniería, Universidad Nacional de Mar del Plata (FI-UNMdP)
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
 Mar del Plata, Buenos Aires, Argentina
 mariano.morel@live.com.ar

Resumen—En este trabajo se diseñó e implementó un generador de señal modulada en fase. El circuito se implementó mediante una FPGA (Field Programmable Gate Array) y un conversor digital analógico (DAC). El sistema desarrollado permite recibir los datos a través de una entrada externa o los genera en forma interna mediante un generador de números pseudo aleatorios (PRNG). El diseño permite variar en un gran rango la cantidad de ciclos de portadora por dato y la frecuencia de portadora, así como la potencia de salida. El circuito utiliza una mínima cantidad de recursos de la FPGA gracias a la implementación de la portadora mediante el almacenamiento de un cuarto de ciclo y su correspondiente lógica de lectura.

Palabras clave—Modulación de fase, FPGA, DAC, BPSK, PRNG.

I. INTRODUCCIÓN

Este trabajo surge de un proyecto mayor en el que se pretende desarrollar sensores espectrales para Radio Cognitiva [1]. Estos sensores deben ser capaces de detectar la presencia de señal modulada en situaciones de relación de señal a ruido muy bajas. En este contexto, es necesario contar con un generador que entregue señales moduladas y permita realizar variaciones en cuanto a potencia, frecuencia de portadora, velocidad de datos, entre otras.

Existen trabajos en los que se proponen diseños en FPGA de moduladores BPSK (Binary Phase Shift Keying). En estos trabajos no se encontraron implementaciones con flexibilidad en los parámetros de la transmisión, tales como variación de frecuencia de portadora, cantidad de ciclos de portadora por dato, frecuencia de dato y potencia de salida en tiempo de ejecución. En [2] los autores realizan la simulación de la parte digital de un transmisor BPSK. Allí se almacenan en LUTs (Look Up Tables) 13 muestras de un período de una señal sinusoidal, y obtienen una frecuencia de portadora de 50 MHz. Proponen enviar la señal modulada digital en forma serie mediante Serial Peripheral Interface (SPI) al DAC embebido en la placa de desarrollo Spartan 3. Esto limita la máxima frecuencia de portadora.

En [3] los autores utilizan el entorno System Generator de Matlab de la empresa Xilinx para hacer el diseño de un modulador BPSK. La señal de portadora es implementada mediante un bloque Direct Digital Synthesis Compiler, almacenan en

Este trabajo fue parcialmente financiado por el Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) y la Universidad Nacional de Mar del Plata (FI-UNMdP), Argentina.

dos memorias ROMs (Read-Only Memory) dos ciclos de señal sinusoidal en contra-fase. La implementación la realizan en dos placas Spartan 3E Starter Kit. Obtienen una frecuencia de portadora 31,250 kHz.

En [4] el diseño es también realizado mediante el entorno System Generator. Para la implementación utilizan una FPGA Virtex-4 de Xilinx. Generan la portadora almacenando 20 muestras de un ciclo de sinusoidal en una memoria ROM que actúa como LUT, cada dato se representa con 16 bits en aritmética de punto fijo con signo (14 bits fraccionarios). La cantidad de ciclos por dato es fija en uno, con una portadora de 5 MHz.

Para lograr la flexibilidad requerida, en este trabajo se realizó un diseño que almacena en memoria una gran cantidad de muestras de portadora. Luego, a través de entradas, se seleccionan las muestras a ser leídas y entregadas a un DAC que las recibe en forma paralela. Este conversor se encarga de generar la señal analógica de salida. La implementación desarrollada utiliza una mínima cantidad de recursos gracias a la arquitectura mejorada que aprovecha el almacenamiento interno de un cuarto de señal sinusoidal para generar la señal de portadora.

II. CIRCUITO IMPLEMENTADO

El circuito desarrollado (cuyo código está en [5]) presenta dos bloques principales, uno se implementó en una placa que contiene un conversor digital analógico y un circuito de adecuación de señal, y el segundo se diseñó en la FPGA mediante un lenguaje de descripción de hardware VHDL (Very High-Speed Integrated Circuit Hardware Description Language). En la Fig. 1 se puede ver un esquema del sistema implementado. Se observan los dos bloques principales mencionados y las entradas de selección que permiten variar los parámetros de la modulación en tiempo de ejecución. Cada dato a transmitir tiene una duración T_b y el período de portadora es T_c por lo que se transmiten N_c períodos de portadora por dato ($T_b = N_c \cdot T_c$). La FPGA entrega al DAC las muestras de la señal modulada a una frecuencia f_s y siendo $4 \cdot N$ la cantidad de muestras de un ciclo de portadora, el período de portadora resulta $T_c = 4 \cdot N \cdot T_s$. De esta forma el tiempo de dato a transmitir $T_b = N_c \cdot 4 \cdot N \cdot T_s$, siendo $T_s = 1/f_s$.

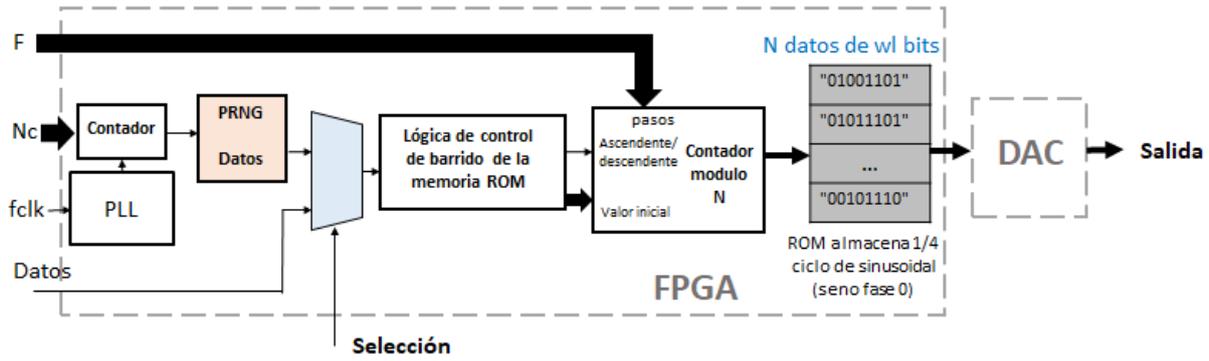


Fig. 1. Esquema del circuito implementado. La parte digital implementada en la FPGA y el circuito que convierte la señal digital a una señal analógica.

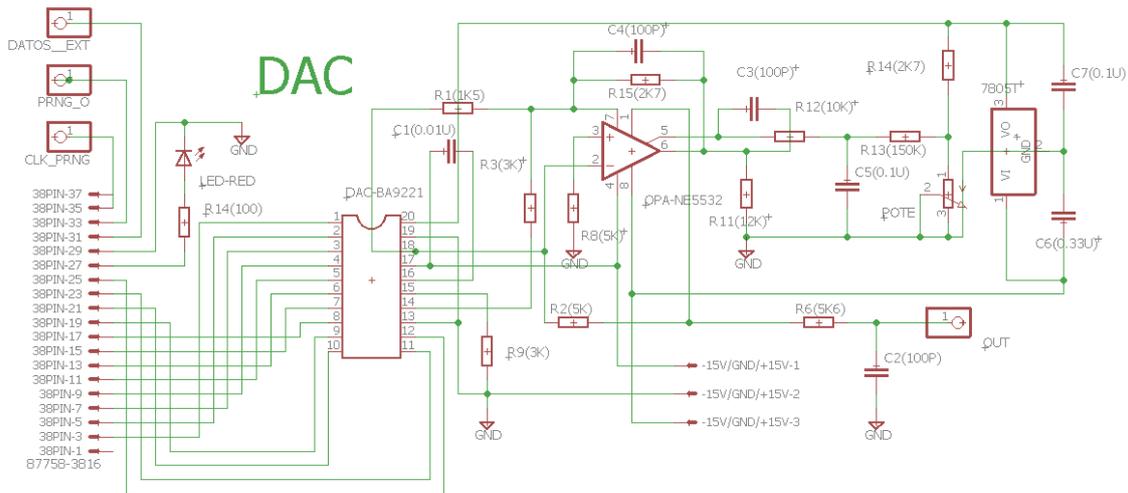


Fig. 2. Circuito esquemático de la placa convertora DAC.

A. Circuito DAC

Un esquemático del circuito implementado se muestra en la Fig. 2. Este circuito se encarga de recibir la señal digital de la FPGA y entregar la señal analógica modulada en BPSK. Incluye el circuito integrado DAC BA9221 [6], el mismo es un convertor digital-analógico que recibe 12 bits en formato de complemento a dos y entrega una corriente proporcional a una corriente de referencia. Este circuito permite variar la corriente de referencia mediante un potenciómetro (POTE en la figura) para obtener distintas amplitudes de la señal de salida, con un máximo de 4 mA.

Luego, esta corriente se convierte a una tensión de 10 Volts pico a pico. Finalmente, un filtro pasabajos se encarga de atenuar las altas frecuencias debidas al ruido de cuantificación.

B. Diseño en la FPGA

Un esquema del circuito implementado en la FPGA puede verse en la Fig. 1. El sistema consiste principalmente en la lectura de una memoria en la que se tiene almacenado un cuarto de ciclo de una señal senoidal. Esta memoria será leída y entregada al DAC según los parámetros de entrada (Nc y F de Fig. 1) y el dato a transmitir. Una lógica de control se encarga de establecer si el contador que recorre la memoria

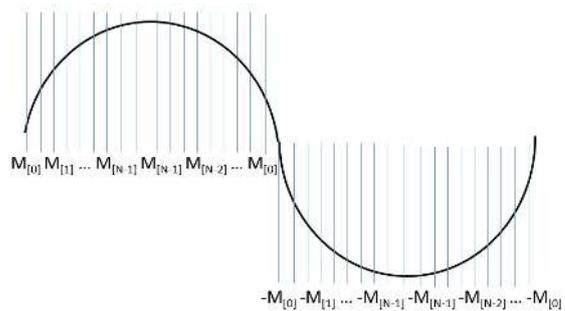


Fig. 3. Lectura de memoria de tamaño N para la generación de un ciclo de portadora.

realiza una cuenta ascendente o descendente, el valor inicial de esta cuenta y los pasos de incremento/decremento. También determina si el valor de la memoria debe ser negado o no. Por ejemplo, cuando se desee entregar un período senoidal a la mínima frecuencia de salida, se leerá la memoria de la siguiente forma:

- 1) Cuenta ascendente de 0 a N-1, en pasos de 1.
- 2) Cuenta descendente de N-1 a 0, en pasos de 1.
- 3) Cuenta ascendente de 0 a N-1, en pasos de 1 y salida

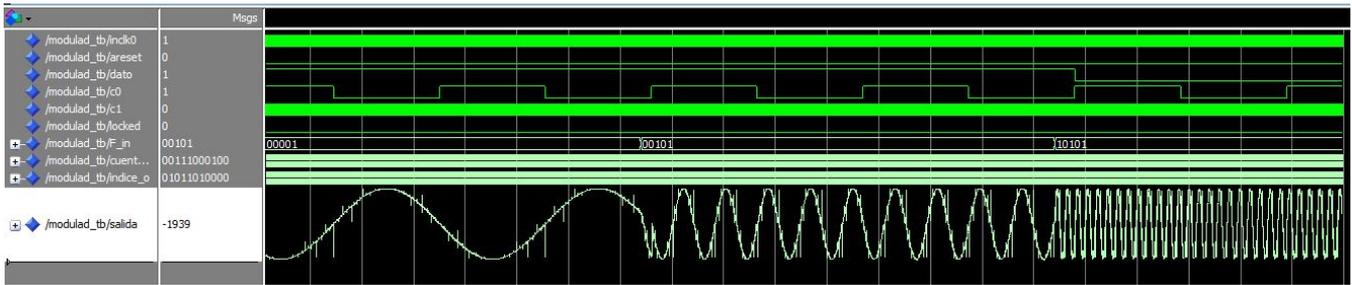


Fig. 4. Simulación mediante ModelSim de la salida ante la variación de la frecuencia de portadora.

negada.

- 4) Cuenta descendente de N-1 a 0, en pasos de 1 y salida negada.

Esto genera $4.N$ muestras de un ciclo de portadora, a la mínima frecuencia posible, que es $f_c = f_s / (4.N)$. A diferencia de trabajos existentes en los que usualmente se almacena un pequeño número de muestras por ciclo de portadora, como en [2] (13 muestras por ciclo) y [4] (20 muestras por ciclo), en esta implementación tomaron $N=256$ datos por cuarto de ciclo, o 1024 muestras por ciclo. Esta mayor cantidad de muestras permite una amplia variación de la frecuencia de portadora a la salida. En la Fig. 3 se ve cómo sería la lectura

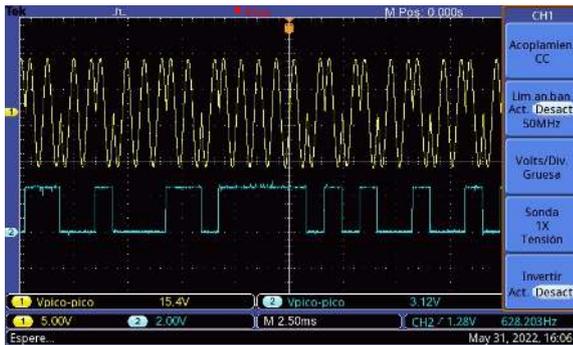


Fig. 5. Captura de la pantalla del osciloscopio en la medición de los datos moduladores (curva inferior en azul) y la señal modulada (curva superior en amarillo).

de la memoria de tamaño N para la generación de un ciclo de portadora completo. La entrada de Selección determina si los datos modulantes son generados internamente mediante un PRNG, que en este caso se trata de un LFSR (Linear Feedback Shift Register) o se reciben de forma externa.

En la Tabla I se muestran los recursos empleados en una FPGA Cyclone III de Altera para el circuito implementado. Se puede ver que, a pesar de almacenar un gran número de muestras de portadora (lo que permite variar los parámetros mencionados), consume muy pocos recursos gracias al diseño desarrollado.

El diseño propuesto admite una frecuencia máxima de operación (f_s) de 181,62 MHz lo que permitiría una máxima frecuencia de portadora de 90,81 MHz, y una mínima de 177 kHz. En la Fig. 4 se puede ver cómo varía la frecuencia de portadora ante el incremento de la entrada F (en la figura llamada F_{in}). En la Fig. 5 se puede ver la señal generada por

Elementos lógicos (EL) totales (119.088)	655
Combinacional de EL	518
Registros de EL	44
Registros dedicados(121.673)	137
PLL	1
f_{Max} [MHz]	181,62

TABLE I

RECURSOS EMPLEADOS EN LA IMPLEMENTACIÓN UTILIZANDO UNA FPGA CYCLONE III DE ALTERA.

el circuito, la señal de portadora (en amarillo) varía su fase según los datos de entrada (en celeste).

III. CONCLUSIONES

En este trabajo se diseñó e implementó un generador de señal modulada en fase. El diseño implementado permite variar los parámetros de la señal modulada de salida como la frecuencia de portadora, potencia de salida, etc. La cantidad de recursos empleados en la FPGA es mínimo gracias a la utilización de una memoria que almacena un cuarto de ciclo sinusoidal la cual es leída mediante una lógica de control para generar la señal de portadora. También se diseñó e implementó la placa que recibe la señal digital y entrega la señal analógica, la cual permite variar la potencia de salida de la misma.

AGRADECIMIENTOS

Este trabajo fue parcialmente financiado por CONICET, Agencia I+D+i (PICT19 3024) y la FI-UNMdP, Argentina.

REFERENCIAS

- [1] Y. Arjoune and N. Kaabouch, "A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions," *Sensors (Switzerland)*, vol. 19, no. 1, 2019.
- [2] A. K. Chaudhary, P. Pratik, S. Gupta, and V. Kakkar, "Improved digital design of bpsk modulator using look-up table technique," in *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2013, pp. 47–52.
- [3] S. Popescu, A. Gontean, and G. Budura, "Bpsk system on spartan 3e fpga," in *2012 IEEE 10th International Symposium on Applied Machine Intelligence and Informatics (SAMII)*. IEEE, 2012, pp. 301–306.
- [4] Y. Chye, M. Ain, and N. M. Zawawi, "Design of bpsk transmitter using fpga with dac," in *2009 IEEE 9th Malaysia International Conference on Communications (MICC)*. IEEE, 2009, pp. 451–456.
- [5] M. Morel, "GitHub," GitHub repository, 2022. [Online]. Available: <https://acortar.link/2RpRLF> (accedido el 1 de julio de 2022).
- [6] Ba9221. [Online]. Available: <http://www.synfpo.nl/datasheets/BA9221.pdf>

Categoría

ARTÍCULO

Implementación de Sistemas Embebidos

Área Temática

LabOSat-02: Development of an on-board computer for small satellites

L. Gagliardi
ICIFI
UNSAM, CONICET
Buenos Aires, Argentina
lgagliardi@unsam.edu.ar

F. Di Nardo
ECyT
UNSAM
Buenos Aires, Argentina
fededinardo2@gmail.com

G. A. Sanca
ECyT
UNSAM
Buenos Aires, Argentina
gsanca@unsam.edu.ar

F. Izraelevitch
ICIFI
UNSAM, CONICET
Buenos Aires, Argentina
fhi@unsam.edu.ar

M. Cveczilberg
ICIFI
UNSAM, CONICET
Buenos Aires, Argentina
mcvec@unsam.edu.ar

F. Golmar
ICIFI
UNSAM, CONICET
Buenos Aires, Argentina
fgolmar@unsam.edu.ar

Abstract—

LabOSat is an applied research group whose main goal is to increase the Technology Readiness Level of electronic devices and systems for their use in space. To achieve such objective, the group designs and fabricates satellite payloads. Between 2014 and 2020 LabOSat has participated in 9 missions. In recent years, LabOSat started the development of LabOSat-02, an on-board computer designed for small satellites. In this communication, LabOSat-02 hardware and firmware design is described and presented.

*Index Terms—*On-Board Computer, space mission, hardware, firmware, operation modes, small satellites

Categoría

ARTÍCULO

Inteligencia Artificial

Área Temática

Corn plant counting with point supervision

Alfonso Rafel^{†1}, Juan I. Cornet^{†2}, Lucas. C. Uzal^{†3}
[†]DeepAgro

¹arafel@deepagro.com.ar

²jicornet@deepagro.com.ar

³luzal@deepagro.com.ar

Abstract—Unprecedented progress in deep learning and computer vision has influenced many industries, in particular the agriculture. Among the challenges of this field, plant counting has an special importance, as it can provide insights about yield and growing capabilities. This document describes the implementation of a corn plant counting algorithm based on a weak type of annotation that only requires a pixel label by object instance, known as point supervision. Thanks to this method, the annotation time invested is optimized. The system aims at corn phenological stages between V1 and V4, considering several uncontrolled environmental conditions such as presence of weed, fallow land and plant overlapping. We show results over a set of images with diverse camera configurations, lightning conditions and plant densities. We found a successful performance with a model trained with only 99 annotated images. The application was motivated by a concrete need of the ag-tech company DeepAgro.

Index Terms—plant counting, instance segmentation, point supervision, convolutional neural networks

I. INTRODUCTION

Unprecedented progress in deep learning has influenced many industries, in particular the agriculture sector. Due to the increase of computed power, many solutions have become feasible and technological progress is slightly becoming a part of agriculture [1]. According to [2], over the next ten years, more global cereal production will originate from yield growth and area intensification given limits on available arable land.

Crop yield depends indirectly on plant population density, through the increase in the level of competition. In the case of corn, it is essential to optimize stand uniformity and that can be done by minimizing the plant-to-plant variation [3]. Stand count estimation demands extensive time and labor requirements, therefore the automation of the process has already been approached [4]. Thinking abstractly, we can reduce it to an object counting problem.

Object counting methods can be divided into three main categories: clustering, regression or detection [5]. Regression methods have defined state-of-the-art results in many benchmarks, outperforming detection methods [6]. The cause of this relies on the fact that regression loss functions are optimized for the object count, while the detection ones must also focus on the size, shape and location.

Besides the count, knowing the location is helpful to validate the precision of the predictions and understand the limitations of the solution. As it was described above, regression methods do not focus on the location, so this analysis could not be carried out.

Since crops vary by both environmental and genetic conditions, acquisition of huge amount of labeled image datasets is expensive and time-consuming. Thus, optimizing annotation labor is essential, for example through point supervision based active learning [7].

II. COUNTING ALGORITHM

Object counting is achieved by a weakly supervised semantic segmentation algorithm taken from [5]. They proposed a novel loss function that encourages the network to output a single blob per object instance using point level annotation only. From there, the counting problem is straightforward.

A. Model architecture

The architecture of the model can be seen in Figure 1. Basically it is a fully convolutional neural network (FCN) [8]: the first block correspond to the network backbone VGG16 [9], pretrained with ImageNet [10], which encourages feature extraction and the second one carries out an upsampling path to obtain blobs for the objects. The networks outputs a score for each pixel indicating the probability of the object class. Then a binary threshold is applied over the image, leaving clearly defined blobs that are separated and their amount is computed as the count (See [5] for more details).

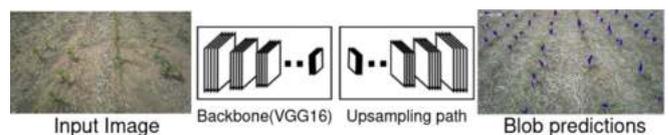


Figure 1: Model architecture.

B. Loss function

The localization-based counting loss (LC) consists of four different terms, that we qualitative describe here (see [5] for each term formulation):

- Image level loss: increases the probability that the model labels at least one pixel as object instance.
- Point level loss: encourages the model to correctly label the small set of supervised pixels contained in the ground-truth.
- Split level loss: discourages the model from predicting blobs that have two or more point-annotations. It uses a

split method to enforce blob splitting, each corresponding to a unique object.

- False positive level: discourages the model from predicting a blob with no point annotations, in order to reduce the number of false positive predictions.

C. Metrics

In order to measure the model counting performance, we use the mean absolute error (*MAE*), computed as:

$$MAE = \frac{1}{N} \sum_{i=1}^n |p_i - c_i| \quad (1)$$

where p_i is the predicted count and c_i the ground truth.

The precision of the predicted locations for the instances is evaluated through:

$$F_{Score} = \frac{2 * t_p}{2 * t_p + f_p + f_n} \quad (2)$$

where t_p (true positives) is the number of blobs that contain at least one point annotation, f_p (false positives) is the number of blobs that contain no point annotation and f_n (false negatives) is the number of point annotations minus the true positives.

This metric allows us to understand how the model is working, but as the main objective is to count plants, it is not used to measure its performance.

III. DESIGN AND IMPLEMENTATION

A. Data acquisition

Data was collected through a device known as *Explorer*[®], composed of a RGB camera, a GPS module and a processing unit. It is mounted on a self-propelled sprayer, one on each boom and takes pictures on regular intervals of time. The camera is installed with a pitch angle of 45°, but the height depends on the sprayer's use. This camera configuration adds image perspective as a challenge, but it allows the device to acquire high quality images during normal operation of the sprayer, between 15 km/h and 20 km/h. Real time counting will not be faced, as the concrete need of the ag-tech company embody an offline statistical analysis of field images.

B. Data annotation

As the annotation of a corn plant relies only in one pixel, the decision was to select the stalk or leaf collar. We designed a labeling tool capable of performing point level annotation: (1) The input image is divided into four sub-images. (2) Each sub-image is fed to the tool, which allows plant labeling with only one click. Then the tool will draw a red rectangle on the pixel selected to avoid mistakes (See left panel of Figure 3). (3) Finally we save the sub-image and a binary mask with zeros in the background and ones in the annotated pixels (points).

The process was carried out by one person, who required approximately 1.5 sec per annotation and 3 sec for checking. Considering that the maximum amount of plants in one sub-image is about 35, the total maximum time invested will be 55 sec. As it is shown in Table I, the total amount of labeled sub-images arises to 165, resulting in a maximum of 2.5h.

We selected three sets of images with different characteristics regarding acquisition date, phenological stage, plant density and environmental conditions. In Table I the most important features of the image sets are presented:

Table I: Data information

Feature	Set name		
	Set 1	Set 2	Set 3
acquisition date	10-30-2020	21-05-2021	01-11-2020
phenological stage	V3 - V4	V1 - V2	V2 - V3
resolution	1280x720	1920x1080	1640x922
images	861	2894	691
labeled images	28	4	4
sub-image size	640x360	960x540	960x540
labeled sub-images	113	16	16



(a) Set 1

(b) Set 2

(c) Set 3

Figure 2: Sample images from Sets

Data collection took place in an uncontrolled environment, during the spraying process. Notice that in Figure 2b, it is clearly shown the effect of sun and shadows of the sprayer. In Figure 2a, fallow land is present, as well as plant overlapping in the upper part.

C. Training and validation procedure

We splitted *Set 1* and *Set 2* labeled sub-images into training and validation sets, which remained fixed during all the process. In that way, the validation set was never used for training. The steps can be described as follows: (1) We fed the training sub-images and labels to the model without any pre processing. (2) We execute the forward process, we compute the loss, we run back-propagation and we update the model's weights. (3) After all the training set has been processed, we fed the validation set to the model and compute the metrics.

In Figure 3 we can see from left to right: the annotated instances, named as ground truth, the predicted blobs in blue and finally the heat-map, based on pixel probabilities. The pipeline was implemented using the framework *Pytorch* and mainly the library *OpenCV*.

D. Test procedure

For testing, we use the labeled sub-images from *Set 3*, totally unseen from the model. The process steps consisted on feeding the test set to the model and computing validation metrics.

IV. EXPERIMENTS

A. Simulation parameters

Labeled sub-images were randomly divided into two sets: a 99 sub-images training set and a 30 sub-images validation



Figure 3: Training process.

set. We use Adam optimizer [11] with a learning rate of 10^{-5} and weight decay of 5×10^{-5} . The experiment was run for 35 epochs, using a batch size equal to 1. For testing, we selected the best model from training, based on the smallest MAE on validation, and we used 16 sub-images.

B. Results

As we can see from Figure 4 a positive bias is present in the point distribution, meaning that the model tends to predict more instances than annotated. Nevertheless, the obtained MAE , shows a successful performance, not only in validation with a value of 2.0, but also in testing with a value of 2.7. We can compare the validation results with the ones obtained in [5] over the Penguins Dataset from [12], with a MAE of 3.74 and a F_{Score} of 0.61. Using the same model architecture we got a F_{Score} of 0.62.

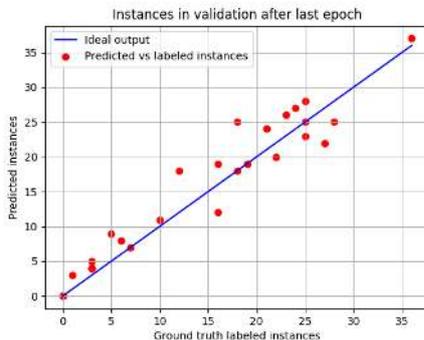


Figure 4: Validation results.

The value used for the binary threshold acts over the probability distribution generated by the model and determines the final size of the blobs. As the model has not the purpose of segment plants, it is enough for the blobs to be positioned where the plants are. Due to the long and thin shape of corn it could happen that, unless a blob is located over a labeled plant, it does not contain a ground truth point inside it. Consequently that situation does not fit as a true positive, decreasing the value of F_{Score} . In Figure 5, it is clearly shown how the blobs identify a correct instance of a corn plants, but as their size fluctuate, it could happen that they do not include the annotation point.

V. CONCLUSIONS AND FUTURE WORK

We found successful results over the corn plant counting problem, while we optimized the annotation time and the

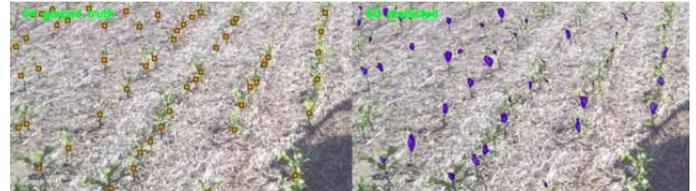


Figure 5: Sub-image example from test process.

necessary data. The main goal of the experiment was to analyze the capabilities of the system with images collected in uncontrolled conditions, using a reduced set of annotated images. We also discovered that it has an acceptable performance regarding plant location prediction. However, as limitations, we can say that is not prepared for instance segmentation and it is sensible to the labeling quality.

Next steps would be to train and validate the system progressively with more labeled images, analyze the effect of different values for the binary threshold after the output of the model and experiment with other parts of the plant as annotation points.

REFERENCES

- [1] H. Tian, T. Wang, Y. Liu, X. Qiao, and Y. Li, "Computer vision technology in agricultural automation—a review," *Information Processing in Agriculture*, vol. 7, pp. 1–19, mar 2020.
- [2] OECD-FAO, *OECD-FAO Agricultural Outlook 2022-2031*. OECD, jun 2022.
- [3] V. Fasoula and M. Tollenaar, "The impact of plant population density on crop yield and response to selection in maize," *Maydica*, 2005.
- [4] S. Khaki, H. Pham, Y. Han, W. Kent, and L. Wang, "High-Throughput Image-Based Plant Stand Count Estimation Using Convolutional Neural Networks," Oct. 2020.
- [5] I. H. Laradji, N. Rostamzadeh, P. O. Pinheiro, D. Vazquez, and M. Schmidt, "Where are the Blobs: Counting by Localization with Point Supervision," July 2018.
- [6] P. Chattopadhyay, R. Vedantam, R. R. Selvaraju, D. Batra, and D. Parikh, "Counting everyday objects in everyday scenes," Apr. 2016.
- [7] A. L. Chandra, S. V. Desai, V. N. Balasubramanian, S. Ninomiya, and W. Guo, "Active learning with point supervision for cost-effective panicle detection in cereal crops," *Plant Methods*, vol. 16, mar 2020.
- [8] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," Nov. 2014.
- [9] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Sept. 2014.
- [10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, jun 2009.
- [11] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Dec. 2014.
- [12] C. Arteta, V. Lempitsky, and A. Zisserman, "Counting in the Wild," in *Computer Vision – ECCV 2016*, pp. 483–498, Springer International Publishing, 2016.

Algoritmo de optimización por lobos grises para aplicaciones de filtrado adaptativo

Guillermo Salinas, Eduardo Pichardo, Ángel A. Vázquez, Juan G. Avalos, Giovanni Sánchez
SEPI ESIME Culhuacan, Instituto Politécnico Nacional
Ciudad de México, México

gsalinasr1500@alumno.ipn.mx, edua_95pim@hotmail.es, avazquezp1301@alumno.ipn.mx, javaloso@ipn.mx,gsanchezriv@ipn.mx

Resumen—Recientemente, se ha probado que los algoritmos metaheurísticos ofrecen buenos resultados en aplicaciones de filtrado adaptativo. No obstante, existen algoritmos que pueden caer en mínimos locales, un ejemplo, es el algoritmo de optimización por enjambre de partículas (PSO - Particle Swarm Optimization), cuya técnica de optimización se basa en una sola solución, lo cual provoca un alto nivel de desajuste y requiere una población muy grande para encontrar la solución óptima. Por otra parte, los algoritmos basados en soluciones por población, como el algoritmo de optimización por lobos grises (GWO - Grey Wolf Optimization), requieren una población más pequeña, poseen una gran capacidad de optimización global y requieren menos parámetros de control, facilitando así su implementación en dispositivos embebidos. En este trabajo, proponemos el uso del algoritmo GWO para las aplicaciones de: identificación de sistemas y cancelación de eco acústico. Los resultados demuestran que el algoritmo propuesto logra un error cuadrático medio (MSE - mean square error) más bajo en comparación al algoritmo PSO.

Palabras clave—algoritmo GWO, algoritmo PSO, filtrado adaptativo, identificación de sistemas, cancelación de eco acústico

Red neuronal aplicada a la predicción de caudales

Rodolfo Alberto Frabotta
 Facultad de Ingeniería
 Universidad de Buenos Aires
 Buenos Aires, Argentina
 rodolfofrabotta@yahoo.com.ar

Ricardo A. Veiga
 Instituto de Ingeniería Biomédica, Facultad de Ingeniería
 Universidad de Buenos Aires
 Buenos Aires, Argentina
 rveiga@fi.uba.ar ORCID: 0000-0003-1278-6007

Resumen—En este trabajo se propone un modelo de red neuronal recurrente para predecir escorrentías o caudales debidos a precipitaciones, y se lo compara con otro modelo de redes neuronales del tipo Long Short-Term Memory (LSTM). Se utilizan dos conjuntos de datos distintos: a) la base de datos Catchment Attributes and MEteorology for Large-Sample studies (CAMELS), referida a cuencas rurales de EE.UU. de América; b) varias cuencas sintetizadas a partir de un modelo extraído del programa Storm Water Management Model (SWMM). El modelo propuesto es más sencillo que el modelo de referencia, y en las pruebas realizadas logra aproximaciones del mismo orden.

Palabras clave—Redes neuronales, LSTM, NARX, lluvia, escorrentía, caudal.

I. INTRODUCCIÓN

El pronóstico de escorrentías en el terreno o caudales en cursos de agua debidos a precipitaciones, consiste en predecir una serie temporal de escorrentías o caudales a partir de series temporales de variables meteorológicas que incluyen las precipitaciones. Este pronóstico tiene diversas aplicaciones en la administración de recursos hídricos y la prevención y mitigación de daños por desastres naturales. En esta área de la Hidrología, el aprendizaje automático permite hacer predicciones con buena aproximación sobre la base de datos meteorológicos y características físicas del terreno; siendo especialmente útil en situaciones en que no es posible representar detalladamente los fenómenos que intervienen en la transformación de precipitación en escorrentía, ya sea por su complejidad o por ausencia de datos necesarios para otras clases de modelos.

La aplicación de modelos de redes neuronales a problemas hidrológicos es frecuente desde la década de 1990 [1], [2]. A lo largo del tiempo, fueron probadas distintas arquitecturas dinámicas, siendo las redes *feed-forward* las más empleadas al comienzo, sin conexiones de realimentación; con una mayor participación de arquitecturas más complejas en los últimos años [3]. Entre estas últimas se cuentan las redes neuronales recurrentes, que tienen memoria gracias a conexiones de realimentación, y no dependen sólo de la cantidad de pasos de tiempo considerados en las variables de entrada.

Los diferentes escenarios que se plantean en el pronóstico de caudales difieren en cuanto a las variables tenidas en cuenta como entradas del modelo. En general se considera la precipitación y a veces otras variables meteorológicas como entradas dinámicas; y en los casos en que se aplica un mismo modelo a varias cuencas, se usan como entradas constantes en el tiempo distintas características topográficas y del terreno. Los pronósticos que se realizan, son frecuentemente de un paso de tiempo; aunque también hay predicciones con varios pasos de tiempo de anticipación, como, por ejemplo, en [4], donde se presenta un modelo compuesto por una secuencia de

redes *feed-forward*, (SANN, *Sequential Artificial Neural Network*). La primera red de la secuencia usa como entrada, además de variables meteorológicas, el caudal real del paso de tiempo anterior; luego cada una de las siguientes redes toma como entrada el caudal estimado por la precedente corregido con un modelo de pronóstico del error.

Las redes recurrentes *Long Short-Term Memory* (LSTM) [5] han sido objeto de distintas variantes y modificaciones, siendo empleadas con buenos resultados [6]–[9], dado que estos modelos de redes neuronales realizan predicciones con mejor aproximación que otros comúnmente empleados en la práctica. En particular, en [8] se presenta la red *Mass-Conserving LSTM* (MC-LSTM), que es una variante de la LSTM, para aplicar al modelado de sistemas en los que se cumple un principio de conservación de una o más variables que se denominan variables de masa. En la aplicación a precipitación y escorrentía que se describe en [8], a lo largo del tiempo se mantiene un balance de la masa de agua que entra al sistema, sale de él o está contenida en el mismo. Los resultados muestran que MC-LSTM se desempeña mejor que todos los modelos con los que se lo compara en caudales extremos altos, por lo que se considera que representa el estado del arte en el pronóstico de caudales aplicado a predicción de inundaciones.

A continuación, se presenta el modelo propuesto, se explican los experimentos realizados y se muestran los resultados obtenidos. Finalmente se presentan las conclusiones de este trabajo.

II. MODELO PROPUESTO

En este trabajo se propone el empleo de un modelo de redes neuronales recurrentes *Non-linear Autoregressive with eXogenous input* (NARX) [10], [11], que es un diseño relativamente simple, con gran capacidad de aproximar funciones diversas [12], y que puede expresarse como:

$$\hat{y}_t = F(y_{t-1}, y_{t-2}, \dots, u_{t-1}, u_{t-2}, \dots) \quad (1)$$

donde u_t representa la entrada, y_t la salida e \hat{y}_t es la estimación de la salida en el instante t .

El objetivo es entrenar la red neuronal para modelar el comportamiento del sistema de tal forma que aprenda la función F . En particular, las redes NARX se caracterizan por realimentar la salida de la red hacia la entrada de la misma, sin conexiones de realimentación internas a la red. Esto permite el entrenamiento mediante propagación estática hacia atrás (*backpropagation*) reemplazando la realimentación por la salida real del sistema físico representado en el modelo. Dado que los valores de entrada y salida de pasos de tiempo futuros no están disponibles, se reemplazan por la salida estimada por el modelo NARX y las entradas por valores estimados por otro modelo externo. En el caso particular de predicción de un solo paso de tiempo, el modelo tiene como dato la salida real y puede funcionar como una red *feed-forward* (no recurrente).

III. EXPERIMENTOS Y RESULTADOS

El modelo NARX de este trabajo se implementó en lenguaje Python, usando las bibliotecas Keras y Tensorflow. Se indican a continuación los experimentos realizados y sus resultados junto con las métricas utilizadas.

A. Experimentos "CAMELS"

En la comparación se empleó una parte de los datos que fueron usados en [8], correspondientes a la base de datos CAMELS [13], [14]. Para reducir los tiempos de procesamiento, en lugar de emplear 28 años de datos hidrometeorológicos con frecuencia diaria (9 para entrenamiento, 9 para validación y 10 para prueba), para cada una de las 447 cuencas hídricas, se usaron 10 años (4 para entrenamiento, 2 para validación y 4 para prueba) en 6 cuencas elegidas al azar. Se mantuvo el esquema de predicción de [8] con datos de 365 pasos de tiempo (días) para predecir la escorrentía en el siguiente paso. Se usaron 5 variables meteorológicas y 26 atributos estáticos de las cuencas (constantes en el tiempo) como entradas. En el modelo MC-LSTM se usaron 64 celdas de estado, donde una de las entradas es "de masa", conservándose un balance con la salida y el estado de las celdas. Las otras 4 entradas son "auxiliares". En cambio, en el modelo NARX se usó como entrada la salida de 7 pasos anteriores. La red tiene una capa oculta de 100 nodos con activación logística y un nodo lineal en la capa de salida. En estos experimentos con datos CAMELS el modelo funcionó en forma no recurrente, empleando siempre valores reales de los pasos anteriores de la salida. MC-LSTM se ejecutó en un entorno Google Colab con GPU, con 30 épocas de entrenamiento en 12 minutos. NARX se ejecutó en una PC con CPU i7 y 8 GB de RAM, sin GPU, con 2397 épocas de entrenamiento en 38 minutos.

B. Experimentos "Example 9"

En otros experimentos, se generaron series de tiempo de escorrentías a partir de datos meteorológicos usando el programa Storm Water Management Model (SWMM), con el modelo presentado en [15], cap. 9, como *Example 9*. Las series son de 3 de las cuencas con frecuencia de 15 minutos, de 10 años en total, usándose 11, 4 y 8 meses para entrenamiento, validación y prueba, respectivamente. En este caso se predijeron múltiples pasos de tiempo, usándose una secuencia de 96 pasos de datos meteorológicos (1 día) para predecir la escorrentía en 4 pasos (1 hora). Se usaron 2 variables de entrada y 3 atributos estáticos. En el modelo MC-LSTM con 64 celdas, una variable de entrada es de masa y la otra es auxiliar. En el modelo NARX, se usan también como entradas 4 pasos anteriores de la salida (real o estimada según la antelación) y la red es de 64 nodos de activación logística en una capa oculta y un nodo lineal en la capa de salida. MC-LSTM se ejecutó en Google Colab con GPU, con 30 épocas de entrenamiento en 62 minutos; mientras que NARX en una PC i7 con 8 GB de RAM, sin GPU, con 1012 épocas de entrenamiento en 139 minutos.

C. Métricas de desempeño de los modelos

A continuación, se definen las métricas de desempeño utilizadas. Algunas de sus propiedades se describen en [6]–[8], [16] y [17]. En las siguientes ecuaciones, T es la cantidad total de pasos de tiempo, y es la variable a considerar (en este caso, caudal o escorrentía), Q representa al caudal de un curso de agua o escorrentía de una cuenca, y σ es la desviación estándar. Los valores estimados de y se indican como \hat{y} ,

mientras que los valores medios (medias aritméticas) se indican con \bar{y} .

- Eficiencia de Nash-Sutcliffe (*NSE*, *Nash-Sutcliffe Efficiency*) (valor ideal: 1):

$$NSE = 1 - \frac{\sum_{t=1}^T (\hat{y}_t - y_t)^2}{\sum_{t=1}^T (y_t - \bar{y})^2} \quad (2)$$

- Raíz del error cuadrático medio (*RMSE*, *Root Mean Squared Error*) (valor ideal: 0):

$$RMSE = ((1/T) \sum_{t=1}^T (\hat{y}_t - y_t)^2)^{1/2} \quad (3)$$

- Términos α y β de *NSE* (valores ideales: 1 y 0 respectivamente):

$$\alpha\text{-NSE} = \hat{\sigma} / \sigma \quad (4)$$

$$\beta\text{-NSE} = (\bar{\hat{y}} - \bar{y}) / \sigma \quad (5)$$

- Coeficiente de correlación de Pearson (r , *Pearson correlation coefficient*) (valor ideal: 1):

$$r = \frac{\sum_{t=1}^T (\hat{y}_t - \bar{\hat{y}})(y_t - \bar{y})}{(\sum_{t=1}^T (\hat{y}_t - \bar{\hat{y}})^2)^{1/2} (\sum_{t=1}^T (y_t - \bar{y})^2)^{1/2}} \quad (6)$$

- Eficiencia de Kling-Gupta (*KGE*, *Kling-Gupta Efficiency*) (valor ideal: 1):

$$KGE = 1 - ((s_r(r-1))^2 + (s_\alpha(\alpha-1))^2 + (s_\beta(\beta_{KGE}-1))^2)^{1/2} \quad (7)$$

siendo $\beta_{KGE} = \bar{\hat{y}} / \bar{y}$, y s_i los factores de ponderación; adoptándose $s_r = s_\alpha = s_\beta = 1$.

- Sesgo porcentual en los caudales pico de la *Flow Duration Curve* (*FDC*) (*FHV*, *Percent peak flow bias of the flow duration curve*) (valor ideal: 0):

$$FHV = 100\% \cdot \sum_{h=1}^H (\hat{Q}_h - Q_h) / \sum_{h=1}^H Q_h, \quad (8)$$

donde H es la cantidad de caudales reales por encima del correspondiente a una probabilidad de excedencia dada, (se adoptó 2%); y *FDC* es la curva de probabilidad de excedencia de caudales, que grafica la probabilidad de que ocurra un caudal mayor que un cierto valor o la proporción de tiempo durante un período de estudio en que ese valor fue excedido.

D. Resultados

En las siguientes tablas se muestran los promedios y desviaciones estándar de varias métricas de desempeño, calculadas en cada experimento con los resultados de cada tipo de modelo (MC-LSTM y NARX). Los promedios fueron calculados sobre todas las cuencas de cada experimento (6 y 3 cuencas, respectivamente).

TABLA 1. MÉTRICAS (CAMELS)

Métrica	MC-LSTM	NARX
<i>NSE</i>	0,645 ± 0,188	0,601 ± 0,182
<i>RMSE</i> (mm/día)	1,523 ± 0,630	1,517 ± 0,557
<i>KGE</i>	0,648 ± 0,195	0,702 ± 0,108
$\alpha\text{-NSE}$	0,751 ± 0,141	0,857 ± 0,142
$\beta\text{-NSE}$	-0,098 ± 0,137	-0,012 ± 0,069
r	0,822 ± 0,092	0,791 ± 0,101
<i>FHV</i>	-25,80 ± 11,55	-16,16 ± 15,69

En el experimento indicado en la Tabla 1 sobre los datos *CAMELS*, MC-LSTM tuvo mejor desempeño solamente en las métricas *NSE* y *r*. En cambio, el modelo NARX obtuvo un mejor promedio en todas las demás métricas.

En los experimentos sobre los datos del *Example 9*, las métricas fueron calculadas para pronósticos con distinta antelación. Se presentan resultados para antelaciones de 1 y 4 pasos de tiempo en las Tablas 2 y 3, respectivamente.

TABLA 2. MÉTRICAS (EXAMPLE 9, ANTELACIÓN 1 PASO)

Métrica	MC-LSTM	NARX
<i>NSE</i>	0,90066 ± 0,01311	0,96000 ± 0,01389
<i>RMSE (in/h)</i>	1,8150e-03 ± 0,5396e-03	1,1027e-03 ± 0,0650e-03
<i>KGE</i>	0,81844 ± 0,05811	0,90724 ± 0,01003
α - <i>NSE</i>	0,84503 ± 0,05296	1,00318 ± 0,03015
β - <i>NSE</i>	-0,00654 ± 0,00233	0,00689 ± 0,00067
<i>r</i>	0,95654 ± 0,00080	0,98052 ± 0,00631
<i>FHV</i>	-17,68 ± 7,22	2,81 ± 0,41

TABLA 3. MÉTRICAS (EXAMPLE 9, ANTELACIÓN 4 PASOS)

Métrica	MC-LSTM	NARX
<i>NSE</i>	0,90114 ± 0,01251	0,94405 ± 0,01394
<i>RMSE (in/h)</i>	1,8102e-03 ± 0,5339e-03	1,3221e-03 ± 0,1468e-03
<i>KGE</i>	0,81840 ± 0,05800	0,74617 ± 0,02431
α - <i>NSE</i>	0,84494 ± 0,05297	1,02204 ± 0,02957
β - <i>NSE</i>	-0,00655 ± 0,00231	0,01978 ± 0,00172
<i>r</i>	0,95684 ± 0,00040	0,97345 ± 0,00542
<i>FHV</i>	-17,62 ± 7,16	7,00 ± 0,53

En estos últimos experimentos, con predicciones con distinta antelación y datos sintéticos generados por un modelo completamente determinístico e invariante en el tiempo, ambos modelos realizaron pronósticos con métricas cercanas a los valores ideales. Las métricas del modelo MC-LSTM se mantuvieron relativamente constantes frente al incremento en los pasos de predicción. Aunque las métricas obtenidas con el modelo NARX presentan mayores variaciones, resultan mejores que las del modelo MC-LSTM en la gran mayoría de los indicadores; en particular en la métrica *FHV* que permitiría mejorar la predicción de inundaciones.

IV. CONCLUSIONES

En este trabajo se analizaron dos modelos que permiten pronosticar escorrentías o caudales originados en precipitaciones, uno basado en redes neuronales tipo NARX y el otro en redes neuronales MC-LSTM, que representa el estado del arte en el pronóstico de inundaciones. En los experimentos realizados, ambos modelos brindan una mejor estimación con datos sintéticos que con datos medidos, lo que es natural porque el modelo SWMM es en sí una aproximación. Los experimentos con datos sintéticos también mostraron una buena estimación incluso para distintas antelaciones en el pronóstico de caudales.

Por otra parte, se halló que el modelo NARX alcanza métricas del mismo orden y en la mayoría de los casos puede

tener mejor desempeño que el modelo MC-LSTM, para los datos e hiperparámetros empleados, por lo que podría ser de interés su aplicación, al ser un modelo más sencillo. Además, al tener un mejor desempeño en la métrica *FHV*, el modelo NARX permitiría una mejor predicción de inundaciones. Futuras modificaciones y pruebas podrían permitir establecer las condiciones requeridas para aplicarlo de manera más general.

REFERENCIAS

- [1] H. Maier, y G. Dandy, "Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications", *Environmental Modelling and Software*. 15. 101-124. 10.1016/S1364-8152(99)00007-9, 2000.
- [2] H. Maier, A. Jain, G. Dandy, y K. Sudheer, "Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions". *Environmental Modelling & Software*. 25. 891-909. 10.1016/j.envsoft.2010.02.003, 2010.
- [3] A. Mosavi, P. Ozturk, y K. Chau, "Flood prediction using machine learning models: literature review", *Water* 2018, 10(11), 1536; <https://doi.org/10.3390/w10111536>, 2018.
- [4] O. Prakash, K. Sudheer, y K. Srinivasan. "Improved higher lead time river flow forecasts using sequential neural network with error updating". *J. Hydrol. Hydromech.*, 62, 2014, 1, 60-74 DOI: 10.2478/johh-2014-0010. 62. 60-74. 10.2478/johh-2014-0010, 2014.
- [5] S. Hochreiter, y J. Schmidhuber, "Long short-term memory", *Neural computation*. 9. 1735-80. 10.1162/neco.1997.9.8.1735, 1997.
- [6] F. Kratzert, M. Herrnegger, D. Klotz, S. Hochreiter, y G. Klambauer, "NeuralHydrology – Interpreting LSTMs in Hydrology", 10.1007/978-3-030-28954-6_19, 2019.
- [7] F. Kratzert, D. Klotz, G. Shalev, G. Klambauer, S. Hochreiter, y G. Nearing, "Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets", *Hydrol. Earth Syst. Sci.*, 23, 5089-5110, <https://doi.org/10.5194/hess-23-5089-2019>, 2019.
- [8] P. Hoedt, F. Kratzert, D. Klotz, C. Halmich, M. Holzleitner, G. Nearing, S. Hochreiter, y G. Klambauer, "MC-LSTM: Mass-conserving LSTM", doi: 10.48550/arXiv.2101.05186, 2021.
- [9] M. Paredes Quiñones, M. Zortea, y L. Martins, "Fast-slow streamflow model using mass-conserving LSTM", doi: 10.48550/arXiv.2107.06057, 2021.
- [10] N. Nørgaard, "Neural network based system identification toolbox," Tech. Report. 00-E-891, Department of Automation, Technical University of Denmark, 2000.
- [11] M. Hagan, H. Demuth, M. Beale, O. De Jesús, "Neural network design," 2nd edition, eBook, ISBN-10: 0-9717321-1-6, ISBN-13: 978-0-9717321-1-7, 2014.
- [12] H. Siegelmann, B. Horne y C. Giles, "Computational capabilities of recurrent NARX neural networks," in *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 27, no. 2, pp. 208-215, April 1997, doi: 10.1109/3477.558801, 1997.
- [13] A. Newman, K. Sampson, M. Clark, A. Bock, R. Viger, y D. Blodgett, "A large-sample watershed-scale hydrometeorological dataset for the contiguous USA", Boulder, CO: UCAR/NCAR. <https://dx.doi.org/10.5065/D6MW2F4D>, 2014.
- [14] N. Addor, A. Newman, M. Mizukami, y M. Clark, "Catchment attributes for large-sample studies", Boulder, CO: UCAR/NCAR. <https://doi.org/10.5065/D6G73C3Q>, 2017.
- [15] J. Gironás, L. Roesner, y J. Davis, "Storm Water Management Model applications manual", U. S. Environmental Protection Agency, EPA/600/R-09/000 July 2009, 2009.
- [16] H. Gupta, H. Kling, K. Yilmaz, y G. Martinez-Baquero, "Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling", *Journal of Hydrology*, 377, 80-91, (2009).
- [17] K. Yilmaz, H. Gupta, y T. Wagener, "A process-based diagnostic approach to model evaluation: Application to the NWS distributed hydrologic model", *Water Resources Research - WATER RESOUR RES.* 44. 10.1029/2007WR006716, 2008.

Categoría

ARTÍCULO

Internet de las Cosas (Internet of Things, IoT)

Área Temática

Comunicación de datos para dispositivos IoT de bajos recursos: RS485 sobre la red eléctrica

Guillermo R. Friedrich

Dpto. Ingeniería Electrónica

Univ. Tecnológica Nacional, Facultad Regional Bahía Blanca

Bahía Blanca, Argentina

gfried@frbb.utn.edu.ar

Guillermo H. Reggiani

Dpto. Ingeniería Electrónica

Univ. Tecnológica Nacional, Facultad Regional Bahía Blanca

Bahía Blanca, Argentina

ghreggiani@frbb.utn.edu.ar

Resumen—Internet de las Cosas es un paradigma que ha venido teniendo un gran despliegue en diversas áreas, conectando dispositivos y objetos de uso cotidiano, desde el ámbito doméstico hasta la industria y la salud, entre otras. Por lo general los dispositivos IoT tienen limitada capacidad de memoria y procesamiento, y es conveniente que tengan un reducido consumo de energía. En tal sentido, este trabajo está orientado a tratar de aprovechar un estándar clásico de las comunicaciones por cable, RS485, para permitir la conexión de dispositivos sencillos a la red TCP/IP mediante un nodo intermediario. Así, el dispositivo IoT sólo requeriría de un puerto serial asincrónico y una interfaz a la red eléctrica. Se han realizado ensayos de transmisión de datos con una interfaz basada en el circuito integrado THVD8000, con resultados obtenidos han sido promisorios y alientan a continuar explorando este campo.

Keywords—IoT, RS485, Power Line Communication

Categoría

ARTÍCULO

Procesamiento Digital de Señales (Digital Signal
Processing, DSP)

Área Temática

Image descriptors inspired in the human retina

Pablo A. Salvadeo
Depto. de Ciencias Naturales
Universidad Nacional de Río Cuarto
Mendoza, Argentina
pablo.a.salvadeo@hotmail.com

Emmanuel D. Bello
Berlin, Alemania
emabello42@gmail.com

Abstract—In this paper, we consider two models of the human retina that describe the distribution of the midget retinal ganglion cells (mRGC), making modifications to the original sampling pattern of FREAK, a current bio-inspired binary descriptor extraction method. We conclude that more accurate, Watson's model, presents in average a better performance than the simple model, Exponential. Furthermore, new retinas achieve competitive results compared to classical algorithms and are better than FREAK. On the other hand, the more complex retinal patterns allow for the best performance, and Watson's model goes in this direction. Finally, irrespective of the model used, the best performance is obtained from a kernel size distribution that grows slowly.

Keywords—*bio-inspired sampling pattern, binary descriptors extraction, FREAK, human retina, midget retinal ganglion cells*

Categoría

ARTÍCULO

Protocolos y Comunicaciones

Área Temática

Codificación segura a nivel de hardware

Raúl Eduardo Lopresti, Maximiliano Antonelli, Jorge Castiñeira Moreira y Luciana De Micco
Instituto de Investigaciones Científicas y Tecnológicas en Electrónica (ICYTE)
Facultad de Ingeniería, Universidad Nacional de Mar del Plata (FI-UNMdP)
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)

Resumen—En este documento se analiza un esquema de Códigos Turbo (TC), diseñados en base a Codificadores Convolutivos. El esquema incluye polinomios de encriptación variables, cuyos coeficientes son seleccionados azarosamente cada L pasos, de un juego de polinomios óptimos. Este tipo de codificadores fue estudiado en trabajos previos, en donde se conjeturó cierta correspondencia entre la aleatoriedad de la secuencia codificada y el desempeño del TC. En este trabajo se propone, además de cambiar azarosamente los coeficientes, variar el tiempo que se mantienen. Esto es, la cantidad de pasos L , sobre los que se itera uno de los polinomios es variable y también es una variable aleatoria.

Se demuestra que es posible mejorar la seguridad de los datos transmitidos, sin comprometer la probabilidad binaria de error (P_{be}).

Palabras clave—Códigos turbo, cifrado, LSFR

Categoría

ARTÍCULO

Software Embebido

Área Temática

Respiratory rate estimation on embedded system

Isabel Morales, Leonardo Martínez Hornak, Alfredo Solari, and Julián Oreggioni,
Universidad de la República, Montevideo, Uruguay
Email: {imorales,leonardo.martinez.hornak,asolari,juliano}@fing.edu.uy

Abstract—We present the design, implementation, and results of an algorithm for respiratory rate estimation using respiratory-induced frequency, intensity, and amplitude variation calculated from the photoplethysmography (PPG) signal. The algorithm was developed in Python (on a PC) using synthesized signals and publicly respiration and PPG available data. Later, we ported it to an MSP432P401R microcontroller. Preliminary results are promissory and show that respiratory rate estimation can be performed on the selected platform. This work also includes a graphical user interface that runs on a PC to process data from sensors, configure alarms and display vital signs in real-time.

Index Terms—Respiratory rate estimation, photoplethysmography, signal processing, low-power embedded system

Categoría

FORO TECNOLÓGICO

Bioingeniería

Área Temática

Evaluación de un algoritmo de compresión de imágenes médicas basado en la Transformada Wavelet y Redes Neuronales de Kohonen

Daniel M. Saldarriaga
Escuela de Oficiales
Fuerza Aérea del Perú
Lima, Perú

dsaldarriaga@fap.mil.pe, d.saldarriaga@hotmail.com

Ricardo A. Veiga
Instituto de Ingeniería Biomédica, Facultad de Ingeniería
Universidad de Buenos Aires
Buenos Aires, Argentina
rveiga@fi.uba.ar ORCID: 0000-0003-1278-6007

Resumen—En este trabajo se evalúa un algoritmo de compresión de imágenes médicas (mamografías, resonancias magnéticas y tomografías computadas). El mismo está basado en la Transformada Wavelet Discreta y Redes Neuronales de Kohonen. El principal objetivo de este algoritmo es reducir la cantidad de datos a transmitir y almacenar, al tiempo que se busca mantener un nivel de calidad aceptable según su aplicación. Con el objeto de analizar la posible utilidad de este algoritmo para realizar análisis clínicos con las imágenes reconstruidas, se llevaron a cabo una serie de evaluaciones subjetivas con médicos especialistas y también con métricas objetivas. Los primeros resultados muestran que este algoritmo sería útil solamente para algunos tipos de imágenes médicas.

Palabras clave—Compresión de imágenes médicas, Transformada Wavelet, Redes Neuronales de Kohonen, Codificación Huffman.

I. INTRODUCCIÓN

Desde el año 2019, todo el planeta hace frente a una emergencia sanitaria causada por la pandemia coronavirus, la que se suma a otras enfermedades ya existentes. Esto provocó el retraso en el proceso de detección y diagnóstico a través de imágenes médicas digitales como las Tomografías Axiales Computarizadas (TAC), las Resonancias Magnéticas (RMI) y las Mamografías (MG), entre otras [1]. Tales imágenes, no solamente demandan una buena calidad para su uso, sino también una adecuada velocidad de transferencia de datos y capacidad de almacenamiento; de tal manera que no se generen retardos, ni pérdidas de información en los centros médicos que se encuentran con recursos limitados.

Así, se ha incrementado la búsqueda de nuevos métodos que optimicen la compresión de imágenes médicas, con la intención de procesarlas con sistemas embebidos. Una de las técnicas más utilizadas para la compresión de imágenes es JPEG (*Joint Photographic Experts Group*), basada en la Transformada Coseno; siendo luego mejorada por JPEG 2000, basada en el uso de la Transformada Wavelet [2] con cuantización escalar. Sin embargo, sigue siendo un tema abierto de investigación, por lo que recientemente se han propuesto mejoras que incluyen la cuantización vectorial. Por ejemplo, Rahali, Loukil y Bouhleb [3] propusieron un algoritmo basado en un esquema clásico de compresión de imágenes con pérdida de información: una primera etapa de descomposición o transformación que permite reducir la redundancia, una segunda etapa de cuantización vectorial para reducir la entropía y finalmente una de codificación de

entropía [4]. En particular usan la Transformada Wavelet Discreta (DWT, por sus siglas en inglés), luego una red de Kohonen, y finalmente la codificación de Huffman. En otros contextos, también se han utilizado algoritmos similares [5]. En este tipo de métodos de compresión, luego de la fase de descompresión y reconstrucción, las imágenes serán distintas a las originales, con algún grado de error que se busca minimizar [2].

Rahali, Loukil y Bouhleb [3] establecieron las bases teóricas de su algoritmo y lo estudiaron con una única imagen RMI y usaron solamente métricas objetivas, sin evaluar los resultados por médicos especialistas.

En este trabajo se presenta el procesamiento de tres tipos diferentes de imágenes médicas (MG, RMI y TAC), utilizando el algoritmo propuesto en [3], y con un mayor número de muestras. También se evalúan sus resultados a partir de indicadores objetivos y la opinión de especialistas en la interpretación de imágenes médicas, que son en definitiva las métricas más importantes desde el punto de vista funcional.

II. DESARROLLO

El algoritmo propuesto en [3] comprime las imágenes médicas realizando una primera etapa de compresión, usando la DWT, y luego efectúa una cuantización vectorial de las partes menos significativas del primer proceso, usando una red de Kohonen. Finalmente se utiliza la codificación de Huffman de los vectores representativos que permite efectuar una última etapa de compresión, y así transmitir esos datos al receptor. Cada etapa agrega un grado de compresión adicional y en el receptor se procede a realizar la decodificación y descompresión en un proceso inverso al del transmisor.

Para evaluar el desempeño de este algoritmo, se ha desarrollado una aplicación de software que realiza el procesamiento mencionado. El hardware utilizado ha sido una Laptop Acer Aspire E1-471-6806, con un procesador Intel Core i3 de 2da generación, Tarjeta de video Intel HD Graphics 3000, memoria RAM de 4GB, un disco duro de 500 GB, y el Sistema Operativo Windows 7 Ultimate.

Las imágenes de prueba se eligieron de una base de datos pública [6], donde cada imagen tiene 512x512 píxeles con 256 niveles de escala de grises. Dado que se deseaba realizar una evaluación por médicos especialistas, se seleccionaron 5 imágenes de cada tipo (MG, RMI y TAC), a ser procesadas por 4 familias de DWT (al igual que [3]: Coiflets, Haar, Daubechies y Symlets [7]), y cada una en 5 niveles de compresión diferentes, resultando un conjunto de 305

imágenes. A su vez, el procesamiento se realiza con un tamaño de bloque igual a 4 y con 1000 épocas de entrenamiento.

A continuación, se presentan los resultados obtenidos mediante los mismos indicadores de rendimiento y calidad objetiva utilizados en [3] y [8]: *Compression Ratio* (CR), *Mean Square Error* (MSE), *Peak Signal to Noise Ratio* (PSNR) y tiempo de compresión (TC). Adicionalmente, el foco se centra en la interpretación de los resultados desde el punto de vista de su utilidad por parte del personal médico especializado.

III. RESULTADOS

A. Rendimiento y calidad objetiva

A continuación, se presentan los valores mínimos y máximos logrados en los procesamientos efectuados sobre las imágenes.

TABLE I. INDICADORES OBJETIVOS

(*)			Haar	Symlets	Coiflets	Daubechie s
MG	CR	Mín.	54	54	0	54
		Máx.	89	89	87	89
	MSE	Mín.	27	30	24	28
		Máx.	586	586	1118	537
	PSNR	Mín.	20	20	18	21
		Máx.	33	33	34	34
TC	Mín.	00:30:38	00:10:02	00:25:00	00:30:03	
	Máx.	01:37:44	00:30:14	01:46:32	01:30:37	
RMI	CR	Mín.	86	86	71	86
		Máx.	95	95	94	95
	MSE	Mín.	2	4	4	4
		Máx.	293	285	272	281
	PSNR	Mín.	23	24	24	24
		Máx.	45	42	29	42
TC	Mín.	00:57:16	00:59:20	00:58:46	00:39:41	
	Máx.	02:11:37	01:30:46	02:13:39	01:52:46	
TAC	CR	Mín.	86	86	71	86
		Máx.	92	92	91	92
	MSE	Mín.	4	5	3	4
		Máx.	73	52	88	45
	PSNR	Mín.	30	31	29	32
		Máx.	42	41	43	42
TC	Mín.	00:23:47	00:23:56	00:27:41	00:21:49	
	Máx.	00:36:47	00:44:33	00:56:32	00:44:12	

(*) *Compression Ratio* (CR) en veces, *Mean Square Error* (MSE) en intensidad de pixel al cuadrado, *Peak Signal to Noise Ratio* (PSNR) en dB y tiempo de compresión (TC) en horas, minutos y segundos.

En la Tabla I puede observarse que las CR de las RMI y TAC suelen ser mejores que las CR de las MG. Los valores del MSE generalmente disminuyen en el siguiente orden: MG, RMI y TAC; mientras que los valores de PSNR se incrementan en ese mismo orden. Además, los TC de las TAC suelen ser menores, comparadas con las MG y RMI.

B. Experimentos visuales subjetivos

Para estos experimentos se ha requerido la colaboración de tres médicos expertos (uno especialista en Imagenología y dos en Radiología). Cada uno de ellos realizó tres experimentos visuales de cada imagen médica descomprimida, a partir de

las compresiones del Nivel 1 con cada familia; que en general eran las de mejor calidad.

Los experimentos fueron diseñados para poder detectar y eventualmente corregir posibles sesgos involuntarios de los observadores médicos. Los tres experimentos tratan de medir aspectos que se superponen en cierta medida y que permiten analizar la coherencia de los resultados.

1) Elegir la imagen más parecida a la original

Este primer experimento consistió en presentarle cinco imágenes a cada experto: la original y las reconstruidas con cada una de las cuatro familias Wavelet. Luego se les pidió que indicaran cuál era la que tenía mayor grado de similitud con la original.

Se puede observar en la Tabla II que la familia Wavelet Daubechies ha sido la más seleccionada. Además, hay mayor consenso en que dicha familia podría ser más adecuada para procesar las imágenes médicas MG y TAC que las RMI.

TABLE II. ELECCIÓN DE LA MG, RMI Y TAC MÁS PARECIDA A LA ORIGINAL

Tipo de imagen		MG			RMI			TAC		
Médico		A	B	C	A	B	C	A	B	C
Coiflets	N° de imagen	1			√		√			
		2			√	√	√			
		3								
		4				√	√	√		
		5								
Daubechies	N° de imagen	1	√	√	√		√		√	√
		2	√	√	√				√	√
		3	√		√	√	√		√	√
		4	√	√	√				√	√
		5	√	√				√	√	√
Haar	N° de imagen	1								
		2								
		3		√				√		
		4								
		5			√	√	√			
Symlet	N° de imagen	1								
		2								
		3								
		4								
		5								

2) Mean Opinion Score (MOS)

En este experimento, los médicos debían calificar las imágenes ya mencionadas (5 excelente, 4 bueno, 3 regular, 2 pobre, 1 malo), promediándose el resultado.

En la Tabla III se observa que las imágenes TAC fueron identificadas por los médicos como las de mejor calidad de reconstrucción (entre bueno y excelente). Por otra parte, puede observarse que la familia Wavelet Daubechies cuenta con el mayor indicador tanto para las MG como las TAC. A su vez, las RMI presentan puntuaciones generalmente muy bajas (entre malo y regular).

3) Grado de utilidad para el diagnóstico médico

En este experimento, los médicos debían ordenar las imágenes de mayor a menor según la utilidad clínica, indicando con 4 la de mayor utilidad y con 1 la de menor utilidad.

TABLE III. MOS DE LAS MG, RMI Y TAC

	MG		RMI		TAC	
	Nº de imagen	MOS	Nº de imagen	MOS	Nº de imagen	MOS
Coiflets	1	1,0	1	2,3	1	4,0
	2	1,7	2	1,3	2	4,7
	3	1,0	3	1,3	3	4,0
	4	2,0	4	2,7	4	4,7
	5	2,7	5	2,7	5	4,3
Daubechies	1	4,7	1	2,0	1	5,0
	2	4,3	2	2,7	2	5,0
	3	4,7	3	2,0	3	5,0
	4	5,0	4	3,0	4	5,0
	5	5,0	5	3,0	5	5,0
Haar	1	3,3	1	1,0	1	5,0
	2	3,0	2	2,7	2	5,0
	3	4,0	3	2,0	3	5,0
	4	4,7	4	2,7	4	5,0
	5	4,3	5	3,0	5	5,0
Symlets	1	3,0	1	1,0	1	4,3
	2	3,0	2	2,3	2	5,0
	3	3,7	3	2,0	3	4,3
	4	3,7	4	2,3	4	4,7
	5	3,7	5	3,0	5	4,0

TABLE IV. GRADO DE UTILIDAD DE LAS IMÁGENES RECONSTRUIDAS

Imagen		MG			RMI			TAC			
Médico		A	B	C	A	B	C	A	B	C	
Coiflets	Nº de imagen	1	1	2	1	4	3	3	1	2	1
		2	1	1	1	4	4	4	1	1	1
		3	1	2	1	1	1	1	1	1	2
		4	1	1	1	4	4	4	2	2	1
		5	2	1	2	1	1	1	2	1	2
Daubechies	Nº de imagen	1	4	4	4	3	4	4	4	4	4
		2	4	4	4	2	2	2	4	4	4
		3	4	4	4	4	4	4	4	4	4
		4	4	4	4	3	3	3	4	4	4
		5	4	4	4	3	3	3	4	4	4
Haar	Nº de imagen	1	3	3	3	1	2	2	3	3	2
		2	3	3	3	1	1	1	2	3	3
		3	3	3	3	3	3	3	3	3	3
		4	3	3	3	2	1	1	3	3	3
		5	3	3	3	4	4	4	3	3	3
Symlet	Nº de imagen	1	2	1	2	2	1	1	2	1	2
		2	2	2	2	3	3	3	3	2	3
		3	2	1	2	2	2	2	2	2	2
		4	2	2	2	1	2	2	1	1	1
		5	1	2	1	2	2	2	1	2	1

Conforme al orden de utilidad que indicaron los médicos en la Tabla IV, la familia Wavelet Daubechies ha sido la más adecuada. También se puede observar que no habría un consenso en la utilidad de alguna familia Wavelet para el tratamiento de imágenes RMI; justamente los experimentos subjetivos que no fueron realizados en [3].

IV. CONCLUSIONES

Si bien la cantidad de imágenes de cada tipo, así como la cantidad de evaluadores subjetivos con los que se pudo contar,

fueron relativamente pocos para ser considerados estadísticamente, esta primera aproximación puede ayudar a evaluar preliminarmente la posible utilidad de estas técnicas para procesar las imágenes médicas mencionadas.

En particular, determinados consensos en las evaluaciones subjetivas ayudan a validar o descartar ciertas técnicas.

Los resultados mostrados en [3], solamente indican los obtenidos con métricas objetivas de una sola imagen RMI. En cambio, en el presente trabajo se muestra la aplicación a 3 tipos de imágenes diferentes (RMI, TAC y MG), con 5 imágenes por cada tipo. Además, se analizaron resultados tanto objetivos como subjetivos, siendo estos últimos los más importantes a la hora de evaluar su utilidad práctica.

Las tasas y los tiempos de compresión, así como los indicadores de calidad, dependen de cada imagen analizada y de los niveles de descomposición de la DWT. Sin embargo, hay cierta correspondencia con los experimentos subjetivos indicados en las Tablas II y IV, especialmente.

Entre los experimentos subjetivos se detectó un posible sesgo por parte de los médicos, por ejemplo, al evaluar las tomografías computadas procesadas con la familia Haar en el experimento de *Mean Opinion Score*, dado que las califican como “excelente”, cuando en los otros dos experimentos opinan en forma relativamente diferente. Por lo tanto, ese resultado podría considerarse como inválido para dicho experimento.

De las pruebas subjetivas realizadas, se puede concluir que la familia Daubechies podría ser útil para posibles diagnósticos médicos, al ser la más adecuada para procesar las mamografías y tomografías computadas. En cambio, para las resonancias magnéticas, que fueran las analizadas en [3], no parecen resultar adecuadas ninguna de estas técnicas.

REFERENCIAS

- [1] A. Sarhan, “Run length encoding based wavelet features for COVID-19 detection in X-rays,” *BJR Open*, vol. 3, 2021.
- [2] A. J. Hussain, A. Al-Fayadh, and N. Radi, “Image compression techniques: A survey in lossless and lossy algorithms,” *Neurocomputing*, vol. 300, pp. 44–69, 2018.
- [3] M. Rahali, H. Loukil and M. Bouhleb, "Theoretical improvement of the image compression method based on wavelet transform," in *2016 13th International Conference on Computer Graphics, Imaging and Visualization (CGiV)*. IEEE., pp. 273-277, 2016.
- [4] P. S. Narayana and A. M. Khan, “MRI image compression using multiple wavelets at different levels of discrete wavelets transform,” in *Journal of Physics Conference Series, IOP Publishing*, vol. 1427, no. 1, pp. 012002, 2020.
- [5] J. Anvy, A. Damayanti and A. Pratiwi, "Clustering of fish freshness using discrete wavelet transform and Kohonen self organizing map," in *Journal of Physics Conference Series, IOP Publishing*, vol. 1494, pp. 012008, 2020.
- [6] MedPix. (12 de 03 de 2018). MedPix. Obtenido de <https://medpix.nlm.nih.gov/home>.
- [7] P. Singh, P. Singh, and R. Kumar Sharma, “JPEG Image Compression based on Biorthogonal, Coiflets and Daubechies Wavelet Families,” *International Journal of Computer Applications*, vol. 13, pp. 1–7, 2011.
- [8] S. J. Pinto and J. P. Gawande, "Performance analysis of medical image compression techniques," *2012 3rd Asian Himalayas International Conference on Internet*, pp. 1-4.

Implementación de placa adquisidora para el estudio de la técnica de Time Interleaving aleatorio

Matias Medina, Raul Lopresti, Gustavo Zabaleta, Lucas Rabioglio y Luciana De Micco
Instituto de Investigaciones Científicas y Tecnológicas en Electrónica (ICYTE)
Facultad de Ingeniería, Universidad Nacional de Mar del Plata (FI-UNMdP)
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
 Mar del Plata, Buenos Aires, Argentina

Resumen—En este trabajo se analiza el efecto de la selección aleatoria de conversores analógico digitales (ADCs, Analog to Digital Converters) en sistemas de adquisición Time Interleaving (TI) aleatorios. Más específicamente el efecto de la longitud del período de la secuencia aleatoria utilizada sobre la performance final del sistema. Para ello, se implementó un sistema adquisidor de señal cuyo control se desarrolló en una FPGA (Field Programmable Gate Array) para permitir el testeado de distintos generadores pseudo aleatorios.

Palabras clave—Muestreo, Time Interleaving aleatorio, conversor analógico digital, ADC, LFSR.

I. INTRODUCCIÓN

La demanda de velocidades cada vez mayores para la adquisición de señales en diversos ámbitos de la electrónica, como por ejemplo en los sistemas de comunicación óptica para aplicaciones como 5G, requiere realizar conversiones analógica-digitales a una velocidad muy alta. Esto implica además, que todo el sistema de adquisición opere a frecuencias altas, el almacenado de datos en memoria, la transmisión, etc. La técnica Time Interleaving permite disminuir la velocidad de operación del sistema mediante la utilización en paralelo de ADCs idénticos. Estos ADCs reciben señales de reloj desfasadas, lo que permite la adquisición de señales a una frecuencia mayor a la de cada conversor individual. Más precisamente, la frecuencia de muestreo se incrementa la misma cantidad de veces que el número de conversores empleados. El uso de esta técnica permite además, que todos los componentes del sistema operen a frecuencias menores, como las memorias, ya que los datos se almacenarán a la frecuencia de cada conversor. Sin embargo, este método presenta el problema de introducir distorsión en la señal muestreada debido a que las conversiones realizadas por los ADCs no son exactamente iguales. Se presentan tres tipos de errores de desajuste [1]:

- 1) Error temporal (jitter): causado por los distintos retardos que tiene la señal de reloj que alimenta cada ADC. Esto implica que la señal será muestreada periódicamente pero no de manera uniforme.
- 2) Error de offset de amplitud: causado por pequeñas diferencias en el nivel de GND de los convertidores. Esto implica que habrá un desplazamiento constante de amplitud en la conversión de cada ADC.
- 3) Error de ganancia: causado por posibles diferencias de ganancia entre los convertidores.

Muchos trabajos sugieren técnicas para estimar y corregir estos errores [2], [3]. Estas técnicas proponen realizar calibraciones

iniciales y también dinámicas, en ambos casos se complejiza en gran medida el circuito por lo que se buscan otras opciones. Otra solución es aleatorizar la selección del ADC que debe usarse en cada instante de tiempo. Esto significa que un ADC se elige al azar en cada instante de muestreo. Esta técnica es conocida como TI aleatorio [4].

En este trabajo se diseñó e implementó en FPGA un sistema de adquisición que permite estudiar dicha técnica. Para ello, primero se analizó el período mínimo necesario en el PRNG (Pseudo Random Number Generator), utilizado en la selección aleatoria, para conseguir una disminución adecuada en la potencia interferente. Ésta fue medida mediante el cálculo del Rango Dinámico Libre de Espurios (SFDR). Además, el diseño incluye un novedoso método propuesto para la generación aleatoria de la señal de reloj de cada ADC mediante la secuencia aleatoria.

II. DISEÑO DEL CIRCUITO

Se realizaron simulaciones para analizar las características del PRNG que genera la secuencia de elección de los ADCs [5]. Como PRNG se utilizaron registros de desplazamiento con retroalimentación lineal (LFSR) de distintos períodos y se buscó una relación entre el período del PRNG y la atenuación de las armónicas interferentes. En las simulaciones se utilizó una base de tiempo de $T = \frac{1}{1000}$ s, esto determina una grilla fina de tiempo, por lo que el error de jitter deberá ser un múltiplo de este tiempo. Esta es una limitación inherente a las simulaciones. Para la señal analógica de entrada se adoptó una señal sinusoidal con período $T_c = 20.000.T = 20$ s. ($f_c = 0,05$ Hz). Se realizaron simulaciones considerando distintas frecuencias de muestreo. En la Fig. 1 se muestra, en negro, la señal sinusoidal muestreada con un solo ADC y, en naranja, la señal muestreada secuencialmente con 4 ADCs mediante la técnica de TI. Se puede ver en la Fig. 1.a que cada ADC presenta un error distinto de fase, ganancia y amplitud, lo que produce que la señal digitalizada tenga una apariencia ruidosa, con un ruido que se repite en forma periódica. La Fig. 1.b muestra los espectros de frecuencia de cada una de estas señales digitalizadas, se observa que la señal muestreada con TI (naranja) presenta componentes espectrales no deseados de mayor amplitud, producidos por la distorsión de los ADCs. Se ve claramente en la Fig. 1.b como disminuye el SFDR al emplearse la técnica de TI con la selección secuencial de ADCs. La Fig. 2 muestra la mejora al emplear TI aleatorio para distintos períodos del PRNG utilizado en la selección del

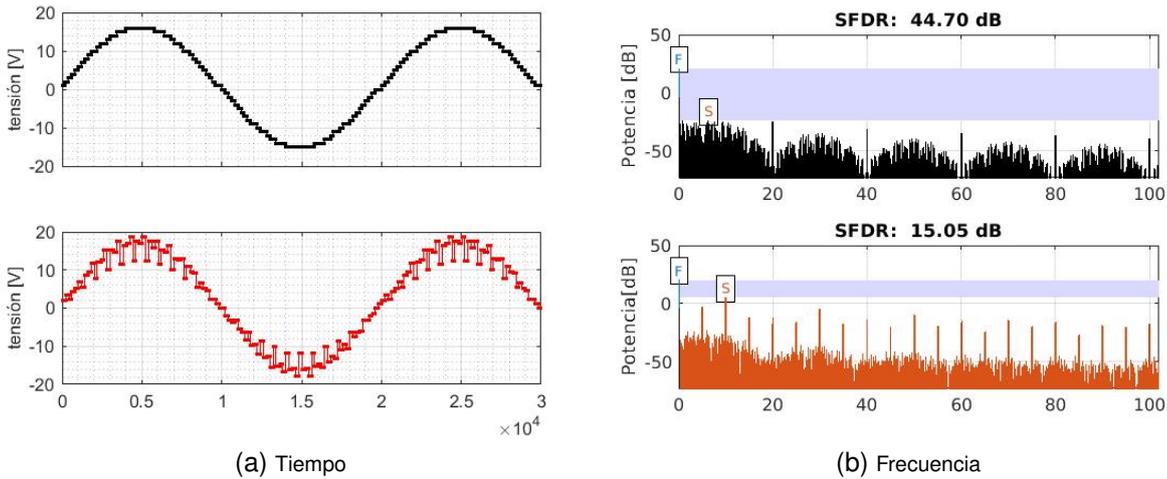


Fig. 1. En negro señal muestreada con un único ADC operando a f_s , en naranja señal muestreada con 4 ADCs trabajando a $f_s/4$ cada uno.

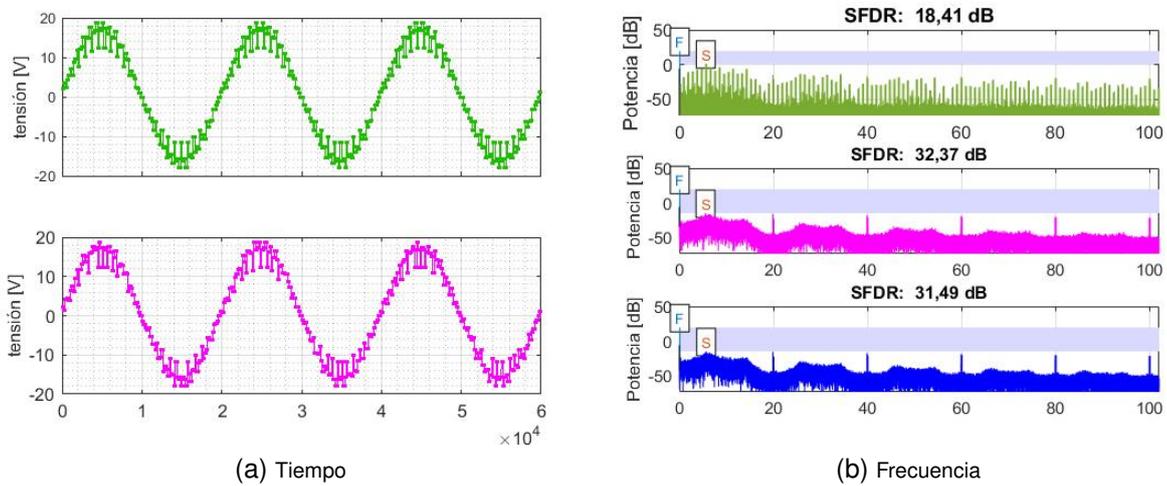


Fig. 2. Señales muestreadas con 4 ADCs y la técnica de TI aleatorio con diferentes PRNGs. En verde con un LFSR de período 7, en magenta con LFSR de período 8.191 y en azul la función rand de Matlab. Nótese que se ha omitido la señal temporal con rand y la LFSR período 8.191 ya que no se puede detectar diferencia en el tiempo.

ADC. En la Fig. 2.a no es posible detectar diferencia en el empleo de distintos períodos, sin embargo en el espectro (Fig. 2.b) se ve claramente como las interferencias disminuyen y el SFDR aumenta a medida que el período del LFSR aumenta (en verde aleatorizando con LSFR de período 7, en magenta LSFR de período 8.191 y en azul utilizando la función rand de Matlab). Se puede ver que la potencia total (ruido e interferencias) es la misma en todos los casos, el efecto de seleccionar los ADCs aleatoriamente es disminuir la amplitud de la interferencia.

En la Fig.3 se puede ver cómo varía el SFDR para cada período de PRNG empleado. Se realizaron simulaciones para distintas frecuencias de muestreo, los detalles de la simulación pueden verse en [5]. En la figura se ve que a partir de un período del LFSR de 511 el SFDR se mantiene prácticamente constante, inclusive para el caso de la función rand de Matlab. En el margen derecho de la figura se reporta el valor de SFDR del caso ideal en el que se utiliza un único ADC.

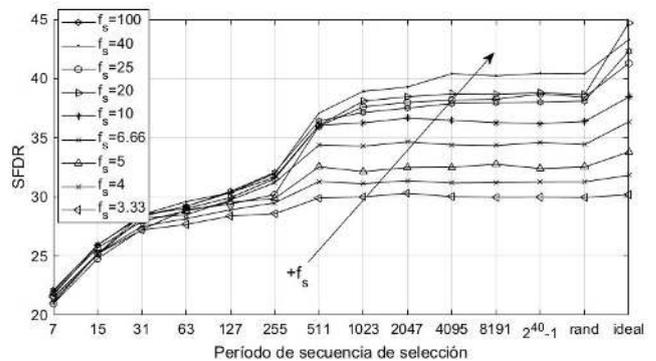


Fig. 3. Rango Dinámico Libre de Espurios versus períodos de secuencias de aleatorización para diferentes frecuencias de muestreo. Las secuencias se generaron con distintos LFSRs y con la función rand de Matlab (indicada como rand en la figura). En el extremo derecho, indicado como ideal, se puede ver el SFDR para el caso del muestreo con un único ADC.

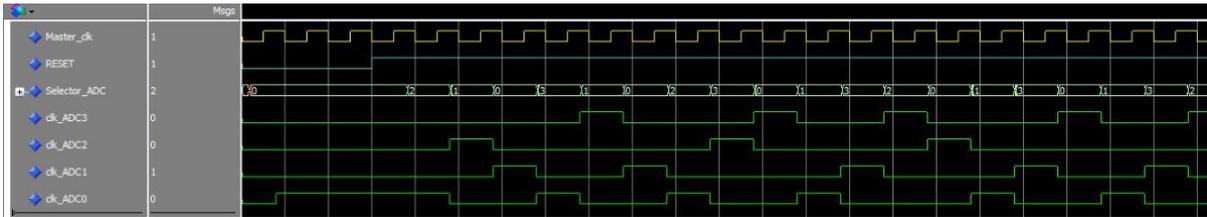


Fig. 4. Simulación del circuito de generación de las señales de reloj para los ADCS mediante ModelSim.

III. CIRCUITO IMPLEMENTADO

Un esquema del circuito implementado puede verse en la Fig. 7. La FPGA se encarga de generar las señales de reloj para que los ADCs estén debidamente desfasadas. También se encarga de recibir los datos y almacenarlos en la memoria del dispositivo, para luego ser transmitidos a una PC mediante transmisión serie. El circuito incluye también el generador de números pseudoaleatorios y el mecanismo de selección de ADC, considerando que la frecuencia de señal de reloj de cada ADC no debe superar f_s/M . Donde M es el grado de disminución de la frecuencia y esta determinado por la cantidad de ADCs que se empleen. En el caso TI aleatorio será la cantidad de ADCs menos uno, ya que uno de los ADCs deberá utilizarse para permitir la aleatorización. En este caso se emplearon 4 ADCs por lo que $M=3$, y en consecuencia, cada ADC recibirá una frecuencia $f_s/3$. La Fig. 5 muestra el árbol de selección de ADC para el caso $M=3$. Cada nodo representa uno de los cuatro ADCs. Se supone que se comienza seleccionando el ADC0 luego, aleatoriamente (mediante el PRNG), se va decidiendo entre una de las dos ramas que salen de cada nodo. Nótese que cada ADC es seleccionado no antes de $M-1$ ciclos ya que de otra forma aumentaría la frecuencia de reloj que esta recibiendo.

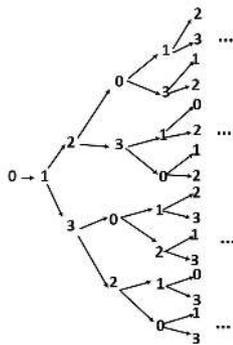


Fig. 5. Árbol de secuencia de selección para 4 ADCs ($M=3$).

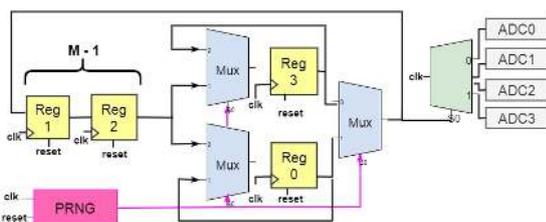


Fig. 6. Esquema propuesto de selección de ADC implementado en la FPGA.

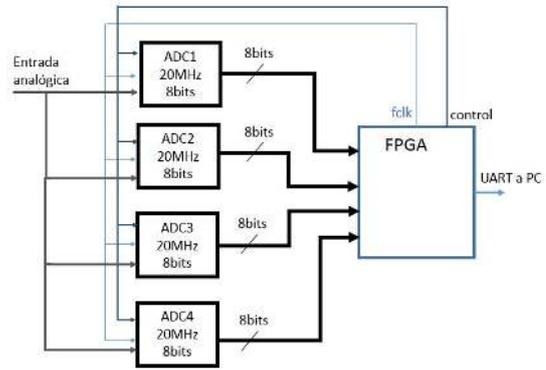


Fig. 7. Esquema general del conversor implementado.

En la Fig. 6 se puede observar el esquema del selector propuesto que permite, en base al PRNG, generar las señales de reloj a cada ADC. Estas señales de reloj son mostradas en la Fig. 4 donde se observa que la frecuencia de la señal de reloj recibida por cada ADC es 1/3 de la señal de reloj entrante.

IV. CONCLUSIONES

Se diseñó e implementó un adquisidor de señales mediante la técnica de TI aleatorio. Se estudió el efecto del período del PRNG empleado para la selección del ADC, sobre la performance del sistema. Se encontró que a partir de un período de 511 de la secuencia aleatoria el SFDR no aumenta en forma significativa. Como trabajo futuro se pretende estudiar el efecto de utilizar distintos PRNGs.

AGRADECIMIENTOS

Este trabajo fue parcialmente financiado por CONICET, Agencia I+D+i (PICT19 3024) y la FI-UNMdP, Argentina.

REFERENCIAS

- [1] G. Manganaro and D. Robertson, "Interleaving adcs: Unraveling the mysteries," *Analog Dialogue*, vol. 49, no. 7, 2015.
- [2] M. Chakravarthi and B. Chandramohan, "Calibration of mismatches in time-interleaved adcs using teacher learner-based optimization algorithm," *Journal of Circuits, Systems and Computers*, p. 2250163, 2022.
- [3] V. N. C. Manepalli and C. Bhuma, "Optimization algorithms based compensation of mismatches in time interleaved analog to digital converters-a review," *International Journal of Electrical and Computer Engineering Systems*, vol. 13, no. 3, pp. 237-244, 2022.
- [4] M. Tamba, A. Shimizu, H. Munakata, and T. Komuro, "A method to improve sfd with random interleaved sampling method," in *Proceedings International Test Conference 2001 (Cat. No. 01CH37260)*. IEEE, 2001, pp. 512-520.
- [5] L. DE Micco, "GitHub," GitHub repository, 2022. [Online]. Available: <https://acortar.link/7L9dnD> (accedido el 25 de junio de 2022).

Sistema para el registro continuo de variables cardiovasculares durante pruebas de basculación

Luis Antonio Noa Llorens
Departamento de Bioingeniería
Ministerio de Educación Superior
La Habana, Cuba
luis.nl@automatica.cujae.edu.cu

Yamel Jesús Almeida Pichardo
Departamento de Bioingeniería
Ministerio de Educación Superior
La Habana, Cuba
yamelalmeida97@gmail.com

Angel Regueiro Gómez
Departamento de Bioingeniería
Ministerio de Educación Superior
La Habana, Cuba
regueiro@automatica.cujae.edu.cu

Carmenhu Regueiro Busoch
Departamento de Automática
Ministerio de Educación Superior
La Habana, Cuba
carmenhu@automatica.cujae.edu.cu

Carmen B. Busoch Morlán
Departamento de Automática
Ministerio de Educación Superior
La Habana, Cuba
cbusoch@automatica.cujae.edu.cu

Resumen—Se ha diseñado un nuevo sistema para la adquisición diferencial y continua de la presión arterial durante las pruebas de basculación basado en el uso del método oscilométrico de oclusión parcial. El canal de medición se conecta a un kit ARDUINO UNO como unidad de control y envía la información obtenida a un ordenador vía USB. Se ha desarrollado una aplicación con empleo de LabVIEW que permite el control y procesamiento de diferentes variables. Los resultados muestran una buena detección de la presión arterial media relacionada con el ángulo de inclinación del sujeto. A medida que aumenta el ángulo, aumenta la diferencia en la presión arterial media. El sistema permite el estudio de enfermedades cardiovasculares como el síndrome vasovagal.

Palabras clave—presión sanguínea, prueba basculante, adquisición continua diferencial, método de oclusión parcial

I. INTRODUCCIÓN

La prueba de la mesa basculante (PMB) se ha utilizado desde 1986 de forma continua como prueba complementaria para el diagnóstico de pérdida transitoria de conciencia o síncope. Su forma más común es el síncope vasovagal debido a una disfunción del sistema autónomo. Durante el síncope la presión sanguínea arterial tiende a disminuir sin compensación periférica con el aumento de la inclinación de la cama basculante (Fig. 1) [1, 2].

A. Registros necesarios durante la prueba de la cama basculante

Durante el estudio cardiovascular de pacientes en la PMB es necesario caracterizar de forma continua (latido a latido) el registro de varias variables fisiológicas del sistema cardiovascular, entre ellas: electrocardiograma (ECG) y presión sanguínea arterial (PSA) para la detección de eventos hemodinámicos característicos de patologías asociadas al sistema circulatorio [1]. El registro pulso a pulso de la PSA en relación al ECG es trascendental pues esta variable depende indirectamente de diversos factores, tales como frecuencia cardíaca y tiempo de tránsito de pulso (TTP) [3].

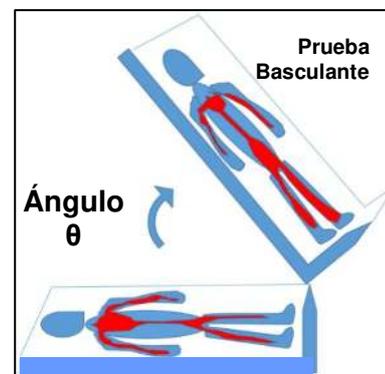


Fig. 1. Representación de una prueba de basculación (la presión arterial cambia con diferentes posiciones)

Durante la PMB los síntomas iniciales típicos incluyen taquicardia compensatoria y empeoran hasta que se produce el síncope que resulta en hipotensión y bradicardia. Estos eventos fisiológicos se ven reflejados en el registro de la PSA y el ECG, donde se observará la hipotensión y los cambios de frecuencia cardíaca subyacentes latido a latido [4].

B. Métodos de obtención de la presión sanguínea arterial

La onda de presión arterial se puede obtener de forma invasiva mediante la inserción de un catéter arterial o de forma no invasiva a través de lecturas intermitentes o continuas [5]. Los métodos invasivos conllevan riesgos asociados a la punción de la arteria que incluyen hemorragias, infección e isquemia por lo que no se emplean durante la PMB. Es entonces necesario considerar métodos no invasivos capaces de detectar los eventos hemodinámicos asociados al síncope [6].

Hoy día los sistemas convencionales para la medición no invasiva de PSA suelen ser intermitentes, a menudo se establecen las mediciones a intervalos bastante largos (entre 5 y 10 minutos), lo que provoca que la hipotensión no se detecte adecuadamente, mientras que los métodos de medición continua no invasiva de la PSA suelen ser caros, difíciles de calibrar e incómodos [7].

Los estudios cardiovasculares en la PMB tienen una duración aproximada entre 60 y 80 minutos, tiempo durante el cual varía la inclinación de la camilla (entre 0 y 70 grados).

Cuando el especialista clínico utiliza un método no invasivo (auscultatorio u oscilométrico), debe detener la adquisición en un intervalo mínimo de cinco minutos entre lecturas, debido a que los pacientes necesitan restablecer la circulación en el miembro ocluido (típicamente la extremidad superior) durante el ciclo de llenado/vaciado del brazalete [8, 9].

La nueva alternativa desarrollada resuelve estos inconvenientes y consiste en el registro continuo pulso a pulso a través de la colocación de dos sensores de presión conectados a dos brazaletes en las extremidades del sujeto, (brazo y pierna), de tal forma que es posible la medición diferencial continua de la variable presión sanguínea arterial durante todo el desarrollo de la prueba de basculación mediante una oclusión parcial y controlada del inflado de los brazaletes. La adquisición paralela de ECG abre la posibilidad de detectar cambios en la frecuencia cardiaca analizar la variabilidad del TTP [3].

II. MATERIALES Y MÉTODOS

Se han estudiado 25 sujetos voluntario supuestamente sanos (60 % hombres y 40 % de mujeres) con edades entre 20 y 61 años, los cuales estuvieron de acuerdo en la realización de las pruebas de laboratorio con consentimiento escrito informado.

A. Sistema biomédico para medición continua de parámetros fisiológicos

Utilizando la herramienta LabVIEW (GUI) se desarrolló una aplicación de control de un Kit de desarrollo (Arduino UNO) [10] implementándose un sistema de adquisición que permite registrar simultáneamente la señal de PSA de manera diferencial y continua para su empleo clínico durante el desarrollo de la prueba de basculación (PMB), y además permite adquirir otras señales clínicas (ECG y onda de pulso fotopleletismográfico) como referencias temporales.

El sistema de registro para las variables cardiovasculares de interés clínico se muestra en la figura 2.

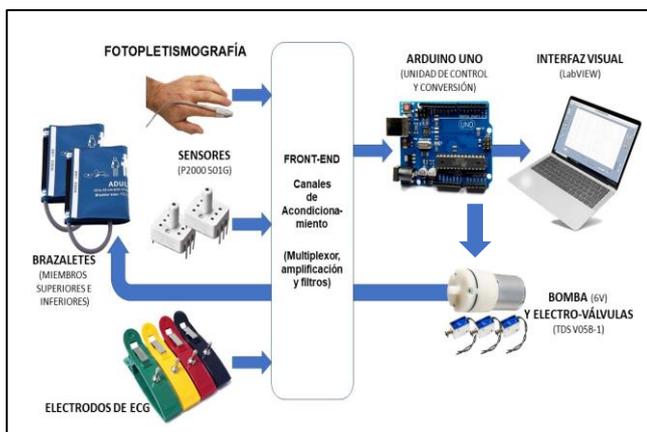


Fig. 2. Diagrama en bloques general del sistema de registro para las variables cardiovasculares de interés clínico durante la PMB

B. Protocolo de medición del Sistema Biomédico desarrollado

El experimento consta de dos fases fundamentales:

- 1) medición de presión arterial media y parámetros cardiocirculatorios basales del sujeto en posición horizontal
- 2) medición continua diferencial a través de la oclusión parcial de los miembros (brazo y pierna) durante un ciclo de basculación

A través de un protocolo básico de medición oscilométrica de la presión sanguínea se calcula la presión arterial media (PAM) que servirá de referencia para la segunda fase del estudio cardiovascular. La variabilidad y línea base de la frecuencia cardiaca son registrados por medio de los complejos QRS luego de un posterior procesamiento digital a todo el registro a través el algoritmo de Pan Tompkins [11]; similarmente se calcula el TTP diferencial latido a latido utilizando como referencia el pie de onda de la curva fotopleletismográfica. Los electrodos de ECG son colocados en la derivación aVL donde se aprecia un marcado complejo QRS.

En un segundo momento se toma como referencia los parámetros calculados previamente en posición horizontal y se comienzan a registrar de forma continua los cambios fisiológicos que ocurren durante una basculación completa. Los datos son interpretados en un *software* especializado donde el usuario puede seleccionar el archivo de registro y analizar los parámetros personalizados del sujeto bajo estudio.

Se han adicionado algunas funciones simples de procesamiento a la aplicación (filtrado digital, FFT y otros) que sirven para eliminar bandas de ruido e interferencias en cada registro en particular durante el preprocesamiento de la información.

III. RESULTADOS Y DISCUSIÓN

A. Fase I del protocolo de medición

La figura 3 muestra el registro simultáneo de la PAM en ambos brazaletes junto al registro de ECG durante un ciclo de basculación. La medición inicial de la PAM del sujeto permite personalizar el estudio del registro continuo de esta compleja variable.

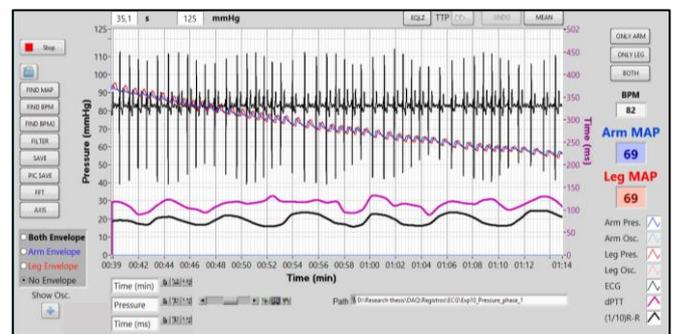


Fig. 3. Ejemplo de un registro de señales. (a) Sección superior: ECG (negro) y registro de presión en ambos brazaletes (azul-rojo). (b) Sección inferior: Variabilidad del TTP diferencial (rosado) y variabilidad del período R-R/10 (negro)

B. Fase II del protocolo de medición

En la figura 4a se ilustran los cambios de la presión en ambos brazaletes que aumentan conforme avanza el ángulo de basculación, mientras que en la figura 4b y 4c se observan los cambios en la amplitud de las oscilaciones de ambos brazaletes. La figura 4d muestra la variación de la variabilidad y la línea base latido a latido del período R-R/10.

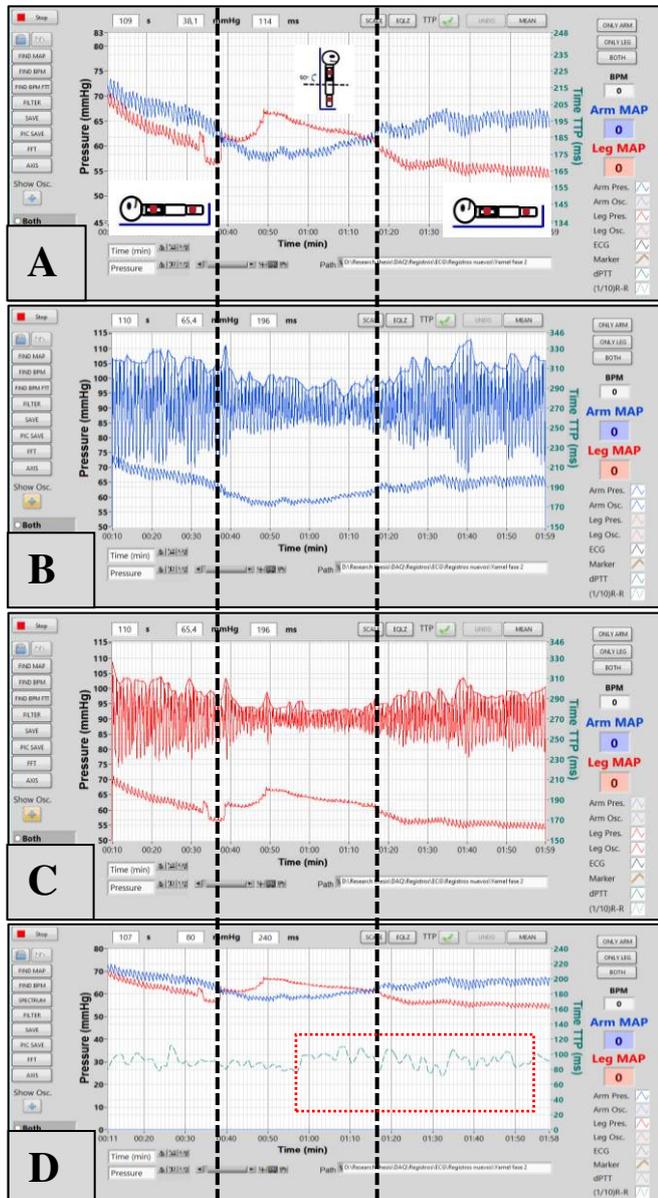


Fig. 4. a) Registro de PSA en ambos brazaletes en distintos ángulos de inclinación. b) Registro de PSA en el brazaletes de brazo y sus respectivas oscilaciones. c) Registro de PSA en el brazaletes de pierna y sus respectivas oscilaciones. d) Registro del tiempo R-R/10 (ms) (celeste), se destaca en rojo los cambios fundamentales en la última etapa

En todos los sujetos estudiados el comportamiento de los registros de presión arterial fue similar y no se apreciaron diferencias significativas en los resultados entre hombres y mujeres. A medida que se desarrolla la prueba de inclinación, aumenta la diferencia entre los registros de presión arterial media del sujeto bajo estudio, siendo siempre mayor la presión en la extremidad inferior (pierna) a medida que aumenta el ángulo de inclinación, esto puede ser considerado como un indicador clínico del buen estado del sistema circulatorio (propagación del flujo sanguíneo).

El empleo del registro diferencial continuo de presión arterial permite mejorar los análisis clásicos de la prueba de basculación, basados en mediciones simples e intermitentes generalmente en la extremidad superior (brazo). El empleo del método oscilométrico con oclusión parcial no afecta la circulación periférica del sujeto bajo estudio, lo cual permite un registro continuo durante las pruebas de basculación con mejores prestaciones que otros métodos no invasivos (método fotopletofráfico), y abre nuevas opciones de análisis de esta importante variable en la detección de enfermedades cardiovasculares de forma personalizada.

IV. CONCLUSIONES

Durante la PMB, prueba diagnóstica para síncope vasovagal, es necesario un registro latido a latido de frecuencia cardiaca y presión sanguínea debido a los eventos abruptos que ocurren desembocando en la pérdida del tono simpático. El método propuesto registra continuamente los cambios en estos parámetros fisiológicos de forma no invasiva con un diseño de bajo costo y con el mayor peso en una interfaz desarrollada en LabVIEW (Interfaz de registro de variables), que brinda el acceso a todos los parámetros temporales de los registros adquiridos brindando adecuada trazabilidad y robustez para los estudios cardiovasculares con ayuda de pruebas de basculación.

V. REFERENCIAS

- [1] M. Dani et al., "Vasovagal Syncope: A Review of Current and Future Strategies," *European Journal of Arrhythmia & Electrophysiology*, pp. 40-47, 2021, doi: <https://doi.org/10.17925/EJAE.2021.7.1.40>.
- [2] S. G. Chrysant, "The tilt table test is useful for the diagnosis of vasovagal syncope and should not be abolished," 2020 Wiley Periodicals, Inc., vol. 22, pp. 686-689, 2020, doi: <https://doi.org/10.1111/jch.13846>.
- [3] R. C. Block et al., "Conventional pulse transit times as markers of blood pressure changes in humans," *Scientific Reports*, 2020, doi: <https://doi.org/10.1038/s41598-020-73143-8>.
- [4] R. Kohno, W. O. Adkisson, and D. G. J. H. E. Benditt, "Tilt table testing for syncope and collapse," vol. 29, no. 2, pp. 187-192, 2018, doi: <https://doi.org/10.1007/s00399-018-0568-9>.
- [5] K. Babadağ and A. Zaybak, "Comparing Intra-Arterial, Auscultatory, and Oscillometric Measurement Methods for Arterial Blood Pressure," *Florence Nightingale Journal of Nursing*, vol. 2, 29, pp. 194-202, 2021, doi: <https://doi.org/10.5152/FNJJN.2021.19103>.
- [6] S. Devasahayam, C. Surekha, B. Baskaran, N. Gangadharan, F. A. Mukadam, and S. J. m. Subramani, "Intra-Arterial Blood Pressure Measurement: Sources of Error and Solutions," 2020.
- [7] K. Shimura, M. Hori, T. Dohi, and H. Takao, "The Calibration Method for Blood Pressure Pulse Wave Measurement Based on Arterial Tonometry Method," in 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2018: IEEE, pp. 1-4.
- [8] Manual, electronic, or automated sphygmomanometers, AAMI, 2002. [Online]. Available: www.aami.org
- [9] G. S. Stergiou et al., "A universal standard for the validation of blood pressure measuring devices: Association for the Advancement of Medical Instrumentation/European Society of Hypertension/International Organization for Standardization (AAMI/ESH/ISO) Collaboration Statement," *Hypertension*, vol. 71, no. 3, pp. 368-374, 2018, doi: <https://doi.org/10.1161/HYPERTENSIONAHA.117.10237>.
- [10] D. H. Sallo, "Design and implementation of low cost data acquisition system for ECG signal and EDR using LabVIEW and Arduino," *Journal of University of Duhok*, pp. 106-112, 2019, doi: <https://doi.org/10.26682/sjuod.2019.22.1.11>
- [11] M. Fariha, R. Ikeura, S. Hayakawa, and S. Tsutsumi, "Analysis of Pan-Tompkins algorithm performance with noisy ECG signals," in *Journal of Physics: Conference Series*, 2020, vol. 1532, no. 1: IOP Publishing, p. 012022.

Categoría

FORO TECNOLÓGICO

FPGAs, HDLs y ASICs

Área Temática

Implementation of CIC filters for decimation of 532 nm atmospheric LIDAR signals on FPGA.

1st Martín Alejandro Heredia
 Grupo ASE - UTN FR Haedo
 Haedo, Argentina
 martinalejandroheredia@gmail.com

2nd Facundo Larosa
 Grupo ASE - UTN FR Haedo
 Haedo, Argentina
 facundolarosa@gmail.com

3rd Héctor Alberto Lacomí
 Grupo ASE - UTN FR Haedo
 DRL - CITEDEF
 Haedo, Argentina
 hlacomí@frh.utn.edu.ar

Abstract—Atmospheric LIDAR (Light Detection And Ranging) systems are widely used to detect and characterize aerosols, clouds and particles suspended in troposphere and stratosphere. The study of such phenomena in the atmosphere is important to understand the climate system of our planet and predict its evolution. For those reasons, application of DSP (Digital Signal Processing) techniques to process and adequate LIDAR signals is mandatory. This paper focuses on the implementation of efficient CIC (Cascaded Integrator-Comb) architectures on FPGA (Field Programmable Gate Array) to minimize resource utilization and complexity to process LIDAR signals in an efficient way, while adding scalability from a hardware design perspective. The CIC architecture proposed was evaluated using real 532 nm LIDAR signal. Results obtained with the CIC architecture proposed were compared with FIR filter, both synthesized on a Xilinx SoC XC7Z010-1CLG400C, which includes a 28 nm Artix 7 FPGA. This comparison is encouraging since the proposed CIC filter structure uses approximately 4% of LUTs and Logic of the single FIR filter.

Index Terms— LIDAR, Atmosphere, DSP, CIC, decimation, FPGA

I. INTRODUCTION

Atmospheric LIDARs are devices that use laser pulses to study properties of the atmosphere. They are capable of sensing scattering produced by particles, aerosols and air molecules [1]. Those systems are mainly composed of a laser pulse transmitter and a receiver that captures the backscattered signal. This receiver consists of an optical front-end and an acquisition card [2]. Regarding the acquisition electronics, many solutions can be found in literature. They use either specific equipment [3], orbiting instruments on satellites [4] or general purpose hardware plus proprietary software [2]. For many of these cases, their cost and static architecture may be prohibitive for a wider spectrum of applications. On the other hand, there are custom systems that can be adapted to diverse use cases. As an example, acquisition cards based on Zynq [5] devices are proposed in [6], [7] and [8]. This paper will be focused on the improvement of data processing for a custom acquisition system like the one proposed in [8]. For that purpose, CIC filters and decimation will be presented.

CIC filters were introduced in [9] as an economical implementation of moving average filters. This architecture uses 2-input adders and memory elements to compute its output. In addition, they can be scaled by changing the parameter D , which represents the number of input samples to average. The

block diagram of this architecture is presented in Fig. 1. Although some characteristics of CIC filters may be considered a disadvantage [10], their richness is demonstrated when they are used as pre filters in decimation applications especially when these are used in embedded systems based on FPGA [11].

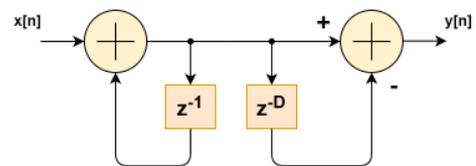


Fig. 1. Architecture block diagram.

Decimation is the process of removing samples from an incoming signal in a R factor [12]. This can be seen as taking only one out of R elements and dropping the rest. Since many samples are dropped by decimation, the computational load for them can be removed. This approach is depicted in Fig. 2, where it can be noted that, assuming downsampling factor R is equal to D parameter of CIC filter, there is only one register needed in comb section. Thus, the introduction of downsampling within the architecture of CIC filters implies a further improvement in resource utilization

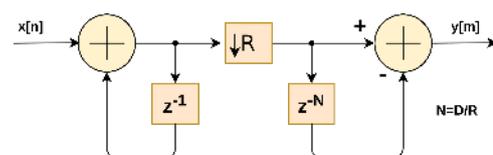


Fig. 2. Downsampling factor in CIC Filter.

Given an input signal sampled at a frequency f_s with a bandwidth well below the sample rate, the implementation of a FIR filter working at high sample rate with narrow pass band and high stopband attenuation implies a large amount of filter taps, increasing the complexity. To save computational load, an architecture composed by a CIC decimation filter followed by a FIR filter (this last working at a reduced sampling rate $\frac{f_s}{R}$) is proposed [13].

In this paper, the proposed CIC+FIR architecture will be implemented in order to downsample 532 nm LIDAR sig-

nals acquired with an FPGA based data acquisition system [8]. Section II will cover the approach followed to analyze the bandwidth of interest in 532 nm LIDAR signals, which will determine the value of decimation factor. In section III, implementation details of the proposed architecture will be addressed. In addition, the implementation of a complex FIR filter working at a high sample rate will be presented and compared with the CIC+FIR design. Section IV presents the functional validation of the system and compares both architectures. To conclude, section V presents future work and improvements that can be introduced to the proposed

II. TESTING METHODS

In order to determine the downsampling factor of the system, bandwidth of LIDAR signals must be known. Since the system used to capture and process those signals has a sampling rate of approximately 77 MHz [8], Nyquist frequency will have a value of 38.5 MHz. However, the highest frequency component of LIDAR signals may be found well below 38.5 MHz. To evaluate the highest frequency components of those signals, a study of spectral energy was proposed. This approach was based on the computation of FFT for averaged LIDAR signals (with low noise), followed by application of Parseval's theorem to obtain the energy. Then, a cumulative sum from 4 KHz up to Nyquist frequency was applied to find the frequency point where 95% of energy was accumulated. The Fig.3 shows the boxplot obtained for two LIDAR channels of 532 nm. It can be seen that 95% of its energy is placed below 1 MHz.

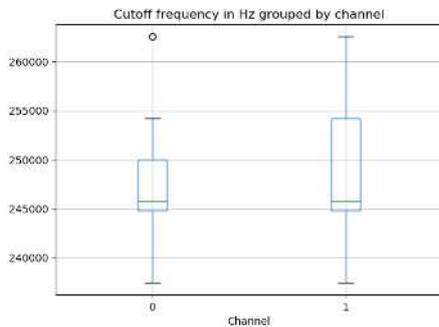


Fig. 3. Boxplot for two 532 nm LIDAR channels.

III. ARCHITECTURE OF DECIMATOR

Given the bandwidth computed in section II, a decimation factor of $R=15$ was proposed to reduce sampling frequency from approximately 77 MHz to 5.13 MHz. In addition, the number of averages of the CIC filter was selected as $D = R = 15$. In order to improve stopband attenuation of CIC filter, a total number of 3 stages (noted as $Q = 3$) were implemented. Fig. 4 shows the magnitude response of the proposed CIC filter.

In cascade with CIC filter, a 17 taps FIR filter working at $\frac{f_s}{R} = 5.13$ MHz was introduced to improve attenuation in transition band. This can be seen in Fig. 5

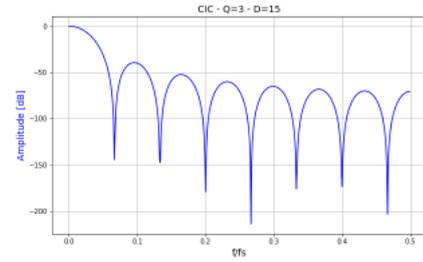


Fig. 4. Magnitude response of CIC filter.

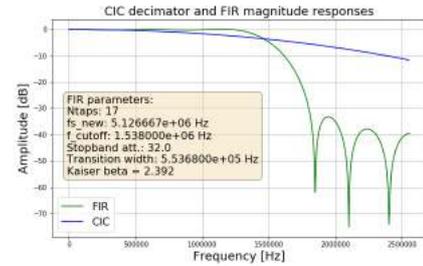


Fig. 5. CIC filter + 17 taps FIR filter response.

If the same decimation process is performed using a single FIR filter, the complexity of the resultant system increases [9], [14]. To evaluate this, a single stage FIR filter working at $f_s = 77$ MHz followed by a $R = 15$ downsample stage was designed using **Kaiser method** [14]. Fig. 6 shows the magnitude response of the filter and the parameters used for Kaiser window. It can be noted that the total amount of filter taps is 234.

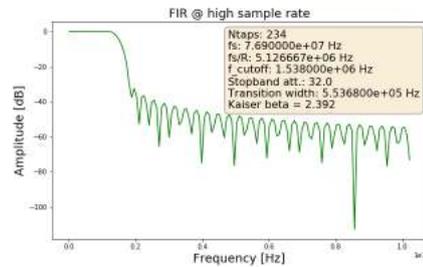


Fig. 6. Filter magnitude response and Kaiser method parameters.

IV. RESULTS

The comparison of input and output (downsampled) signals from CIC+FIR decimator is shown in Fig. 7. The amplitude of both signals is normalized and the horizontal axis represents samples of input signal (note that output signal will have R times less samples than input).

To compare the complexity of proposed CIC+FIR architecture against single stage FIR, both systems were implemented in Verilog and synthesized for a XC7Z010-1CLG400C Xilinx SoC. Table I shows the utilization report after synthesis. It can

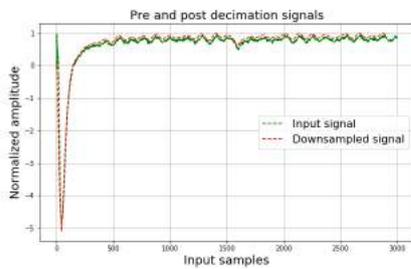


Fig. 7. CIC filter + 17 taps FIR filter response.

TABLE I
CIC + FIR VS FIR EMPLOYED RESOURCES

	LUTs and Logic	Flip Flops	DSP blocks
CIC + FIR	2.9%	0.66%	21.25%
FIR	77.03%	23.7%	0%

be noted that single stage FIR has no DSP blocks utilization, while CIC+FIR architecture consumes 21.25% of the total. Since the device selected has 80 DSP blocks, a FIR filter of 234 taps cannot be fully implemented using those resources. The last increases utilization of LUTs as logic.

V. CONCLUSION AND FUTURE WORK

In this paper, an architecture of 3rd order CIC decimator followed by a 17 taps FIR filter was introduced as an efficient way to downsample LIDAR signals. The system was implemented in Programmable Logic of XC7Z010-1CLG400C SoC and validated with functional simulations. A sample rate reduction of 15 times was achieved, allowing data processing at a lower clock frequency. In addition, the decimation of input samples captured per LIDAR pulse reduces the size of memory blocks required for further digital processing. Regarding scalability, stopband attenuation of CIC decimator can be improved by changing the amount of CIC stages (integrator + comb) used in RTL design. This last can be defined as a parameter Q of the filter. Furthermore, the decimation factor can also be modified with a single parameter (D). For future work, there are some aspects that can be improved in this architecture:

- Utilization of compensation filters [10], [15] to correct passband attenuation
- Take advantage of symmetry in filter taps to implement a reduced FIR stage after CIC decimator.
- Implement FIR filter in its transpose direct form (TDF) to reduce length of critical path and improve timing closure [16].
- Introduce round and truncation within the filters to reduce binary word size and improve resource utilization [9].
- Implementation of a validation bench on SoC to read and process multiple input files in a single test case.
- Capture and downsample LIDAR signals in real time to validate system performance.

REFERENCES

- [1] J. Reagan, M. McCormick, and J. SPinhirne, "Lidar sensing of aerosols and clouds in the troposphere and stratosphere," *Proceedings of the IEEE*, vol. 77, no. 03, p. 433–448, 1989.
- [2] J. Mao, X. Cheng, H. Zhao, H. Sheng, C. Zhou, and B. Zhang, "Design of data acquisition and processing system based on labview for a mie lidar," in *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2019, p. 1942–1946.
- [3] C.-W. Chiang, S. K. Das, H.-W. Chiang, J.-B. Nee, S.-H. Sun, S.-W. Chen, P.-H. Lin, J.-C. Chu, C.-S. Su, and L.-S. Su, "A new mobile and portable scanning lidar for profiling the lower troposphere," *Geoscientific Instrumentation, Methods and Data Systems*, vol. 04, no. 01, p. 35–44, 2015. [Online]. Available: <https://gi.copernicus.org/articles/4/35/2015>
- [4] J. Spinhirne, S. Palm, D. Hlavka, W. Hart, A. Mahesh, and E. Welton, "Aerosol and cloud measurements at 532 and 1064 nm by the glas polar orbiting lidar instrument," in *IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium*, vol. 06, 2004, pp. 3956–3959 vol.6.
- [5] *Zynq-700 SoC Technical Reference Manual*, v1.13 ed., Xilinx, April 2021. [Online]. Available: <https://docs.xilinx.com/v/u/en-US/ug585-Zynq-7000-TRM>
- [6] L. Cheng, C. Xie, Y. Cao, P. Zhuang, Z. Fang, and S. Fu, "Design of miniature data acquisition system for atmospheric detection lidar based on zynq," in *Sixth Symposium on Novel Optoelectronic Detection Technology and Applications*, ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 11455, 2020.
- [7] J. Huang, S. Ran, W. Wei, and Q. Yu, "Digital integration of lidar system implemented in a low-cost fpga," *Symmetry*, vol. 14, no. 06, 2022. [Online]. Available: <https://www.mdpi.com/2073-8994/14/6/1256>
- [8] H. A. Lacomini, T. D. Fiore, N. U. Pintos, and F. S. Larosa, "An fpga-based reconfigurable data acquisition system for lidar signal detection," *International Journal of Embedded Systems*, vol. 14, no. 6, pp. 527–534, 2021.
- [9] E. Hogenauer, "An economical class of digital filters for decimation and interpolation," *IEEE transactions on acoustics, speech, and signal processing*, vol. 29, no. 2, pp. 155–162, 1981.
- [10] A. Fernandez-Vazquez and G. J. Dolecek, "Maximally flat cic compensation filter: Design and multiplierless implementation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, no. 2, pp. 113–117, Feb. 2012. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6122060>; <https://ieeexplore.ieee.org/document/6122060/>
- [11] K. Grover, R. Mehra, and Chandni, "Fpga based decimator using fully parallel technique for hearing aid applications," in *2017 3rd International Conference on Computational Intelligence & Communication Technology (CICCT)*. IEEE, Feb. 2017, pp. 1–6. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7977358>; <https://ieeexplore.ieee.org/document/7977358/>
- [12] D. Sinha and S. Kumar, "Cic filter for sample rate conversion in software defined radio," in *2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)*. IEEE, Feb. 2016, pp. 1–5. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7583972>; <https://ieeexplore.ieee.org/document/7583972/>
- [13] —, "Fir filter compensator for cic filter suitable for software defined radio," in *2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)*. IEEE, Feb. 2016, pp. 1–7. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7583915>; <https://ieeexplore.ieee.org/document/7583915/>
- [14] R. Arya and S. Jaiswal, "Design of low pass fir filters using kaiser window function with variable parameter beta (β)," *International Journal of Multidisciplinary and Current Research*, vol. 3, pp. 220–224, 2015.
- [15] G. Dolecek and S. Mitra, "Simple method for compensation of cic decimation filter," *ELECTRONICS LETTERS-IEE*, vol. 44, no. 19, pp. 1162–1162, 2008. [Online]. Available: <https://www.ingentaconnect.com/content/iee/00135194/2008/00000044/00000019/art00039>
- [16] Aripasath and D. C. Santhi, "Transpose form fir filter design for fixed and reconfigurable coefficient," in *International Research Journal of Engineering and Technology (IRJET)*, vol. 04, Mar. 2017, p. 1857–1862.

Aceleración de simulación de circuitos cuánticos parametrizados en SoC FPGA

Agustin Silva

Facultad de Ingeniería (ICYTE)
Universidad Nacional de Mar del Plata
Av. Juan B. Justo 4302, B7608
Mar del Plata, Argentina.
Email: agustinsilva447@gmail.com

Claudio M. González

Facultad de Ingeniería (ICYTE)
Universidad Nacional de Mar del Plata
Av. Juan B. Justo 4302, B7608
Mar del Plata, Argentina.
Email: cmgonzal@fi.mdp.edu.ar

Omar G. Zabaleta

Facultad de Ingeniería (ICYTE)
Universidad Nacional de Mar del Plata
Av. Juan B. Justo 4302, B7608
Mar del Plata, Argentina.
Email: cmgonzal@fi.mdp.edu.ar

Resumen—Los circuitos cuánticos parametrizados son los principales candidatos a ser los primeros en garantizar una ventaja frente a algoritmos puramente clásicos de optimización y aprendizaje automático. Estos circuitos híbridos se deben ejecutar una gran cantidad de veces antes de garantizar los resultados esperados. En este trabajo, se diseña un sistema basado en FPGA que acelera notablemente (desde el orden de los mili a los nano segundos) la simulación de circuitos cuánticos parametrizados, en el contexto de la teoría de juegos cuántica. Con el fin de explicar los resultados obtenidos, se desarrollan tanto las consideraciones teóricas como prácticas que fueron tomadas en cuenta para obtener el rendimiento logrado. Asimismo, se realizan las simulaciones correspondientes para comparar las salidas y los tiempos de ejecución tanto en un microcontrolador como en una FPGA.

Palabras Claves—SoC FPGA, Circuitos Cuánticos, Simulación.

I. INTRODUCCIÓN

La computación cuántica ha llegado para quedarse. A pesar de las dificultades que enfrentan físicos e ingenieros a la hora de construir computadoras cuánticas que superen a las mejores computadoras clásicas, 40 años después de la primera propuesta de Richard Feynman, los avances en el área, tanto científicos como tecnológicos, no han cesado desde aquel momento [1].

Las potenciales aplicaciones de las computadoras cuánticas van desde optimización [2], seguridad cibernética [3] hasta el diseño de materiales [4]. Sin embargo, en la actualidad, no existen computadoras cuánticas ideales (con muchos qubits y tolerantes a errores) sino más bien imperfectas (pocos qubits y sensibles al ruido) [5]. Estas computadoras son limitadas, en cuanto a cantidad de qubits y compuertas disponibles, pero cuando se las combinan con computadoras clásicas, se pueden obtener algunas ventajas frente a los sistemas puramente clásicos.

En este contexto, los algoritmos variacionales con circuitos parametrizados son los principales candidatos para mostrar esta ventaja [6]. Estos algoritmos cuentan con un circuito cuántico como se describe en la Fig. 1 y una computadora clásica externa que evalúa una función de costo y, dependiendo de este valor, modifica los parámetros del circuito cuántico. El objetivo de estos algoritmos es similar al de otros tipos

de optimización: encontrar los parámetros que minimicen una cierta función de costo.

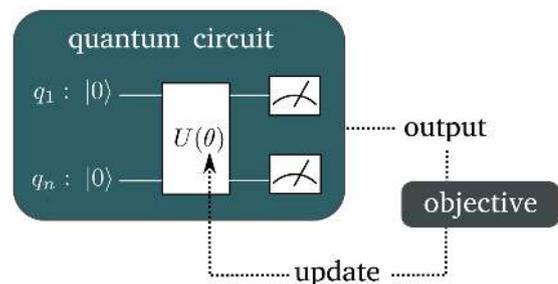


Fig. 1: Circuitos Cuánticos Parametrizados [Creditos: Xanadu.ai].

Este marco de trabajo ha despertado el interés de investigadores que lo utilizan como herramienta para tareas de aprendizaje automático debido a sus similitudes con las redes neuronales [7]. En nuestro caso, lo utilizamos en el contexto de la teoría de juegos cuántica [8] para encontrar el equilibrio de agentes que comparten un ambiente donde cada uno intenta minimizar su propia función de costo independientemente de los intereses de los demás.

El objetivo de este trabajo es lograr una aceleración considerable en el tiempo de procesamiento de las simulaciones de los circuitos cuánticos parametrizados en el contexto de la teoría de juegos cuánticos, utilizando FPGA. Si bien la emulación de circuitos cuánticos en FPGA es un tópico recurrente [9], [10], hasta donde llega el conocimiento de los autores, es la primera vez que se estudia la implementación de circuitos cuánticos parametrizados, en FPGA.

II. CIRCUITOS CUÁNTICOS PARAMETRIZADOS Y TEORÍA DE JUEGOS CUÁNTICA

El objetivo de la teoría de juegos es analizar sistemas de toma de decisiones que involucren dos o más jugadores que compiten entre sí. Por su parte, la teoría de juegos cuántica surge de considerar que las posibles estrategias que pueden seleccionar los jugadores pueden involucrar procesamiento

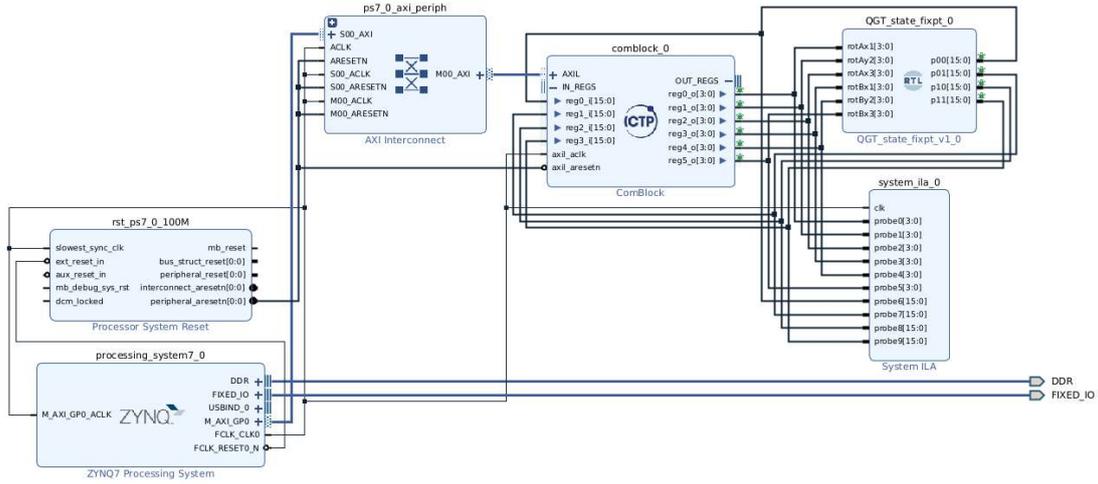


Fig. 3: Diagrama en bloques de la FPGA: Microprocesador Zynq, Comblock (interfaz AXI), Diseño en VHDL (circuito cuántico) e ILA (Analizador Lógico).

cuántico. En su versión más popular [11], las posibles estrategias son las compuertas cuánticas que pueden aplicar sobre un conjunto de qubits entrelazados en un circuito cuántico.

En la Fig. 2 se observa cómo cada jugador modifica los parámetros de un circuito cuántico y en función de eso recibe una recompensa. Los jugadores tienen el objetivo de encontrar el mejor conjunto de estrategias posible, procesando la realimentación R (recompensa inmediata), que le garantice maximizar su propia recompensa a largo plazo.

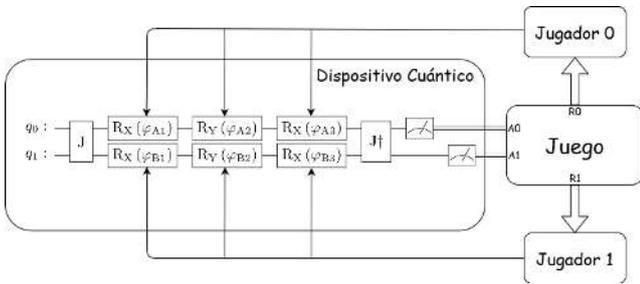


Fig. 2: Circuitos Cuánticos Parametrizados para el protocolo EWL.

En el esquema de dos jugadores de la Fig. 2, los jugadores deben elegir una estrategia (3 ángulos/parámetros): φ_{A1} , φ_{A2} y φ_{A3} (para el jugador 0) y φ_{B1} , φ_{B2} y φ_{B3} (para el jugador 1). Una vez que estas son seleccionadas, se debe ejecutar el circuito para obtener las acciones y recompensas de cada jugador y continuar con la siguiente iteración.

Para calcular el estado cuántico a la salida en función de las estrategias de los jugadores se deben evaluar las siguientes operaciones: $|\psi_{out}\rangle = J^\dagger \times (R_x(\varphi_{B3}) \otimes R_x(\varphi_{A3})) \times (R_y(\varphi_{B2}) \otimes R_y(\varphi_{A2})) \times (R_x(\varphi_{B1}) \otimes R_x(\varphi_{A1})) \times J \times |00\rangle$, siendo:

$$J = \frac{1}{\sqrt{2}}(\mathbb{I}^{\otimes N} + i\sigma_X^{\otimes N}) = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 0 & i \\ 0 & 1 & i & 0 \\ 0 & i & 1 & 0 \\ i & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$R_X(\varphi) = \begin{pmatrix} \cos(\frac{\varphi}{2}) & -i \sin(\frac{\varphi}{2}) \\ -i \sin(\frac{\varphi}{2}) & \cos(\frac{\varphi}{2}) \end{pmatrix} \quad (2)$$

$$R_Y(\varphi) = \begin{pmatrix} \cos(\frac{\varphi}{2}) & -\sin(\frac{\varphi}{2}) \\ \sin(\frac{\varphi}{2}) & \cos(\frac{\varphi}{2}) \end{pmatrix} \quad (3)$$

$|00\rangle = [1 \ 0 \ 0 \ 0]^t$, \otimes representa el producto Kronecker, \times representa el producto matricial y J^\dagger el transpuesto conjugado de J . Luego, con el estado cuántico a la salida se puede calcular las probabilidades de cada acción utilizando la regla de Born:

$$p_{out} = \begin{bmatrix} p_{00} \\ p_{01} \\ p_{10} \\ p_{11} \end{bmatrix} = \begin{bmatrix} |\psi_{00}|^2 \\ |\psi_{01}|^2 \\ |\psi_{10}|^2 \\ |\psi_{11}|^2 \end{bmatrix} \quad (4)$$

Finalmente, se obtiene una expresión que relaciona las estrategias de los jugadores con las probabilidades del conjunto de acciones de los jugadores: $p_{out}(\varphi_{A1}, \varphi_{A2}, \varphi_{A3}, \varphi_{B1}, \varphi_{B2}, \varphi_{B3})$. p_{out} es un vector de 4x1 y su cálculo involucra productos matriciales y de Kronecker utilizando números complejos. El objetivo de este trabajo es acelerar el cálculo del vector de probabilidades p_{out} utilizando FPGA.

III. IMPLEMENTACIÓN EN SOC FPGA

La mejor manera de calcular p_{out} es aprovechar que el circuito es fijo y lo único que cambian son sus parámetros. Por lo tanto, se calculan todas las multiplicaciones matriciales y de Kronecker utilizando valores genéricos de φ_{A1} , φ_{A2} , φ_{A3} , φ_{B1} , φ_{B2} y φ_{B3} utilizando Python y las librerías numpy [12]

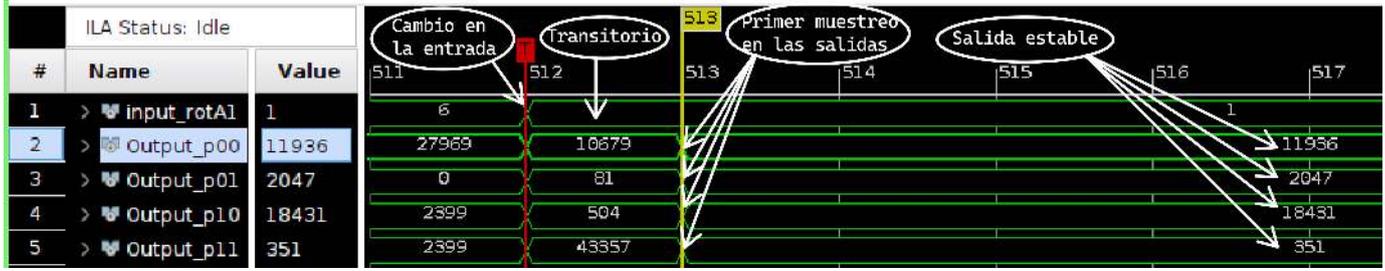


Fig. 4: Monitoreo de las salidas mediante el System ILA y disparando con una de las entradas.

y sympy [13]. Luego, utilizando la función `sympy.simplify` se busca la expresión equivalente más reducida que nos permita calcular p_{out} .

La expresión que se obtiene es la que finalmente se implementa en la FPGA, utilizando lenguaje VHDL. Los cuatro valores del vector de salida se calculan en paralelo y el resultado de los senos y cosenos involucrados, se los almacena en LUTs para acelerar la velocidad de cómputo y minimizar el uso de DSPs. Las sumas y multiplicaciones se ejecutan en punto fijo puesto que el máximo valor absoluto, tanto de las amplitudes como de las probabilidades, es 1.

El diagrama en bloques final se observa en la Fig. 3. El bloque Zynq representa al microprocesador dual-core Arm Cortex-A9 dentro del SoC desde donde se escriben las entradas y se leen las salidas. El bloque QGT_state_fixpt_0 representa el diseño en VHDL con las 6 entradas (φ_{A1} , φ_{A2} , φ_{A3} , φ_{B1} , φ_{B2} y φ_{B3}) y las 4 salidas (las columnas del vector p_{out}).

La comunicación entre la FPGA y el microprocesador se debe hacer utilizando el protocolo AXI (protocolo de Xilinx), para esto se utiliza el Comblock [14]. El bloque en VHDL escribe y lee los registros del Comblock y este se encarga de comunicarse con el microprocesador utilizando el protocolo AXI. Finalmente, el bloque System ILA (Integrated Logic Analyzer) es un analizador lógico útil para monitorear las señales internas.

IV. RESULTADOS Y CONCLUSIÓN

A modo comparativo, en nuestro laboratorio, hasta el momento, se simulan los circuitos cuánticos parametrizados utilizando la librería Qiskit de IBM para Python optimizada en C++ [15]. Si se calcula la salida para 262144 entradas distintas, utilizando una CPU con un microprocesador AMD Ryzen 7 5800U, y se promedian los tiempos de cálculo, obtenemos un valor de 2,894 ms por circuito.

En contraste, se utilizó una placa SoC FPGA Pynq-Z2 con un chip Xilinx ZYNQ XC7Z020. El porcentaje de LUT (12%) y de DSP (27%) utilizados indica que el diseño se podría extender a una mayor cantidad de qubits.

Luego, se verifica que, efectivamente, para distintos conjuntos de entradas se obtienen prácticamente las mismas salidas tanto en Python como en la FPGA. Como ejemplo, para un set de entradas $[\varphi_{A1}, \varphi_{A2}, \varphi_{A3}, \varphi_{B1}, \varphi_{B2}, \varphi_{B3}] = [\frac{\pi}{4}, \pi, \frac{\pi}{2}, \frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}]$, en Python se obtiene $p_{out/python} = [0.364277, 0.0625, 0.5625, 0.010723]_{p_flotante}^t$ y en la

FPGA $p_{out/FPGA} = [11936, 2047, 18431, 351]_{decimal}^t = [0.364258, 0.06247, 0.56247, 0.010712]_{p_fijo}^t$ (en la Fig. 4 se pueden observar las señales de salida medidas con el ILA en representación decimal).

Caben destacar algunas consideraciones tomadas a la hora de medir el tiempo de ejecución de la simulación en la FPGA. Lo primero que se intentó medir fue el tiempo que debe esperar el microprocesador desde que establece las entradas hasta que puede leer una señal estable a la salida (terminado el transitorio en la FPGA). El problema es que este tiempo es bastante menor al tiempo que pasa entre la escritura y la primera lectura de los registros de la FPGA desde el microprocesador (32 ciclos de reloj a 100 MHz aproximadamente).

Por lo tanto, se mide el tiempo de duración del algoritmo con la FPGA utilizando el ILA para monitorear las señales de salida del bloque en VHDL disparando con una señal de entrada. La frecuencia de muestreo del ILA es de 100MHz. Como se observa en la Fig. 4, el tiempo que pasa entre el establecimiento de las entradas (input_rotA1 cambia de 6 a 1 en la muestra 512) y que la salida es valida (las señales Output_p** tienen un transitorio pero se observa que en la muestra 513 ya están estables) es menor o igual a una muestra. Realizando esta medición para distintas variantes de entradas se llega a la conclusión de que el tiempo de ejecución es menor a $\frac{1}{100\text{MHz}} = 10\text{ns}$ (debido a limitaciones en aumentar la frecuencia de muestreo del ILA, no se puede determinar el valor exacto).

Finalmente, esta notable diferencia (de 2,9 ms a 10 ns) se debe a que en la FPGA se aprovechó al máximo la estructura fija del circuito cuántico y se realizaron todas las cuentas en paralelo consiguiendo un diseño puramente combinacional. Esta ventaja nos permitirá acelerar significativamente las simulaciones para facilitar la investigación y la aplicación tanto de la teoría de juegos cuántica como de los algoritmos cuánticos con circuitos parametrizados.

REFERENCIAS

- [1] John Preskill. Quantum computing 40 years later. *arXiv preprint arXiv:2106.10522*, 2021.
- [2] Yangyang Li, Mengzhuo Tian, Guangyuan Liu, Cheng Peng, and Licheng Jiao. Quantum optimization and quantum learning: A survey. *IEEE Access*, 8:23568–23593, 2020.
- [3] Michele Mosca. Cybersecurity in an era with quantum computers: Will we be ready? *IEEE Security Privacy*, 16(5):38–41, 2018.
- [4] Bela Bauer, Sergey Bravyi, and Mario Motta. Quantum algorithms for quantum chemistry and quantum materials science. *Chemical Reviews*, 120(22):12685–12717, 2020. PMID: 33090772.

- [5] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.
- [6] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.
- [7] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4):043001, 2019.
- [8] Faisal Shah Khan, Neal Solmeyer, Radhakrishnan Balu, and Travis S Humble. Quantum games: a review of the history, current state, and interpretation. *Quantum Information Processing*, 17(11):1–42, 2018.
- [9] Agustin Silva and Omar Gustavo Zabaleta. Fpga quantum computing emulator using high level design tools. In *2017 Eight Argentine Symposium and Conference on Embedded Systems (CASE)*, 2017.
- [10] Jakub Pilch and Jacek Długopolski. An fpga-based real quantum computer emulator. *Journal of Computational Electronics*, 18(1):329–342, 2019.
- [11] Jens Eisert, Martin Wilkens, and Maciej Lewenstein. Quantum games and quantum strategies. *Physical Review Letters*, 83(15):3077, 1999.
- [12] Charles R. Harris and K. Jarrod Millman. Array programming with NumPy. *Nature*, 585(7825), 2020.
- [13] Aaron Meurer, Christopher P. Smith, and Paprockin. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3, 2017.
- [14] ICTP Mlab and INTI. The core comblock, accessed on 5 March 2022.
- [15] Abby-Mitchell, Héctor Abraham, and Rochisha Agarwal. Qiskit: An open-source framework for quantum computing, 2021.

Stable type-2 fuzzy logic controller hardware implementation

Yazmin Maldonado and Nohe R. Cazarez-Castro*

Posgrado en Ciencias de la Ingeniería,

Departamento de Ingeniería Eléctrica y Electrónica,

Tecnológico Nacional de México/ Instituto Tecnológico de Tijuana

Tijuana, Mexico

<https://orcid.org/0000-0002-3793-7484> (Yazmin),

<https://orcid.org/0000-0001-6787-3947> (Nohe)

yaz.maldonado@tectijuana.edu.mx, nohe@ieee.org

Abstract—A stable type-2 fuzzy logic controller is implemented in hardware, specifically in an Artix-7 FPGA. For the implementation Vivado, Xilinx system for DSP and Matlab are used for experimental tests. The results are analyzed and compared with and without uncertainty against a stable type-1 fuzzy logic controller. According to the results, the stable type-2 fuzzy logic controller has a better performance as the uncertainty increases, positioning it as an excellent option for a wide variety of applications.

Index Terms—FPGA, type-2 fuzzy logic controller, uncertainty, speed control, stability.

I. INTRODUCTION

The type-1 fuzzy inference systems (T1-FIS) and the type-2 fuzzy inference systems (T2-FIS) in software implementation are useful for systems that do not require real time features, for example, systems that run off-line, like forecasting [1], classification [2], and others more.

The hardware implementation of T1-FIS and T2-FIS is a challenge, and only few works in the state-of-the-art have considered this challenge, for example, [3] [4].

T1-FIS and T2-FIS have been theoretically studied and applied for control [5] and robotics [6]. However, the T2-FIS under extreme conditions (uncertainty) have a better performance [7].

In [4], authors reports the FPGA (Field programmable gate array) implementation of Lyapunov closed-loop stable type-1 fuzzy logic controller (T1-SFLC), applied to the speed control of a direct current (DC) motor. The implementation was in a FPGA, Vivado, Xilinx system for DSP and Matlab. This kind of fuzzy control implementation allows to deal with low magnitude bounded uncertainty, and that is the reason of why in this work, based in [4], a stable type-2 fuzzy inference system (T2-SFIS) is designed in VHDL (an acronym of Very High Speed Integrated Circuit and Hardware Description Language) and implemented in hardware, specifically in a Basys-3 FPGA, in order to solve the speed regulation problem for a DC motor. Then, a stable type-2 fuzzy logic controller

(T2-SFLC) is implemented in Simulink and Xilinx system for DSP. For the implementation 8 bits are used for the data discretization. The results are analyzed and compared against different level of uncertainty, and compared with the results reported in [4].

The rest of the document is organized as follows. The problem in hand is stated in Section II, and the T2-SFLC designed to solve this problem is presented in Section III. The T2-SFIS implementation is presented in Section IV, while in Section V results are discussed. Finally, in Section VI conclusions are presented.

II. PROBLEM STATEMENT

A. Dynamic model of a DC motor

The dynamic model of a DC motor is given as

$$J\ddot{q}(t) + f\dot{q}(t) = u(t) + w(t), \quad (1)$$

where the inertia, viscous friction, acceleration and motor speed are denoted respectively as J , f , $\ddot{q}(t)$ and $\dot{q}(t)$. Signal $u(t)$ is the control action. External disturbance $w(t)$ is introduced in order to consider destabilizing non-linear phenomena and hard to model phenomena. For the particular case $J = 7.1 \times 10^{-6} \text{ kg} \cdot \text{m}^2$ and $f = 0.8 \text{ N} \cdot \text{m}$ estimations are taken from the motor data-sheet.

To formally state the problem, the deviations vector $x = [x_1, x_2]^T$ is introduced as

$$\begin{aligned} x_1 &= \dot{q}_d - \dot{q}(t), \\ x_2 &= -\ddot{q}(t), \end{aligned} \quad (2)$$

where x_1 is the speed error respect to the desired speed \dot{q}_d , and x_2 is the speed derivative.

Then, system (1) represented in terms of x takes the form $\dot{x} = [\dot{x}_1, \dot{x}_2]^T$. So, the objective is to design a stable fuzzy logic controller (SFLC) $u(t)$ in order to obtain a closed-loop system with bounded trajectories, and output $\dot{q}(t)$ tends to the desired speed \dot{q}_d , attenuating the influence of external disturbances $w(t)$.

Remark 1. It is important to empathize that (1)-(2) is not necessary for the approach proposed in this work, and is used just to obtain numerical results.

The research was partially funded by Tecnológico Nacional de México under grants 13326.21-P, 14492.22-P and 15171.22-P.

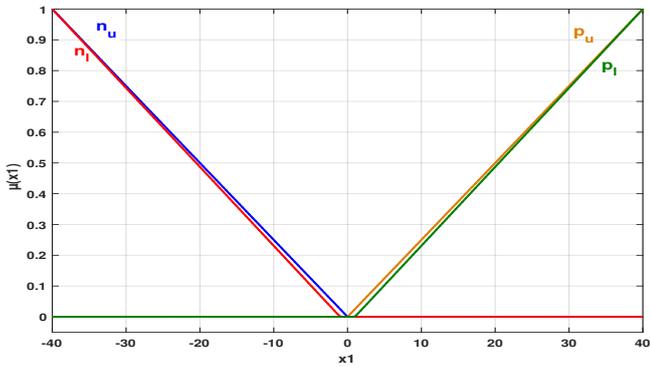


Fig. 1. Type-2 Membership functions for input variable x_1 . In this case n is negative and p is positive.

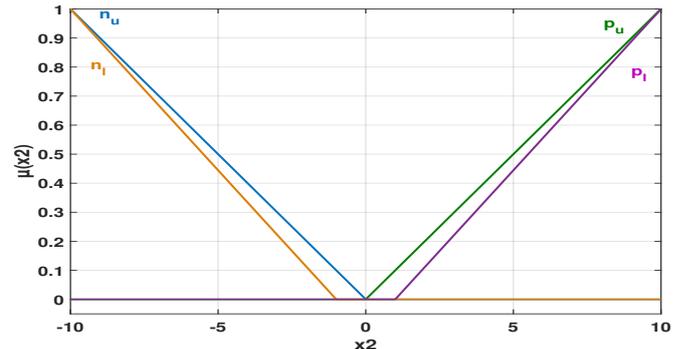


Fig. 2. Type-2 Membership functions for input variable x_2 . In this case n is negative and p is positive.

B. Considerations for implementation

The use of FPGA is attractive due to its speed in execution time, parallel processing, hardware reconfiguration, etc. On the other hand, fuzzy controllers are an excellent option to solve problems where uncertainty is present. In this work it is considered the implementation of the three main stages of a T2-SFIS: fuzzification, inference and defuzzification [3]. Also, Triangular Membership Functions (TMF) are considered for each membership function of each input and output variable. In all the cases, 8 bits are considered for the data discretization.

III. T2 STABLE FUZZY LOGIC CONTROLLER

Considering vector x available, the positive defined and radially unbounded candidate Lyapunov function

$$V(x_1, x_2) = \frac{1}{2} (x_1^2 + x_2^2) \tag{3}$$

is proposed, whose time derivative is

$$V(\dot{x}_1, \dot{x}_2) = x_1 \dot{x}_2 + x_2 \dot{x}_1, \tag{4}$$

and with the knowledge that in a mechanical system \dot{x}_2 is proportionally related to $u(t)$, (4) can be written as

$$V(\dot{x}_1, \dot{x}_2) = x_1 \dot{x}_2 + x_2 u(t), \tag{5}$$

and to guarantee (5) been at least semi-negative defined, the following fuzzy rule base is proposed from [4]:

- If x_1 is positive and x_2 is positive then u is negative big.
- If x_1 is positive and x_2 is negative then u is zero.
- If x_1 is negative and x_2 is positive then u is zero.
- If x_1 is negative and x_2 is negative then u is positive big.

So, inputs variables can be considered as shown in Fig. 1 and Fig. 2, and output variable can be considered as shown in Fig. 3. This design is inspired in [8] and [9], and therefore the origin of the closed-loop system is stable.

IV. T2-SFIS IMPLEMENTATION

The T2-SFIS is implemented in a Basys-3, Xilinx Artix-7 (XC7A35T-1CPG236C) with a 106 IOBs, 41,600 FFs and 20,800 LUTs. The T2-SFIS is designed in VHDL code using

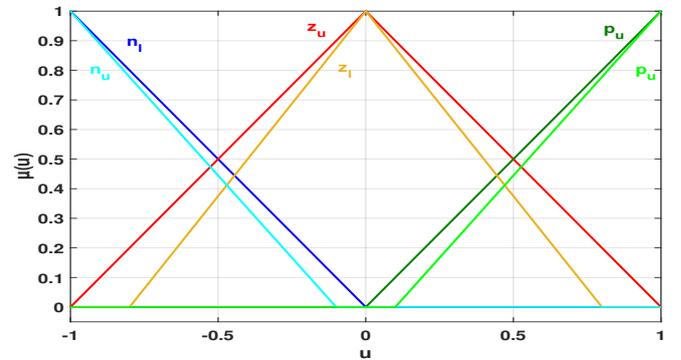


Fig. 3. Type-2 Membership functions for output variable u . In this case n is negative big, z is zero and p is positive big.

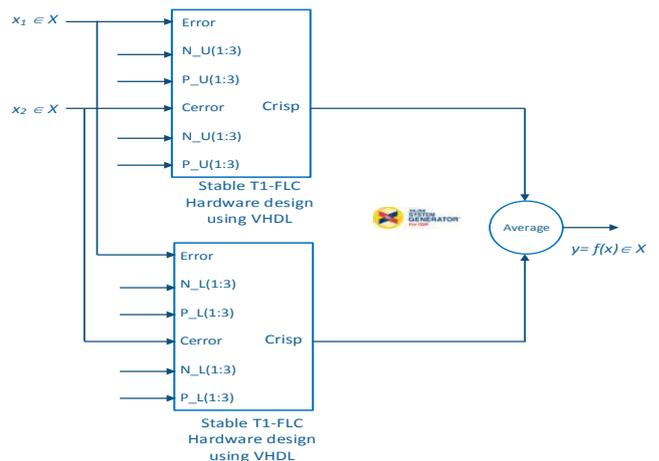


Fig. 4. T2 stable fuzzy inference system schematic in VHDL.

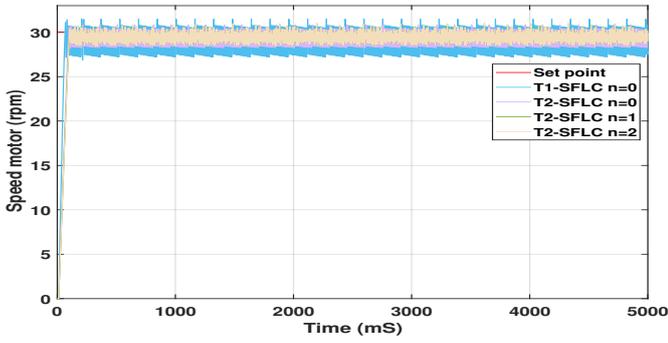


Fig. 6. Comparisons of the speed motor for different stable fuzzy logic controllers with and without uncertainty.

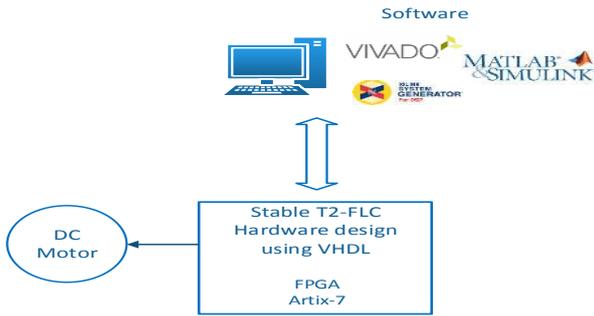


Fig. 5. Stable T2-fuzzy logic controller diagram.

TABLE I
COMPARISONS OF THE ERROR FOR DIFFERENT STABLE FUZZY LOGIC CONTROLLERS.

	ITAE	IAE	ISE
T1-SFLC n=0	12,360,000	5,919	31,640
T2-SFLC n=0	9,837,000	5,346	37,670
T2-SFLC n=1	9,784,000	5,348	37,730
T2-SFLC n=2	9,732,000	5,307	37,290

Vivado with the methodology used in [3]. The schematic diagram of the T2-FIS in VHDL is show in Fig. 4.

The T2-SFLC is designed in Xilinx system for DSP (Digital Signal Processing), Simulink and Matlab. The Fig. 5 shows the methodology for the stable T2-FLC.

To evaluate the performance of the T2-SFLC, experiments were carried out for a speed motor of 30 rpm and an uncertainty source was added. In addition, the results were compared against a T1-SFLC designed in [4] using different metrics, like Integral Time by Absolute Error (ITAE), Integral Absolute Error (IAE) and Integral Square Error (ISE).

V. RESULTS

For the experiments,

$$w(t) = n \sin(2\pi t) + \frac{1}{4} \cos(16\pi t), \quad (6)$$

where n is a constant and t is the sample time, is used for the addition of an uncertainty source to the T1-SFLC. $t = 5000$ is considered for all the experiments.

The T2-SFIS resources used in the FPGA are: 17.8% LUTs, 0.96% FFs and 22.6% IOBs, while the T1-SFIS uses half of the resources in the FPGA because we are using the methodology from [4].

With (6), varying the uncertainty source ($n = 0$, $n = 1$ and $n = 2$), for the controllers T1-SFLC and T2-SFLC, the simulation results are shown in Fig. 6. Also, from Fig. 6, a high-frequency phenomena is observed, this phenomena comes from the definition of the fuzzy logic controller.

Table I shows the performance of the controllers for motor speed regulation with and without uncertainty. It is important to note that for the T2-SFLC, the greater the uncertainty, the response is better.

It is important to note that, as stated in Section III, the closed-loop systems is stable in the sense of Lyapunov according with [8] and [9].

VI. CONCLUSIONS

In this work, a stable type-2 fuzzy logic controller is designed according to Lyapunov stability theory.

The speed regulation control problem for DC motor is addressed.

The stable type-2 fuzzy inference system is designed in VHDL code for FPGA implementation in a Basys-3 board.

Some experiments are carried out, in order to compare the performance of the controller design, results are compared with a stable type-1 fuzzy logic controller with and without uncertainty. The experimental results shown that the stable type-2 fuzzy logic controller is better when there is uncertainty in the source.

REFERENCES

- [1] J. d. J. Rubio, I. Elias, D. R. Cruz, J. Pacheco, G. J. Gutierrez, and A. Zacarias, "A fuzzy algorithm for the prediction of future data," *IEEE Latin America Transactions*, vol. 15, no. 8, pp. 1361–1367, 2017.
- [2] C. Murcia, G. Bonilla, and M. Melgarejo, "Fuzzy classifiers tuning through an adaptive memetic algorithm," *IEEE Latin America Transactions*, vol. 12, no. 2, pp. 197–204, March 2014.
- [3] Y. Maldonado, O. Castillo, and P. Melin, "Particle swarm optimization of interval type-2 fuzzy systems for fpga applications," *Applied Soft Computing*, vol. 13, no. 13, pp. 496–508, 2013.
- [4] Y. Maldonado and N. R. Cazarez-Castro, "Stable fuzzy controller hardware implementation," in *Proc. of the Congreso Argentino de Sistemas Embebidos CASE 2021*, vol. 1, University of Buenos Aires. Buenos Aires: University of Buenos Aires, November 2021, pp. 1–3.
- [5] E. R. M. Medina and D. R. Romero, "Speed control of a dc motor with a type-2 fuzzy logic controller subject to a large disturbance," *Computación y Sistemas*, vol. 22, no. 22, pp. 521–536, 2018.
- [6] A. S. Handayani, N. L. Husni, S. Nurmaini, I. Yani, and P. N. Sriwijay, "Application of type-1 and type-2 fuzzy logic controller for the real swarm robot," *International Journal of Online and Biomedical Engineering*, vol. 15, no. 06, pp. 83–98, 2019.
- [7] H. Moa, F.-Y. Wang, M. Zhou, R. Li, and Z. Xiao, "Footprint of uncertainty for type-2 fuzzy sets," *Information Sciences*, vol. 272, pp. 96–110, 2019.
- [8] O. Castillo, L. Aguilar, N. Cárdenas, and S. Cárdenas, "Systematic design of a stable type-2 fuzzy logic controller," *Applied Soft Computing*, vol. 8, no. 3, pp. 1274–1279, 2008, forging the Frontiers - - Soft Computing.
- [9] N. R. Cazarez-Castro, L. T. Aguilar, S. L. Cardenas-Maciel, C. A. Goribar-Jimenez, and M. Odreman-Vera, "Diseño de un controlador difuso mediante la síntesis difusa de lyapunov para la estabilización de un péndulo de rueda inercial," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 14, no. 2, pp. 133–140, 2017.

Adaptación de un stack de Aceleración por Hardware sobre FPGA

Pedro Ignacio Martos

Laboratorio de Automática y Robótica (LAR)
Universidad de Buenos Aires – Facultad de Ingeniería
pmartos@fi.uba.ar

Abstract— El stack Kria Robotic System (KRS) es una extensión de código abierto de la plataforma Robot Operating System (ROS) que permite implementar nodos (tareas) de ROS en hardware dedicado sobre dispositivos FPGA. La implementación de referencia de KRS se ejecuta sobre una tecnología SoM. En este trabajo se describe la adaptación del stack para poder ejecutarlo en una tecnología MPSoC, la cual es muy utilizada en el ámbito académico e industrial, por lo que esta adaptación permite ampliar la base de usuarios que pueden aprovechar la aceleración por hardware de nodos de ROS

Keywords—Aceleración por Hardware, FPGA, ROS, Robótica.

I. INTRODUCCION

Durante la última década se ha obtenido un progreso significativo en el área de la robótica, desde los algoritmos utilizados hasta las plataformas de hardware utilizadas. Distintos sistemas robóticos, tales como manipuladores, vehículos aéreos no tripulados, o pilotos automáticos vehiculares fueron diseñados para operaciones de búsqueda y rescate [1],[2], exploración [3],[4], entrega de paquetes [5], entretenimiento [6],[7] y otras aplicaciones y escenarios. No obstante, los sistemas robóticos son extremadamente complejos [8],[9],[10] e integran estrechamente distintas tecnologías y algoritmos como por ser sensado, percepción, mapeo, localización, toma de decisiones, control, etc. Esta complejidad genera muchos desafíos para el diseño de sistemas robóticos [11],[12].

Por otro lado, los sistemas robóticos necesitan procesar una enorme cantidad de datos en tiempo real. Los datos de entrada proveen de múltiples sensores y de origen heterogéneo, a su vez, los sistemas robóticos tienen recursos de hardware limitados (memoria, ancho de banda para transferencia de datos, capacidad de cómputo, etc), por lo que se dificulta cumplir requerimientos de tiempo real para tareas tales como sensado 3D, localización, navegación y planificación de camino.

Como respuesta a esta situación, se ha comprobado que las unidades de cómputo especializadas capaces de proveer aceleración por hardware permiten lograr mayor desempeño en sistemas robóticos [13]. Estas unidades de cómputo especializado en el caso de FPGAs se basan en hardware específico adaptado al algoritmo del sistema robótico, lo que permite obtener un tiempo de cómputo más rápido y una ejecución “en tiempo real”, es decir, logrando cumplir las metas de tiempo para cada tarea del sistema. La idea central en

la Aceleración por Hardware es combinar el enfoque tradicional centrado en el control usado en robótica, con el enfoque basado en procesamiento de datos, optimizando el uso de los recursos de hardware y mejorando el desempeño del sistema [14]. De esta manera surge el Hardware Definido por Software (HDS), refiriéndose a la implementación de una aplicación en una FPGA, lo que permite la creación de hardware reconfigurable en tiempo de ejecución mediante software. El HDS maximiza la eficiencia la eficiencia en tiempo de ejecución de los algoritmos y es una alternativa a la arquitectura Von Neumann de los dispositivos de cómputo. El HDS en robótica debe interpretarse como hardware del sistema que se reconfigura en tiempo de ejecución que puede ser reprogramado y adaptado mediante software. El desarrollo de software tradicional en robótica se relaciona a programar cierta funcionalidad en la CPU del robot con una arquitectura y restricciones predefinidas; esto dificulta la adaptación del sistema. Mediante la aceleración por hardware en FPGA, implementar comportamiento en el robot consiste en programar una arquitectura de hardware que resuelva las tareas necesarias. De esta manera se tiene la posibilidad de crear arquitecturas a partir del software e implementarlas en distintas plataformas de hardware.

Robot Operating System (ROS) [15] es el estándar de facto para el desarrollo de aplicaciones de robótica. ROS consiste en diferentes herramientas y librerías para construir y gestionar robots. Open Robotics rediseñó ROS en 2014, dando lugar a ROS2 para subsanar las limitaciones de la primera versión; es por ello que en lo sucesivo nos referiremos a ROS2 como “ROS”.

ROS trabaja con una abstracción llamada “grafo computacional”. En este grafo, cada nodo realiza algún tipo de cómputo e intercambia información con otros nodos a través de un bus común a todos implementado por una capa de comunicaciones. Las comunicaciones se organizan por “tópicos” y el comportamiento del robot queda determinado por el grafo computacional. Dicho de otra manera, la capa de comunicaciones captura la realidad física del robot, incluyendo los buses de comunicación, los componentes (sensores y actuadores) y el mapeo del grafo computacional a los dispositivos de cómputo disponibles en el robot.

Kria Robotics Stack (KRS) [16] es un conjunto de librerías y herramientas integrada con ROS2. Esta diseñado con el objetivo de lograr baja latencia (computo rápido), determinismo (previsibilidad), tiempo real (logro de metas de

temporización), seguridad y alta tasa de transferencia de información mediante la implementación de aceleración por hardware en FPGAs de la empresa Xilinx. La integración se hace a través de las herramientas de construcción de ROS2 (ament y colcon) permitiendo la extensión a otras tecnologías (p.ej. GPU). La generación de nodos se realiza directamente desde el espacio de trabajo de ROS2, automatizándose la compilación cruzada de código, la síntesis e implementación del acelerador. Estos aceleradores se generan sin interactuar con herramientas específicas de la tecnología que se utiliza. Los aceleradores generados se empaquetan en un espacio de trabajo que puede integrarse directamente al sistema robótico. La implementación de referencia de este stack se realiza sobre una tecnología de SoM [17], la cual es muy reciente, por lo que se decidió encarar la adaptación de este stack a la tecnología MPSoC [18] utilizando una placa ultra96v2 de Avnet [19], esta tecnología es muy utilizada en ámbitos académicos e industriales, permitiendo ampliar la base de usuarios de aplicaciones robóticas aceleradas por hardware usando dispositivos FPGA.

II. PROCESO DE ADAPTACIÓN

La adaptación del stack KRS implicó realizar modificaciones en las extensiones del sistema de construcción de ROS2 (colcon y ament) para que reconocieran la tecnología MPSoC como una opción de construcción; se realizó un diseño de hardware, de manera de obtener una descripción del hardware MPSoC disponible en la placa ultra96v2; se generó un sistema Linux compatible con la descripción de hardware del MPSoC y luego se extendió dicho sistema Linux agregándole ROS2 y los componentes de software necesarios para poder cargar aceleradores por hardware y ejecutar tareas de cómputo en el hardware de la FPGA. Esta adaptación se encapsula en una capa de firmware que incluye todos los elementos necesarios para implementar el acelerador como un nodo de ROS.

III. HARDWARE

La descripción de hardware se encuentra encapsulada en un archivo en formato XSA [20]. Este archivo es un contenedor que almacena toda la información necesaria para implementar una plataforma de cómputo y es usado por las herramientas de software para abstraer toda la información necesaria respecto de CPU (o CPUs en el caso de multiprocesadores), buses, bloques IP (tales como gestores de interrupciones), puertos y pines que comprenden el *Processing Subsystem* del dispositivo MPSoC. La descripción de hardware se realizó con la herramienta Vitis 2021.2 y comprende un multiprocesador Quadcore ARM Cortex A53; un gestor de señales de reloj, cuatro gestores de reinicialización, un controlador de interrupciones compatible con el bus AXI y un arbitro de bus AXI. Una guía para realizar este proceso se puede encontrar en [21].

IV. SOFTWARE

Asimismo se implementó un sistema Linux específico para el dispositivo MPSoC utilizando la herramienta Petalinux [22], la cual configura distintas opciones (de kernel, de

sistema de archivos, interfaces de entrada/salida, etc.) y genera los distintos scripts necesarios para la creación del sistema Linux. En particular se generan distintas “recetas” para la construcción de los distintos componentes y se utiliza la herramienta Yocto [23] para construir el sistema desde código fuente. Cabe aclarar que este proceso requiere una conexión a internet cableada y de alta velocidad (>50 MB), gran cantidad de espacio de almacenamiento (>70GB) y en una estación de trabajo con Intel i7 10ma generación y 32GB de RAM es un proceso que tarda de 6 a 12 horas dependiendo de la configuración establecida.

V. EJEMPLOS

Para ejecutar los ejemplos provistos con la implementación de referencia en la nueva plataforma fue necesario realizar ligeras modificaciones a los mismos. En dichos ejemplos la interfase para transferir datos entre el código que corre en el Processing Subsystem y el acelerador implementado en hardware se realiza mediante OpenCL[24] utilizando la técnica de Buffer Mapping[25]. En esta técnica se mapea una región de la memoria del acelerador a una zona de memoria del Processing Subsystem, obteniéndose un puntero a dicho mapeo y el componente en tiempo de ejecución de OpenCL se encarga de las transferencias entre ambas memorias. En el caso de los ejemplos fue necesario establecer una región de memoria mapeada más grande que el espacio ocupado por los datos procesados por el acelerador. Cuando la región de memoria mapeada ocupaba el mismo tamaño que los datos procesados, se obtenían inconsistencias en los resultados, por lo que se cree que hay un problema de implementación en el componente de software que en tiempo de ejecución copia los datos entre ambas memorias, y al usar una región mapeada más grande, estas inconsistencias no suceden.

VI. RESULTADOS EXPERIMENTALES

En la implementación de referencia en tecnología SoM los ejemplos generan nodos de ROS que emiten resultados con frecuencias de 2Hz, 6Hz y 10Hz. Dichos ejemplos (modificados según se describió más arriba) al ejecutarse en la plataforma MPSoC emiten resultados con frecuencias de 1.8Hz, 5.5Hz y 7Hz. Una posible explicación para la reducción en la frecuencia de emisión de resultados puede ser debida a que los ejemplos modificados utilizan mapeos de memoria mucho más grandes (4 veces mayores), por lo que los tiempos de transferencia serán mayores y explicarían la reducción en desempeño. Por otra parte, la tecnología SoM tiene mejor desempeño que la tecnología MPSoC, lo que también explica que esta última tenga un desempeño algo menor. En la Fig.1 se muestra gráficamente la comparación de desempeño entre la implementación de referencia en tecnología SoM y la adaptación en tecnología MPSoC para tres ejemplos (A, B y C) con distintos niveles de optimización.

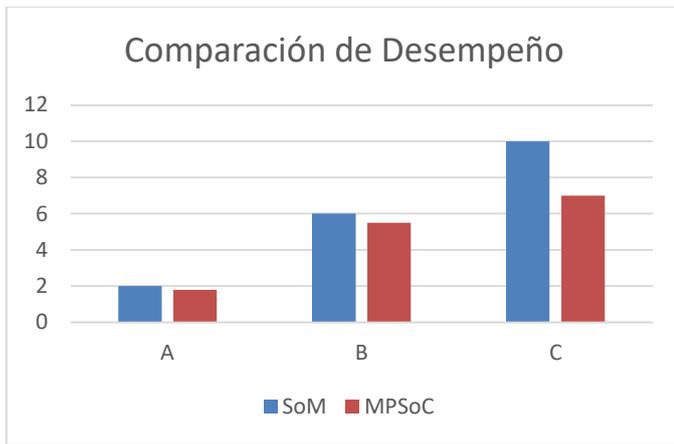


Fig.1 Comparacion de Desempeño

VII. CONCLUSIONES Y TRABAJO A FUTURO

Se realizó exitosamente la adaptación del stack KRS a la tecnología MPSoC utilizando la versión 2021.2 de las herramientas de implementación en FPGA (*Vitis* y *Vivado*) y la versión "Foxy" de ROS. Estas herramientas se ejecutan en el sistema operativo Ubuntu versión 20.04; por lo que es necesario verificar que esta adaptación también funciona correctamente en la versión 2022.1 de las herramientas de implementación en FPGA, utilizando la nueva versión de ROS (*Humble*) y el sistema operativo Ubuntu versión 22.04, estos últimos lanzados en Mayo de 2022 y con un tiempo de soporte hasta 2027, por lo que su validación aseguraría la compatibilidad del ecosistema por los próximos 5 años. Esta adaptación se encuentra disponible en [26] bajo una licencia de código abierto

AGRADECIMIENTOS

A Victor Mayoral Vilches, por su ayuda y sugerencias para este trabajo.

REFERENCIAS

- [1] A. Qiantori, A. B. Sutiono, H. Hariyanto, H. Suwa, and T. Ohta, "An emergency medical communications system by low altitude platform at the early stages of a natural disaster in Indonesia," *Journal of medical systems*, vol. 36, no. 1, pp. 41–52, 2012.
- [2] A. Ryan and J. K. Hedrick, "A mode-switching path planner for uav-assisted search and rescue," in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 1471–1476, IEEE, 2005.
- [3] N. Smolyanskiy, A. Kamenev, J. Smith, and S. Birchfield, "Toward lowflying autonomous mav trail navigation using deep neural networks for environmental awareness," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4241–4247, IEEE, 2017.
- [4] A. Giusti, J. Guzzi, D. C. Ciresan, F.-L. He, J. P. Rodríguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. Di Caro, et al., "A machine learning approach to visual perception of forest trails for mobile robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 661–667, 2015.
- [5] J. K. Stolaroff, C. Samaras, E. R. O'Neill, A. Lubers, A. S. Mitchell, and D. Ceperley, "Energy use and life cycle greenhouse gas emissions of

drones for commercial package delivery," *Nature communications*, vol. 9, no. 1, pp. 1–13, 2018.

- [6] S. J. Kim, Y. Jeong, S. Park, K. Ryu, and G. Oh, "A survey of drone use for entertainment and avr (augmented and virtual reality)," in *Augmented Reality and Virtual Reality*, pp. 339–352, Springer, 2018.
- [7] S. Jung, S. Cho, D. Lee, H. Lee, and D. H. Shim, "A direct visual servoing-based framework for the 2016 iros autonomous drone racing challenge," *Journal of Field Robotics*, vol. 35, no. 1, pp. 146–166, 2018.
- [8] S. Liu, L. Li, J. Tang, S. Wu, and J.-L. Gaudiot, "Creating autonomous vehicle systems," *Synthesis Lectures on Computer Science*, vol. 6, no. 1, pp. i–186, 2011.
- [9] S. Krishnan, Z. Wan, K. Bhardwaj, P. Whatmough, A. Faust, G.-Y. Wei, D. Brooks, and V. J. Reddi, "The sky is not the limit: A visual performance model for cyber-physical co-design in autonomous machines," *IEEE Computer Architecture Letters*, vol. 19, no. 1, pp. 38–42, 2020.
- [10] S. Krishnan, Z. Wan, K. Bhardwaj, P. Whatmough, A. Faust, S. Neuman, G.-Y. Wei, D. Brooks, and V. J. Reddi, "Machine learning-based automated design space exploration for autonomous aerial robots," *arXiv preprint arXiv:2102.02988*, 2021.
- [11] S. Liu and J.-L. Gaudiot, "Autonomous vehicles lite self-driving technologies should start small, go slow," *IEEE Spectrum*, vol. 57, no. 3, pp. 36–49, 2020.
- [12] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, and W. Shi, "Edge computing for autonomous driving: Opportunities and challenges," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1697–1716, 2019.
- [13] Z. Wan, B. Yu, T. Y. Li, J. Tang, Y. Zhu, Y. Wang, A. Raychowdhury, and S. Liu, "A survey of FPGAbased robotic computing," *IEEE Circuits and Systems Magazine*, vol. 21, no. 2, pp. 48–74, 2021.
- [14] V. Mayoral-Vilches and G. Corradi, "Adaptive computing in robotics, leveraging ROS 2 enabled software-defined hardware," *Xilinx*, WP537, 2021.
- [15] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.
- [16] V. Mayoral-Vilches "Kria Robotics Stack – A ROS2-centric approach for hardware acceleration in robotics" *Xilinx*, WP540, 2021.
- [17] (<https://www.xilinx.com/products/som/kria.html>) Obtenido en Junio de 2022.
- [18] (<https://www.xilinx.com/products/silicon-devices/soc/zynq-ultrascale-mpsoc.html>) Obtenido en Junio de 2022.
- [19] (<https://www.avnet.com/wps/portal/us/products/new-product-introductions/npi/aes-ultra96-v2/>) Obtenido en Junio de 2022.
- [20] ([https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/18841693/HSI+debugging+and+optimization+techniques?f=print#HSIdebuggingandoptimizationtechniques-TheHardwareDefinitionFile\(XSA\)%3A](https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/18841693/HSI+debugging+and+optimization+techniques?f=print#HSIdebuggingandoptimizationtechniques-TheHardwareDefinitionFile(XSA)%3A)) Obtenido en Junio de 2022.
- [21] (<https://highlevel-synthesis.com/2021/08/14/how-to-create-ultra96v2-linux-based-platform-in-xilinx-vitis-2020-2/>) Obtenido en Junio de 2022.
- [22] (<https://www.xilinx.com/products/design-tools/embedded-software/petalinux-sdk.html>) Obtenido en Junio de 2022.
- [23] (<https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/18841883/Yocto>) Obtenido en Junio de 2022.
- [24] (<https://www.khronos.org/opencv/>) Obtenido en Junio de 2022.
- [25] (<https://www.khronos.org/opencv/>) Obtenido en Junio de 2022.
- [26] (https://github.com/pimartos/acceleration_firmware_ultra96v2/) Obtenida en Junio de 2022.

Migración de una plataforma de hardware no certificada hacia una certificada

Luis David Diaz-Charris
*Departamento de Ciencias de la
 Computación y Electrónica
 Universidad de la Costa, CUC*
 Barranquilla, Colombia
 ldiaz28@cuc.edu.co

Martín N. Menéndez
*Laboratorio de Sistemas Embebidos,
 CONICET-GICSAFe, Facultad de
 Ingeniería*
Universidad de Buenos Aires
 Buenos Aires, Argentina
 mmenendez@fi.uba.ar

Santiago Germino
*Laboratorio de Sistemas Embebidos,
 CONICET-GICSAFe, Facultad de
 Ingeniería,*
Universidad de Buenos Aires
 Buenos Aires, Argentina
 sgermino@fi.uba.ar

Ariel Lutenberg
*Laboratorio de Sistemas Embebidos,
 CONICET-GICSAFe, Facultad de
 Ingeniería*
Universidad de Buenos Aires
 Buenos Aires, Argentina
 alutenb@fi.uba.ar

Ramiro Ghignone
*Facultad Regional Haedo
 Universidad Tecnológica Nacional*
 Buenos Aires, Argentina
 rghignone@frh.utn.edu.ar

Resumen— Este artículo describe el proceso que se llevó a cabo para la migración de un sistema de enclavamiento ferroviario electrónico de una plataforma de hardware no certificada hacia una certificada. El sistema de enclavamiento se generó utilizando el Generador Automático de Códigos (ACG) desarrollado por el Grupo de Investigación en Calidad y Seguridad de las Aplicaciones Ferroviarias (GICSAFe); y la plataforma certificada fue el Sistema de Protección de Reactor de IyC-CNEA desarrollado por la Comisión Nacional de Energía Atómica de Argentina.

Palabras clave— sistemas ferroviarios, sistema de enclavamiento, generación automática de código.

I. INTRODUCCIÓN

La función principal de un sistema de enclavamiento es realizar de forma segura la señalización y maniobra en un sistema de transporte ferroviario. Su relevancia subyace en que evita condiciones de peligro para la integridad física de las personas y bienes materiales, tales como: conflictos entre rutas, distancias no seguras entre formaciones y otras que podrían desembocar en colisiones y descarrilamiento de los trenes [1], [2], [3].

Para Argentina el sistema ferroviario es motivo de orgullo nacional y su cobertura operativa llega al 70% del país según el Ministerio de Transporte [4]. Además, según el reporte anual de la Comisión Nacional de Regulación del Transporte, para el año 2021 se movieron por la vía férrea más de 24 millones de toneladas de carga, de las cuales más del 60% corresponden a transporte de granos y subproductos agrarios [5]. Esto hace que el sistema de transporte ferroviario sea indispensable para la economía argentina.

A pesar de lo anterior, en Argentina la mayor parte de las vías férreas son operadas con sistemas de enclavamiento mecánicos o electromecánicos, los cuales actualmente están obsoletos en términos tecnológicos, aunque funcionales. Sin embargo, por esta misma razón son costosos de mantener y son poco escalables [1]. En contraste, los países europeos implementan el enclavamiento ferroviario de tipo electrónico [2], [6], los cuales están basados en sistemas embebidos y computadoras con características de software y hardware especiales para la aplicación en trenes.

Existen diversas tecnologías digitales aplicadas a implementar enclavamientos ferroviarios de tipo electrónico,

que son implementadas por empresas como Siemens, Alstom, Ansaldo y otras que emplean microprocesadores redundantes; otro ejemplo es el sistema descrito en [7], en el que se describieron los análisis de confiabilidad, modelos, cálculos de características de confiabilidad y la arquitectura de un sistema de seguridad del equipo de enclavamiento ferroviario basado en FPGA y tolerante a fallas, desarrollado para los ferrocarriles en República Checa. Todos estos sistemas pasan por procesos de desarrollo, control de calidad y certificaciones que deben responder a las normas internacionales que regulan el sistema ferroviario [8].

De acuerdo con lo anterior, en el presente artículo se presenta la transición de plataforma para el sistema de enclavamiento electrónico generado automáticamente con el software ACG descrito en [9], desarrollado por el grupo GICSAFe. Concretamente, se presenta el proceso de implementación del código VHDL generado por dicho software para la FPGA XC7Z010-1CLG400C de Xilinx, y su adaptación a la plataforma del Sistema de Protección de Reactor de IyC-CNEA desarrollado por la Comisión Nacional de Energía Atómica de Argentina (CNEA), que cuenta actualmente con certificaciones de hardware que la hacen una buena candidata en la ruta hacia una futura certificación de esta tecnología desarrollada en Argentina.

II. GENERADOR AUTOMÁTICO DE CÓDIGO

El sistema desarrollado en [9] incluye un analizador de código denominado RNA y un generador automático de código VHDL denominado ACG. A este sistema se le proporciona el plano de una red ferroviaria, lo analiza y genera automáticamente el enclavamiento y la señalización requerida para la maniobrabilidad de la red dada. Ese trabajo empleó herramientas automatizadas que toman el trazado de una red ferroviaria utilizando el enfoque topológico y realiza una implementación de hardware de señalización adecuada en el lenguaje de descripción de hardware VHDL. Estas herramientas de análisis de red ferroviaria y generación automática de código son usadas para realizar la implementación de un sistema de enclavamiento electrónico en una placa FPGA XC7Z010-1CLG400C de Xilinx.

El enfoque topológico adoptado en [9] para la generación de código se basó en reglas generales que se aplicaban a cada parte del sistema. La aplicación granular de dichas reglas permitió la construcción del enclavamiento a partir de la

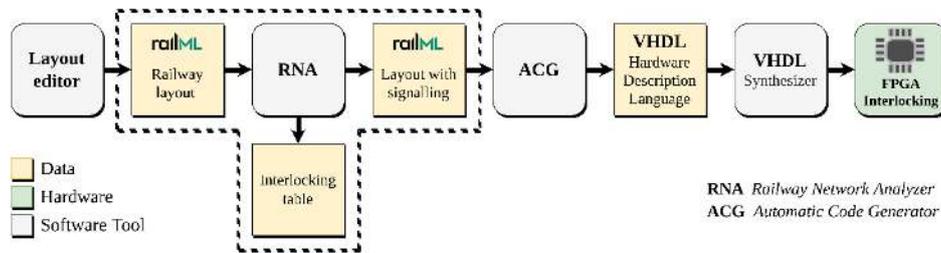


Fig. 1. Generación automática de Código propuesta en [9]

interconexión de todos los componentes de una red dada. Esta última característica favorece la adaptación a cualquier topología de red y a situaciones de topologías cambiantes como vías en renovación o nuevos proyectos. Un diagrama del desarrollo llevado a cabo en [9] se muestra en Fig. 1.

III. SISTEMA DE PROTECCIÓN DE REACTOR DE IYC-CNEA

El sistema de protección de reactor IyC-CNEA es un desarrollo de hardware llevado a cabo en Argentina por la Comisión Nacional de Energía Atómica. Este sistema fue diseñado, desarrollado y puesto en certificación para operar de forma segura las operaciones de los reactores nucleares RA-10 y CAREM [10], [11].

Este sistema está compuesto de diversos módulos de hardware, entre los cuales se encuentran módulos de procesamiento digital con FPGA para aplicaciones multipropósito. Estos módulos presentan diversidad en su composición, puesto que uno de los módulos cuenta con una FPGA Xilinx Spartan 6 XC6SLX9 y otro está basado en la FPGA ProASIC3 A3P1000-PQ208I. Lo cual, para fines de certificación en normas ferroviarias es un requisito indispensable [8]. En la Fig. 2. se puede observar el módulo basado en la FPGA Xilinx Spartan 6.

Una de las aplicaciones que han sido objeto de uso para estos módulos son la del desarrollo de un impulsímetro digital para ser utilizado en la instrumentación nuclear de las cadenas de arranque de los reactores RA-10 y CAREM [11], [12]

IV. CASO DE ESTUDIO

Para desarrollar lo propuesto se llevó a cabo la siguiente metodología:

A. Definición de la topología de red

Como se mencionó en la sección II, la entrada para la generación automática de código es la topología de red a implementar. Para efectos del presente trabajo se tomaron los planos de vía de la estación Belgrano C que se muestran en la Fig. 3. . Esta vía actualmente es operada por Trenes Argentinos.

B. Generación automática de código VHDL

Luego que se definió la topología de la red, se pasó al análisis de las vías usando el RNA. Este modelo se pasó después al generador de código automático VHDL. El código generado corresponde a la lógica que debe seguir el enclavamiento para la topología de red dada según las reglas planteadas en [8]. Cabe mencionar que la generación de código incluye dos módulos UART (_TX, transmisión y _RX, recepción) que permite realizar pruebas sobre el enclavamiento.

A. Síntesis del código VHDL

Luego de obtener el código VHDL que describe el



Fig. 2. Módulo de procesamiento digital basado en FPGA Spartan 6 XC6SLX9 del Sistema de Protección de Reactor de IyC-CNEA.

enclavamiento, se procede a sintetizar y generar el *bitstream* necesario para programar la FPGA de IyC-CNEA. Se realizó el cambio de plataforma respecto al trabajo [9], para lo cual, se debió crear el archivo de configuración de entradas/salidas con la configuración del módulo FPGA de la CNEA. Esto debido a que la implementación original estaba configurada para la placa de desarrollo Arty-z7 de Xilinx.

B. Prueba del enclavamiento electrónico

Luego de sintetizar de código VHDL, se procedió a programar la FPGA de IyC-CNEA y se corrieron pruebas de ejecución usando un módulo UART incorporado. La comunicación por UART se realiza enviando tramas de datos en formato serie desde la computadora; estas tramas son procesadas por el módulo del enclavamiento y luego recibidas nuevas tramas con información procesada según la lógica del enclavamiento (ver Fig. 4.). Los paquetes serie contienen el estado de circuitos de vía (ocupados/desocupados), máquinas de cambio, semáforos y barreras. Así el enclavamiento procesa estos datos de forma concurrente y genera las señales de peligro, gestiona los cambios de vía y bloquea o desbloquea los pasos a nivel según la información que va recibiendo.

En la Tabla 5 se puede observar también, la generación automática de la tabla de enclavamientos, que permite visualizar cuáles son las rutas válidas que podría tomar un tren a lo largo de la estación, así como el estado de cada uno de los elementos del señalamiento para cada ruta.

V. CONCLUSIONES

El caso de estudio presentado puede tomarse como referente para el uso de herramientas digitales disponibles en la literatura científica, como es el caso del generador de código automático propuesto en [9]. Este trabajo es una muestra del potencial de estas herramientas para facilitar el análisis y diseño de nuevos sistemas de enclavamiento.

Replicar el uso del generador de código antes mencionado y su aplicación en un hardware diferente permitió validar su eficacia, al mismo tiempo que demostrar su flexibilidad.

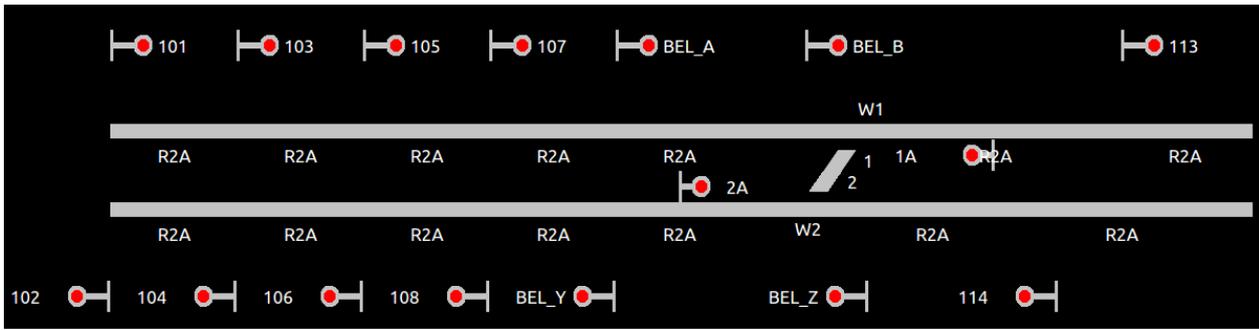


Fig. 3. Planos de vía de la estación Belgrano C, Buenos Aires, Argentina.

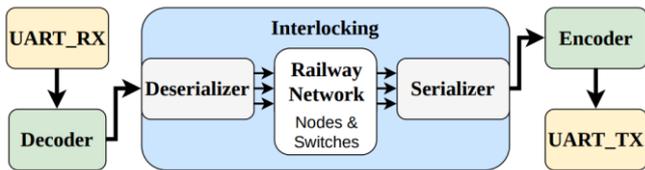


Fig. 4. Modelo de simulación, entrada y salida por UART. [9].

TABLE I. TABLA DE ENCLAVAMIENTOS

Ruta	Inicial	Final	Secuencia	Sentido
1	6	1	6-5-4-3-2-1	<
2	6	9	6-14-13-12-11-10-9	<
3	6	8	6-7-8	>
4	14	9	14-13-12-11-10-9	<
5	14	16	14-15-16	>
6	14	8	14-6-7-8	>

Además, el presente trabajo representa un aporte en el camino de la certificación, del desarrollo del grupo de investigación, ya que se integra de la diversidad de hardware, un requisito importante a la hora de certificar una tecnología según los estándares de las normas ferroviarias.

En el futuro se espera desarrollar nuevas funcionalidades en la búsqueda de generar diversidad en hardware a través de la implementación de estos mismos sistemas en distintas plataformas, aumentando de esta manera la robustez de esta aplicación y su acercamiento en el cumplimiento de las normas.

AGRADECIMIENTOS

Se extienden los más sinceros agradecimientos a Trenes Argentinos por su valiosa disposición y colaboración con compartir la información requerida para el desarrollo de este trabajo. Se agradece al Grupo de Investigación Calidad y Seguridad de las Aplicaciones Ferroviarias (GICSAFe), por el apoyo en conocimientos y el compañerismo.

REFERENCES

[1] R. A. Ghignone *et al.*, “Modelling, Simulation and Code Generation for Electronic Railway Interlocking Systems,” *IEEE Latin America Transactions*, vol. 19, no. 1, pp. 155–162, 2021, doi: 10.1109/TLA.2021.9423859.

[2] Y. Sun, “Study on Application of Regional Computer Interlocking in High-Speed Railway,” in *2019 4th International Conference on Electromechanical Control Technology and Transportation*, 2019, pp. 301–306. doi: 10.1109/ICECTT.2019.00076.

[3] Jörn Pachl, “Railway Signalling Principles,” p. 83, 2020, [Online]. Available: https://www.researchgate.net/publication/341277248_Railway_Signalling_Principles

[4] T. Argentinos, “Lineas ferroviarias.” [Online]. <https://www.argentina.gob.ar/transporte/trenes-argentinos-cargas/lineas> (accessed Jun. 22, 2022).

[5] Comisión Nacional de Regulación del Transporte, “Estadísticas del transporte ferroviario.” <https://www.argentina.gob.ar/transporte/cnrt/estadisticas-ferroviarias>

[6] Alstoms, “Alstoms Smartlock equipment at the heart of modular signalling project in the UK | Alstom.” <https://www.alstom.com/press-releases-news/2011/3/Alstom-Smartlock> (accessed Jun. 23, 2022).

[7] R. Dobias and H. Kubatova, “FPGA based design of the railway’s interlocking equipments,” in *Euromicro Symposium on Digital System Design, 2004. DSD 2004.*, 2004, pp. 467–473. doi: 10.1109/DSD.2004.1333312.

[8] CENELEC, “Norma EN-50128.” www.aenor.es (accessed Jun. 22, 2022).

[9] M. N. Menéndez, S. Germino, F. S. Larosa, and A. Lutenberg, “Automatic generation of VHDL code for a railway interlocking system,” *International Journal of Embedded Systems*, vol. 14, no. 6, pp. 544–552, 2021, doi: 10.1504/IJES.2021.121088.

[10] Ministerio de Economía, “Reactor argentino CAREM | Argentina.gob.ar.” <https://www.argentina.gob.ar/cnea/carem> (accessed Jul. 03, 2022).

[11] C. Varrastro *et al.*, “Sistema de control alternativo para el accionamiento de parada de emergencia con inyección de boro en la central nuclear,” *IAEA TECDOC Series*, vol. IAEA-TECDO, no. August, pp. 26–49, 2016, [Online]. Available: https://www-pub.iaea.org/MTCD/Publications/PDF/TE-1765_web.pdf

[12] IAEA, *Uso de Nuevas Tecnologías para el Diseño, Modernización y Automatización de Sistemas y Procesos Relacionados con Aplicaciones Nucleares*, no. IAEA-TECDOC-1765/S. Austria, 2016.

Categoría

FORO TECNOLÓGICO

Implementación de Sistemas Embebidos

Área Temática

Sistema de ensayo de relés ferroviarios de seguridad

G. Santamarina
 CONICET - GICSAFe
 Laboratorio de Sistemas Embebidos
 Facultad de Ingeniería - UBA
 Buenos Aires, Argentina
 gpsantamarina@gmail.com

A. Laiuppa
 CONICET - GICSAFe
 Departamento de Electrónica
 UTN - FRBB
 Bahía Blanca, Argentina
 alaiuppa@frbb.utn.edu.ar

G. Ramoscelli
 CONICET - GICSAFe
 Depto. de Ing. Eléctrica
 y de Computadoras - UNS
 Bahía Blanca, Argentina
 ramoscel@uns.edu.ar

A. Lutenberg
 CONICET - GICSAFe
 Laboratorio de Sistemas Embebidos
 Facultad de Ingeniería - UBA
 Buenos Aires, Argentina
 alutenb@fi.uba.ar

Resumen—En este artículo se describe el diseño y desarrollo de un probador de relés de uso crítico en sistemas ferroviarios. Estos relés son extremadamente costosos dado que deben asegurar niveles de fiabilidad y seguridad elevados. Contar con un sistema de ensayos que ayude a la validación de estos relés podría impulsar el surgimiento de una industria nacional, lo que permitiría reducir los costos de mantenimiento de la infraestructura ferroviaria Argentina. El probador consta de un diseño de hardware y de software para el control electrónico del relé bajo prueba, así como de una aplicación web que permite su operación de manera remota.

Palabras clave—ensayos electromecánicos, desarrollo de firmware industrial, redundancia en hardware, relés ferroviarios, monitoreo remoto

I. INTRODUCCIÓN

A. Contexto y motivación

Los dispositivos electromecánicos que conforman la infraestructura ferroviaria, tal como los relés de seguridad, deben cumplir con altos estándares de fiabilidad. La normativa europea UNE-EN 50578 [1] describe el procedimiento para la verificación de este tipo de relés. Como se podrá prever, dada la complejidad de los procedimientos allí detallados, la fabricación de estos dispositivos se vuelve extremadamente costosa, por lo que el diseño de un equipo que permita realizar la verificación automatizada de relés fabricados de manera nacional, podría reducir drásticamente los costos de mantenimiento de la infraestructura ferroviaria. En [2] se presenta el diseño de un equipo con estas capacidades y su posterior implementación se detalla en este trabajo.

Si bien, debido a la nula disponibilidad en el mercado de equipos probadores de relés para certificación, es difícil hacer una evaluación de nuestro diseño en relación al de otros probadores, en la tabla I se muestra un análisis comparativo con equipos utilizados para el ensayo de relés en campo [3] [4] [5].

TABLA I
 COMPARACIÓN CON EL ESTADO DEL ARTE.

Marca - Modelo	Entradas analógicas	Tensión variable	Corriente máxima [A]	Interfaz
Omicron CMC 356	10	Sí	128	SCADA
Omicron CMC 256plus	10	Sí	75	SCADA
Megger SMRT 46D	8	Sí	30	Display
Trabajo realizado	18	Sí	30	Interfaz web

II. ARQUITECTURA PROPUESTA

A. Descripción general

En la figura 1 se muestra un diagrama del sistema con cada uno de sus componentes. Para el control y monitoreo del dispositivo bajo pruebas se trabajó en el desarrollo de distintos circuitos de potencia que posibilitan el manejo de las corrientes elevadas necesarias para las pruebas con carga, además de agregar varios canales para la conversión analógica-digital de las señales de interés.

La lógica de los ensayos se implementó en firmware y corre en redundancia en tres placas de desarrollo CIAA-NXP [6], conectadas a Internet a través del router de la figura 1. Dicha redundancia es necesaria para detectar errores en la secuencia de conmutación del relé y aumentar así la disponibilidad del probador. Esto es de suma importancia debido a que la norma requiere ensayos con diez millones de ciclos de apertura y cierre, y es deseable que estos ensayos se realicen de forma ininterrumpida [1].

Finalmente, para el manejo del equipo, se utiliza una interfaz web que permite la configuración de los parámetros específicos de cada ensayo y la visualización de manera tabulada de los datos registrados. Esta aplicación web se describirá en detalle en la sección II-D.

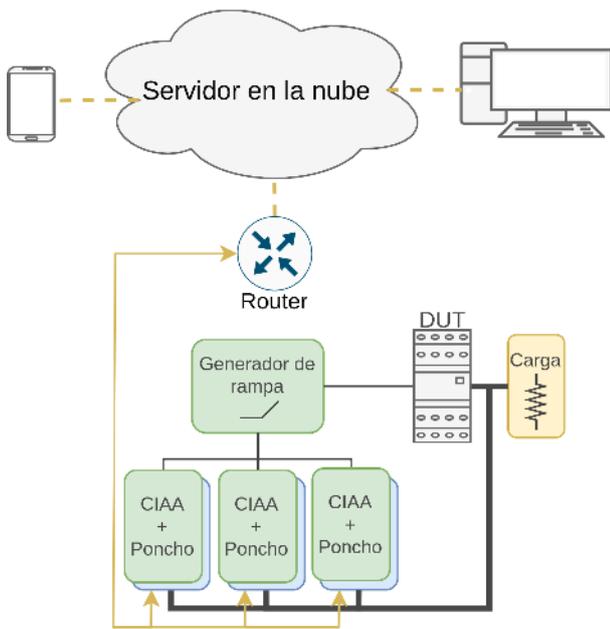


Fig. 1. Diagrama en bloques del probador de relés.

B. Hardware del probador

En la figura 1 se observa un diagrama en bloques del equipo probador de relés donde se incluyen todos sus subsistemas de hardware. Cuenta con un generador de rampa que amplifica la salida de un DAC y aplica esta tensión a la bobina del relé bajo prueba. En la misma placa del generador se incorpora un circuito acondicionador de señales, para adaptar la tensión y la corriente medidas en la bobina, durante los ensayos de rampa.

Para el control de los ensayos se utilizan tres placas de desarrollo CIAA-NXP, cada una con un circuito tipo “poncho” similar a un shield de Arduino [7]. En este circuito se implementa una máquina de estados finitos (o FSM por sus siglas en inglés) que genera la señal de control para la conmutación del relé durante los ensayos de ciclado, además de incorporar múltiples canales para la conversión analógica-digital de las señales de interés.

C. Desarrollo de firmware

El firmware, que se encuentra explicado en detalle en [8], es el encargado del control de las interfaces de hardware y de permitir la operación del probador a través de la aplicación web. En la figura 2 se presentan las capas de la arquitectura implementada. La capa base corresponde al hardware, es decir, la CIAA-NXP junto con el poncho y las etapas de potencia del probador.

Por encima de esta capa se ubican las librerías del fabricante, que permiten abstraerse de los registros del microcontrolador mediante la implementación de funciones sencillas para el manejo de los periféricos [9]. Sin embargo, la aplicación no hace un uso directo de estas librerías, sino que además se incluye una capa de abstracción de hardware [10] (HAL, por

sus siglas en inglés). Además de la HAL, se incluyó el Sistema Operativo de Tiempo Real (RTOS, por sus siglas en inglés) FreeRTOS [11], que permite dividir el diseño de la aplicación en hilos concurrentes.

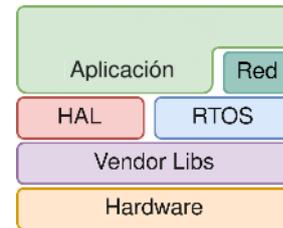


Fig. 2. Capas de abstracción del diseño de firmware.

Finalmente, en la capa de aplicación se implementa la lógica necesaria para la ejecución y el control de los ensayos. La misma se dividió en bloques funcionales para facilitar su diseño. Se puede ver un diagrama de esta división en la figura 3. Cada bloque se encarga de un conjunto de tareas específicas y se comunica con los demás bloques para concretar dichas tareas.

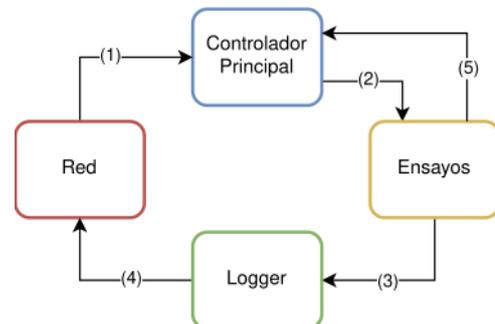


Fig. 3. Diagrama en bloques de la aplicación.

A continuación se detalla el funcionamiento del sistema a nivel de bloques. El Controlador Principal recibe los comandos del servidor a través del módulo de Red (1), actualiza el estado general del sistema y comunica al módulo de ensayos en caso de haberse solicitado la ejecución de una prueba (2). Luego, el bloque de Ensayos ejecuta las pruebas pertinentes y transmite los resultados al Logger (3). El Logger recibe los datos obtenidos de los ensayos, los almacena de forma local a modo de backup y los retransmite al módulo de Red (4) para su posterior envío al servidor. Cuando finaliza con los ensayos pendientes, el bloque de Ensayos notifica al Controlador Principal (5).

D. Servidor

El servidor se implementó como un conjunto de servicios en docker [12] configurados mediante docker-compose [13]. Se compone de un broker MQTT Mosquitto [14] para la comunicación con el equipo, una base de datos relacional hecha en postgres [15] para almacenar los datos recabados y una instancia de Node-red [16] para orquestar todo el conjunto e implementar la interfaz web. Además, incorpora un servidor

Nginx [17] que funciona como proxy inverso para el cifrado TLS, junto con un script para el manejo de los certificados, y un broker MQTT adicional para la intercomunicación de los servicios.

La interfaz web cuenta con un panel para monitorear el estado general del probador y tres paneles individuales para el disparo de los ensayos. En la figura 4 se muestra el panel de control que corresponde al ensayo de rampa.

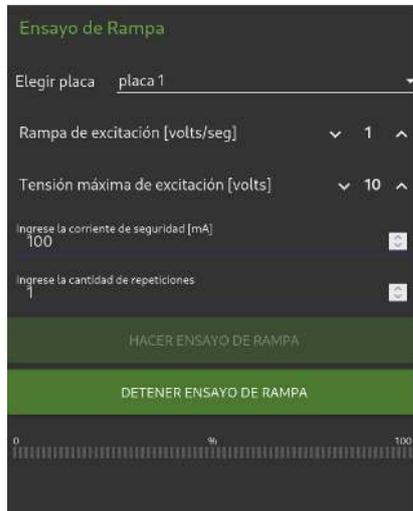


Fig. 4. Panel de control para el disparo de ensayos de rampa.

III. RESULTADOS

En la figura 5 se puede ver una foto del probador de relés implementado. Su funcionamiento se probó mediante ensayos sobre un relé de uso ferroviario. De esta manera se obtuvieron curvas acordes a lo esperado y se comprobó el correcto funcionamiento del sistema desarrollado. En la figura 6 se muestra a modo de ejemplo un gráfico de las mediciones obtenidas durante un ensayo de rampa.

IV. CONCLUSIONES

El diseño realizado permite efectuar en forma satisfactoria los ensayos indicados en la normativa que se aplica a relés de uso ferroviario. Actualmente se encuentran operativos dos probadores idénticos que fueron entregados a Trenes Argentinos y se está trabajando en el diseño de un equipo para el ensayo de relés en campo.

REFERENCIAS

- [1] Aplicaciones ferroviarias - Relés de señalización de corriente continua. Normativa. Normalización Española, 2013.
- [2] A. Laiuppa, M. Amado, J. M. Cruz, F. Larrosa, E. Irrazabal, I. Sambrana, S. H. Gallina, M. A. Gomez Lopez, A. Lutenberg, "Sistema automático para ensayos de ciclo de vida de relés ferroviarios de seguridad, Libro de Trabajos del CASE2018, Congreso Argentino de Sistemas Embebidos, Cordoba, Argentina, 2018.
- [3] Omicron CMC 356, (2022). [Online]. Disponible: <https://www.omicronenergy.com/es/productos/cmc-356/>
- [4] Omicron CMC 256plus, (2022). [Online]. Disponible: <https://www.omicronenergy.com/es/productos/cmc-256plus/>
- [5] Megger SMRT 46D, (2022). [Online]. Disponible: <https://us.megger.com/multi-phase-relay-tester-smrt46d-1>

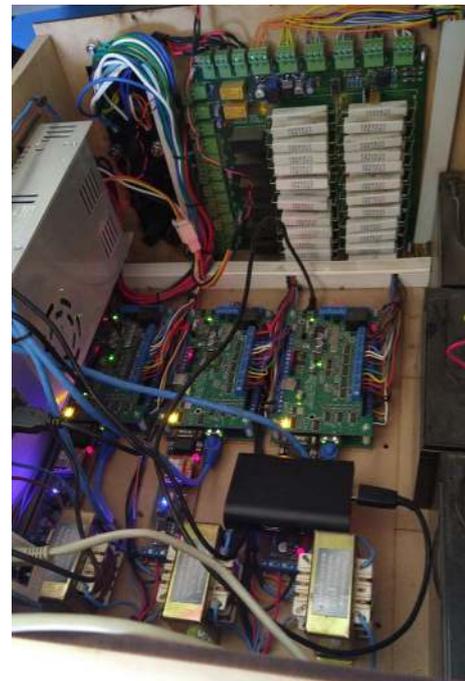


Fig. 5. Fotografía del probador de relés implementado.

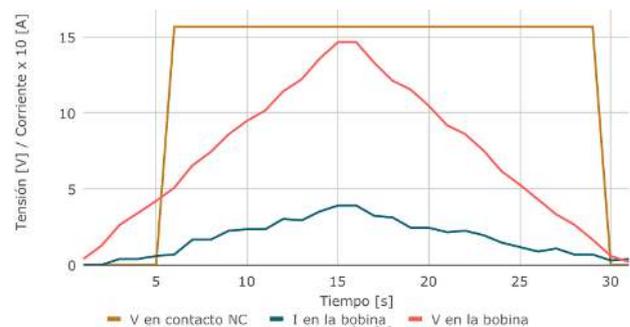


Fig. 6. Gráfico de tensión y corriente en la bobina, obtenido a partir de un ensayo de rampa .

- [6] Proyecto CIAA, (2021). [Online]. Disponible: <http://www.proyecto-ciaa.com.ar>
- [7] Arduino shield, (2022). [Online]. Disponible: <https://arduino.cl/que-es-un-shield/>
- [8] G. Santamarina, A. Laiuppa, G. Ramoscelli, A. Lutenberg, "Desarrollo de firmware para un sistema de ensayo de relés ferroviarios de seguridad", Libro de Trabajos del CASE2021, Congreso Argentino de Sistemas Embebidos, Argentina, 2021.
- [9] LPCOpen, (2022). [Online]. Disponible: <https://www.nxp.com/design/microcontrollers-developer-resources/lpcopen-libraries-and-examples>
- [10] sAPI, sAPI library for microcontrollers, (2022). [Online]. Disponible: https://github.com/epernia/firmware_v3/tree/master/libs/sapi/
- [11] FreeRTOS, (2022). [Online]. Disponible: <https://www.freertos.org/>
- [12] Docker, (2022). [Online]. Disponible: <https://www.docker.com/>
- [13] Docker Compose, (2022). [Online]. Disponible: <https://docs.docker.com/compose/>
- [14] Docker, (2022). [Online]. Disponible: <https://www.docker.com/>
- [15] PostgreSQL, (2022). [Online]. Disponible: <https://www.postgresql.org/>
- [16] Node-red, (2022). [Online]. Disponible: <https://nodered.org/>
- [17] Nginx, (2022). [Online]. Disponible: <https://www.nginx.com/>

Sistema de Comunicaciones Acústicas Subacuáticas aplicando SDUA, a partir de un SDR

Pablo Martin, Celeste Cebedio y David Petruzzi

ICYTE, Depto. de Electrónica y Computación, Facultad de Ingeniería - UNMDP

Mar del Plata, 7600, Argentina

pabfedmartin@gmail.com, {petruzzi, celestecebedio}@fi.mdp.edu.ar

Resumen—Este trabajo aborda el diseño y la implementación de un equipo que permite, utilizando dispositivos de radio definido por software comerciales, obtener una versión adecuada para acústica subacuática.

Se presenta el diseño de un Conversor de frecuencias dual: Ascendente (Up-converter) y Descendente (Down-converter). Este conversor permita obtener un dispositivo de acústicas subacuática definida por software, a partir de dispositivos comerciales de bajo costo utilizados en radio frecuencia.

Por último, se evalúa el desempeño del sistema completo, en un entorno de laboratorio

Palabras Clave—comunicación, acústica, subacuática, radio definida por software, SDR, SDUA, Up-Down Converter.

I. INTRODUCCIÓN

Los sistemas de comunicación cognitiva presentan un paradigma novedoso y atractivo para el futuro [1]. La idea principal de este enfoque es dotar al sistema de algún tipo de inteligencia que le permita responder a los cambios en el canal, adaptando sus características de operación. Dentro de estos tipos de sistemas se pueden encontrar los dispositivos de radio definidos por software (SDR) basados en el paradigma de radio cognitiva (RC) [2] o los dispositivos de acústica subacuática definidos por software (SDUA) basados en radio acústica cognitiva (CAR).

Los SDR son plataformas de comunicación predominantes en aplicaciones de RF, debido a la necesidad de soluciones de radio flexibles y reconfigurables, junto con su capacidad para seguir la rápida evolución de las tecnologías [3]. Estos dispositivos, que operan en la banda de frecuencia de UHF (300 a 3000 MHz) [4], pueden encontrarse comercialmente a un costo relativamente razonable. En cambio, en el ámbito de los SDUA, el rango de frecuencias de interés abarcan desde las bajas frecuencias hasta completar la banda de LF (0 a 300 kHz) y los desarrollos habitualmente se realizan a medida, resultando muy costosos.

Lamentablemente, el aprovechamiento del bajo costo y la gran versatilidad que los SDRs proveen, se ve impedido por la diferencia en frecuencia entre las señales utilizadas en RF y las de LF empleadas para comunicaciones subacuáticas. Se presenta entonces, como una solución a este impedimento, el diseño y la evaluación de un equipo dual: Up/Down-Converter, para que, sumado a un SDR comercial, permita procesar señales de LF.

Hasta la realización de este trabajo, no existen comercialmente dispositivos que realicen la conversión necesaria de forma directa. Hay equipos capaces de realizar la

función de elevar la frecuencia de trabajo, pero quedaría por resolver la función de Down-Converter. Dentro de las posibilidades analizadas, se encuentra un dispositivo de Analog Devices [5] que se asemeja a lo requerido, pero presenta múltiples limitaciones de adaptación a esta aplicación.

El aporte principal de este trabajo radica en la posibilidad de aprovechar estos SDRs comerciales en entornos subacuáticos, creando oportunidades para la investigación de nuevos sistemas de comunicación, adaptados a la dinámica compleja del ambiente acústico subacuático.

II. DISEÑO

En la Fig. 1, se presenta el diagrama del sistema completo.

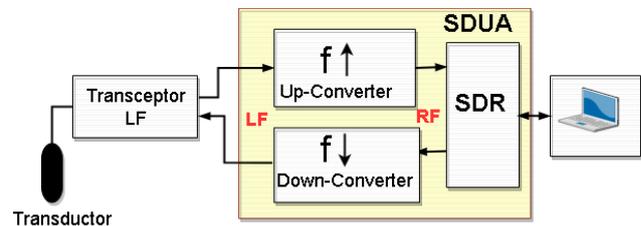


Fig. 1: Diagrama en bloques del Sistema.

El conjunto Up/Down Converter debe realizar dos funciones:

- Convertir las señales de baja frecuencia provenientes del equipo de comunicaciones subacuáticas en señales de alta frecuencia, aceptadas por el SDR. Esta función la realiza el sub-sistema “Up-Converter”.
- Convertir las señales de alta frecuencia provenientes del SDR en señales de baja frecuencia, aceptadas por el equipo de comunicaciones subacuáticas. Esta función la realiza el sub-sistema “Down-Converter”.

El rango de RF consiste en una banda de 600 kHz, centrada alrededor de 400 MHz, para ser así compatible con la mayoría de los SDRs comerciales.

II-A. Requerimientos de Diseño

En primer lugar, se determinan las especificaciones que el equipo propuesto debe cumplir. Debe considerarse que estas conversiones en frecuencia han de ser realizadas minimizando la distorsión, el ruido y las distorsiones de fase y frecuencia de las señales involucradas.

Para ello, se necesita conocer las características de los dispositivos que se conectan en cascada (Sistema LF y

SDR) y, de esta manera, establecer los valores límites para los parámetros característicos del circuito Up-Down a diseñar. Se asume que el sistema LF tiene los parámetros de un equipo de “buena” calidad [6] y se establecen los parámetros de los dispositivos SDR comerciales con un criterio conservador. En función de esto, en la Tabla I se detallan los valores esperados para los parámetros de los bloques Up-Converter y Down-Converter.

Parámetro	Up-Converter	Down-Converter	Comentarios
NF _{MAX}	10 dB	10 dB	
F _{MAX}	10 dB	10 dB	
IP _{3MIN}	0 dBm (in)	2.14 dBm (out)	Referido a in/out
PC _{1dBMIN}	-10 dBm (in)	-6.3 dBm (out)	Referido a in/out

Tabla I: Requerimientos de diseño del Up-Down Converter. IP₃: pto. de intercepción de 3er orden, PC_{1dB}: pto. de compresión 1dB, NF : cifra de ruido, F: factor de ruido,

II-B. Conversión Ascendente y Descendente

La Fig. 2 muestra el diagrama en bloques del sub-sistema Up-Down Converter.

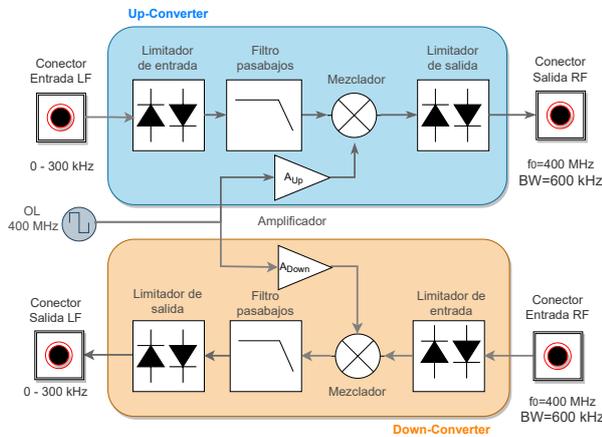


Fig. 2: Diagrama en bloques del Up-Down Converter

Etapas incluidas en el diseño:

- **Limitadores de entrada:** Provee protección al resto del sistema contra niveles de potencia excesivos. Se implementa con diodos PN, ya que esta es la solución más simple y económica.
- **Filtros pasa-bajos (LPF):** Encargado de rechazar la frecuencia imagen y las señales fuera de la banda de interés. Para su diseño, se tiene en cuenta que las señales utilizadas normalmente en comunicaciones subacuáticas son FSK y PSK y que las señales interferentes en esta banda son de AM comercial. Por ello se elige un filtro de Bessel, que minimiza el retardo de grupo [7] ya que es el parámetro que más degrada la calidad de la comunicación. Se calcula el mínimo orden en base a la máxima potencia de salida permitida para los transmisores de AM.
- **Mezcladores:** Traslada la señal de LF proveniente del sistema de acústica subacuática a una señal de RF requerida por el SDR en el caso del Up-Converter y viceversa para el caso opuesto. Por costo y simplicidad, se utilizan mezcladores tipo doble balanceado, basado en el empleo de diodos.
- **Oscilador local (OL):** Genera la señal requerida por el SDR. Es compartido por el Up y el Down-Converter, con una derivación para cada sub-sistema.

- **Amplificadores de OL:** El mezclador requiere que el OL le entregue un nivel de señal mínimo y el dispositivo elegido como OL brinda una potencia menor, por ello, la señal proveniente de este último necesita ser amplificada. Las etapas Down y Up, necesitan una adaptación de impedancias diferente, para evitar una solución circuital compleja, se resuelve que el amplificador no sea común.
- **Limitadores de salida:** Protegen al SDR y al transceptor contra niveles de potencia superiores a los máximos tolerable. Dado que el Up-Converter necesita ser capaz de manejar señales de UHF, se descarta una solución mediante diodos PN y se utiliza la misma topología basada en diodos PIN.
- **Fuente de alimentación:** Se incluye una fuente de alimentación interna, tomando potencia de un puerto de bus serie universal (USB) a través de un conector USB tipo C. Se incorporan protecciones contra sobre-corriente, sobre-tensiones transitorias y polaridad invertida.

III. ESPECIFICACIONES DE DISEÑO

Las especificaciones se presentan en la Tabla II.

Parámetro	Up-Converter	Down-Converter
Frec. Entrada	0-300 kHz	399.7-400.3 MHz
Frec. Salida	399.7-400.3 MHz	0-300 KHz
G _{MAX}	-8.22 dB	-8.28 dB
CL _{MAX}	8.22 dB	8.28 dB
NF _{MAX}	8.99 dB	8.54 dB
F _{MAX}	7.76	7.14
IP _{3MIN} (In)	8.73 dBm	13.81 dBm
IP _{3MIN} (Out)	0.51 dBm	5.53 dBm
PC _{1dBMIN} (In)	-2.81 dBm	3.53 dBm
PC _{1dBMIN} (Out)	-6.41 dBm	-5.75 dBm
P _{inMAX}	27 dBm	33.01 dBm
P _{outMAX}	0 dBm	10 dBm
Z In/Out	50 Ω	50 Ω
P _{MAX} Sistema	528 mW	

Tabla II: Especificaciones de diseño del Up-Down Converter. G: ganancia de potencia, CL: pérdida de conversión, NF_{MAX} : cifra de ruido, P_{in}: potencia de entrada, P_{out}: potencia de salida, y Z: impedancia

Al comparar las especificaciones de diseño con los requerimientos de la Tabla I, se comprueba que el diseño es satisfactorio.

IV. RESULTADOS OBTENIDOS

El dispositivo Up-Down Converter terminado y su gabinete se muestran en la Fig. 3. Sobre este dispositivo, se efectúan las siguientes mediciones: a) Ganancia. b) Distorsión por intermodulación. c) Compresión de la ganancia.



Fig. 3: PCB con componentes y vistas paneles gabinete.

La Tabla III muestra los resultados obtenidos, verificando que éstos coinciden con los planteados en el diseño.

Parámetro	Up-Converter	Down-Converter
G	-5.6 ± 1.1 dB	-8.8 ± 1.0 dB
CL	5.6 ± 1.1 dB	8.8 ± 1.0 dB
IP ₃ (In)	4.94 ± 0.05 dBm	sin medición
IP ₃ (Out)	1.21 ± 0.05 dBm	sin medición
P _{1dB} (In)	1.1 ± 0.4 dBm	4.2 ± 1.4 dBm
P _{1dB} (Out)	-7 ± 0.7 dBm	-5.4 ± 1.1 dBm
NF (1ª etapa)	0 dB	0.08 dB

Tabla III: Parámetros medidos del Up-Down Converter . IP₃ no se mide en el Down-Converter por limitaciones en el equipamiento

Se observa que la ganancia medida es mayor que el valor mínimo calculado, es decir, se está en un caso más favorable. Por otro lado hay una diferencia apreciable entre el IP₃(In) calculado y el medido (alrededor de 4dB). Esto podría deberse a la falta de especificación sobre la variación de IP₃ en el mezclador y la aproximación realizada (por falta de datos) de IP₃ al valor de IP_{1dB} en algunos bloques del diseño. Esta aproximación es válida si la distorsión por intermodulación predominante es de tercer orden, pero pudo observarse que existe distorsión apreciable causada por productos de otros órdenes también. A pesar de ello, el valor medido de IP₃ cumple con los requerimientos establecidos.

Comparando las Tablas II y III, se evidencia que la ganancia de potencia, la pérdida de conversión y el punto de compresión de 1dB, coinciden con lo esperado. Los resultados expuestos también denotan que la etapa de entrada no agrega una cifra de ruido considerablemente alta.

V. OPERACIÓN EN ENTORNO DE LABORATORIO

Para las pruebas del sistema final, se conecta el Up/Down Converter con un LimeSDR Mini [8], programado usando el software GNU Radio [9].

El SDR se configura para generar una señal con una portadora de alta frecuencia (400 MHz) modulada por una de baja frecuencia (100 Hz). Esta señal se envía al Down-Converter, cuya salida debe ser una señal de baja frecuencia semejante a la modulante. La salida del Down-Converter se visualiza en un osciloscopio, y re-ingresa al Up-Converter. El Up-Converter convierte esta señal a una de RF, que se envía a la entrada del SDR, donde es finalmente demodulada en amplitud.

En la Fig. 4 se observa el banco de prueba propuesto y en la Fig. 5 las señales obtenidas en el proceso. Se observa que dichas señales son las esperadas, salvo por una mínima distorsión.

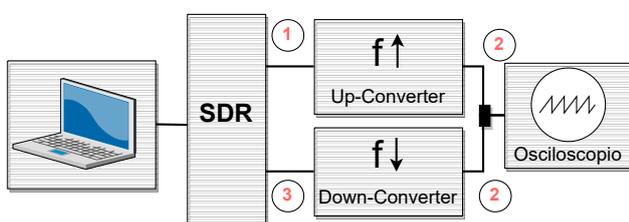


Fig. 4: Banco de prueba.

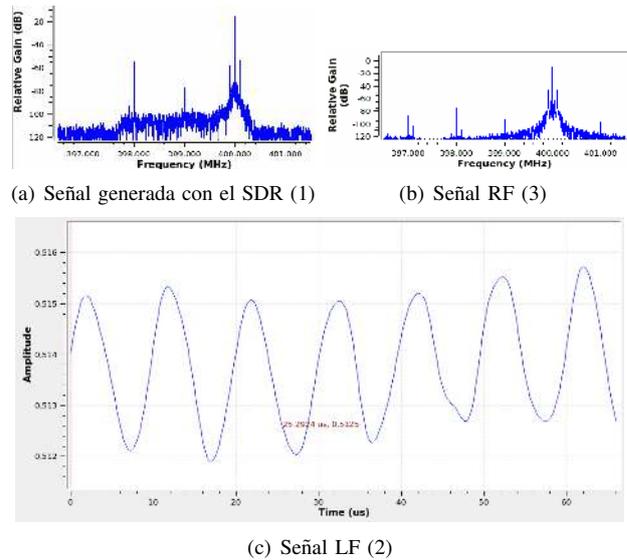


Fig. 5: Imágenes extraídas, a partir del software GNU Radio, durante la prueba en laboratorio

Esta demostración valida la utilización del conjunto Up/Down-Converter en la aplicación propuesta. Se puede observar que es posible generar una señal de alta frecuencia, convertirla a baja frecuencia, volver a convertirla en alta frecuencia y recuperar la señal modulante original.

VI. CONCLUSIONES Y TRABAJOS FUTUROS

Este trabajo mostró la posibilidad de diseñar y construir un dispositivo de bajo costo y portátil, gracias a su bajo consumo, que permite aprovechar el uso de un SDR comercial con señales acústicas subacuáticas.

Se demuestra que es posible aprovechar las ventajas inherentes de los SDR, en nuevos sistemas de comunicación. La posibilidad de niveles más altos de abstracción, aumenta la independencia y permite al usuario concentrar sus esfuerzos en aplicaciones determinadas.

Como trabajo futuro, se plantea utilizar el dispositivo en mediciones de campo y evaluar el desempeño con respecto a equipos de medición homologados. Para estas aplicaciones será necesario desarrollar sistemas de procesamiento de señales adecuados, recomendándose hacerlo con grafos de flujo realizados en GNU Radio.

REFERENCIAS

- [1] W. Yonggang, T. Jiansheng, P. Yue, and H. Li, "Underwater communication goes cognitive," in *OCEANS 2008*, 2008, pp. 1–4.
- [2] "IEEE standard definitions and concepts for dynamic spectrum access: Terminology relating to emerging wireless networks, system functionality, and spectrum management," *IEEE Std 1900.1-2008*, pp. 1–62, 2008.
- [3] G. E. Galioto and et al, "Flumo:flexible underwater modem," in *OCEANS 2019 - Marseille*, 2019, pp. 1–7.
- [4] A. Csete and S. Christiansen, "Evaluation of sdr boards and tool-chains -final report," European Space Agency (ESA) and Libre Space Foundation, Tech. Rep. V1, 2020, (SDR Makerspace).
- [5] A. Devices. Ad-frqvt1-ebz functional overview [analog devices wiki]. Accedido: 2021-06-10. [Online]. Available: <https://acortar.link/g3nRCr>
- [6] U. Rohde and J. Whitaker, *Communications Receivers: DSP, Software Radios, and Design*. McGraw-Hall, 2004, isbn : 9780071361217.
- [7] P. Wilson, *The Circuit Designers Companion*. Newnes, 2017, isbn : 9780081017647.
- [8] L. microsystems. Limesdr mini. Accedido: 2020-12-10. [Online]. Available: <https://limemicro.com/products/boards/limesdr-mini/>
- [9] G. R. Community. Gnu radio. Ver. 3.8.1.0. 2016. [Online]. Available: <https://www.gnuradio.org/about/>

Optimización de la detección de rayos X para microrradiografías de muestras de madera utilizando un sensor de imagen CMOS COTS

Damián Leonel Corzi ^{*†}, Jose Lipovetzky ^{*†‡}, Martin Perez ^{*†}, Fabricio Alcalde Bessia ^{†‡}, Miguel Sofo Haro ^{†‡}, Alejandro Martinez-Meier [§], Guillermina Dalla-Salda [§], Anne Sophie Sergent [§], Mariano Gómez Berisso ^{*†‡}

^{*}Instituto Balseiro, Universidad Nacional de Cuyo (UNCUYO)

[†]Comisión Nacional de Energía Atómica (CNEA)

[‡]Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)

[§]Estación Experimental Bariloche, Instituto Nacional de Tecnología Agropecuaria (INTA)

Resumen—Este trabajo describe la optimización de las condiciones de adquisición de imágenes radiográficas utilizando un sensor de imagen CMOS como detector directo y un tubo de rayos X de micro-foco de tungsteno de tensión de ánodo y corriente configurables. Se estudia la respuesta del sensor en función de la ganancia interna y se estudia la dependencia del contraste obtenido variando dicha ganancia y el potencial del tubo, que modifica el espectro de rayos-X, con miras a su uso futuro en un sistema embebido de micro-radiografías para muestras forestales.

Index Terms—Sensor MT9M001, Arducam, Imager CMOS, Radiografía, Rayos X, Alta Resolución.

I. INTRODUCCIÓN

Los sensores de imagen CMOS comerciales diseñados originalmente para la detección de luz visible son además sensibles a los rayos-X [1] [2], lo cual permite su utilización para capturar imágenes radiográficas de alta resolución [3]. Esto permite su uso en nuevos campos de aplicación, por ejemplo en áreas relacionadas con el estudio de materiales y los ensayos no destructivos. Entre estas áreas se encuentra el estudio de anillos de muestras forestales, que permiten estudiar el crecimiento de la planta a lo largo del tiempo correlacionando los resultados con información climática y por ejemplo estudiar efectos del calentamiento global. [4]

Dentro de los sensores utilizados por el grupo de trabajo, destaca particularmente el MT9M001C12STM, el cual fue previamente empleado para espectroscopia de rayos X [5] e imágenes radiográficas con neutrones [6], y de la que se estudió la tolerancia a los efectos producidos por dosis total [7].

Este trabajo estudia las condiciones óptimas para la adquisición de imágenes radiográficas de alta resolución de muestras de madera empleando el sensor antes mencionado. En la siguiente sección se presentan características básicas del sensor y del arreglo experimental. Luego, se muestran los resultados obtenidos para distintas condiciones de irradiación empleando diferentes ganancias del sensor. Finalmente se discuten futuros trabajos y las conclusiones obtenidas.

II. CONFIGURACIÓN EXPERIMENTAL

El arreglo experimental está formado por tres componentes, como puede observarse en el esquema de la Fig. 1. En primer lugar tenemos el tubo de rayos X, luego la muestra a analizar que atenúa la intensidad del haz, y por último el detector encargado de capturar una imagen radiográfica a partir de la proyección de la muestra sobre su superficie.

Los rayos X utilizados para la irradiación fueron generados por un tubo de tungsteno con un foco nominal de unos $8 \mu\text{m}$. El mismo puede operar en un rango de energías comprendido entre 10-100kV y una corriente máxima de 0.3 mA. El espectro generado es continuo debido al aporte mayoritario de la radiación de frenado o Bremsstrahlung [8].

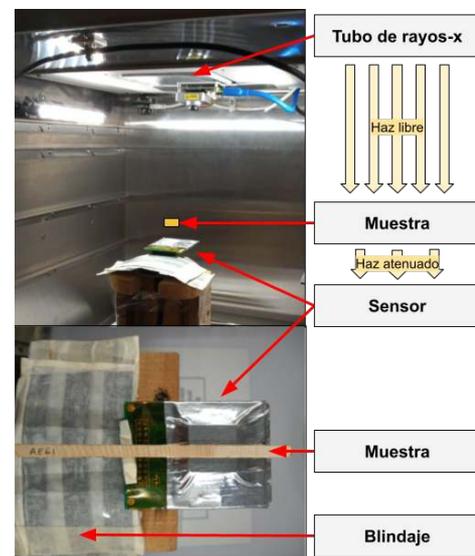


Figura 1. Fotografía del setup experimental empleado durante la irradiación.

El dispositivo empleado para la captura de imágenes es un sensor comercial MT9M001C12STM de la empresa Micron, el cual tiene un área activa de $6.66 \times 5.32 \text{ mm}^2$, con 1280×1024 píxeles de $5.2 \times 5.2 \mu\text{m}^2$, full well capacity de

42 ke- [7], relación señal-ruido 45 dB y rango dinámico de 68.2 dB. Cuenta con 10-bit de resolución de su ADC y escaneo es progresivo, también conocido en inglés como rolling shutter”.

Para la configuración y la lectura del sensor se utilizó la plataforma ArduCAM UC-425 y un adaptador desarrollado por miembros del grupo de trabajo, lo cual puede observarse en la Fig. 2. Este adaptador permite colocar blindaje sobre la electrónica de lectura para reducir los efectos por dosis total sin impedir la exposición del sensor. Para evitar de que fotones de luz visible alcancen el área activa del sensor, el mismo fue recubierto con una delgada lamina de aluminio.



Figura 2. Fotografías de la plataforma ArduCAM UC-425, modificada para permitir la irradiación del sensor. En la imagen de la izquierda se puede observar el sensor MT9M001C12STM, mientras que en la imagen de la derecha se encuentra recubierto con una lamina de aluminio.

III. ADQUISICIÓN Y ANÁLISIS DE DATOS

La adquisición y el análisis de los datos se realizó mediante programas desarrollados en Python debido a la portabilidad del código, lo que permite su utilización en diferentes plataformas.

Para reducir la varianza en la señal de cada píxel es necesario incrementar la cantidad de fotones detectados por el sensor, utilizando un tiempo de exposición incluso más largo que el configurable permitido. Por ello se toman numerosas imágenes y promedian las lecturas de cada píxel. Por otro lado la imagen resultante presenta un offset asociado a los circuitos de lectura empleados en el sensor. Este offset puede eliminarse al restar de la imagen original una segunda capturada con el sensor completamente oscurecido, la cual se conoce como imagen de oscuridad.

Luego de eliminar el efecto del offset, el primer análisis realizado consistió en obtener el valor medio de la intensidad de la imagen para distintas condiciones de irradiación, entre ellas variaciones en la potencia de alimentación del tubo de rayos-X, diferentes espesores del material a radiografiar y distintas ganancias del sensor. A modo de ejemplo, se puede observar en la Fig. 3 una imagen obtenida para una condición en particular interponiendo ninguno o dos espesores diferentes de acetato de celulosa en tres regiones del sensor.

La intensidad de la lectura en cada región pueden visualizarse en la gráficas de la Fig. 4. De este análisis se puede

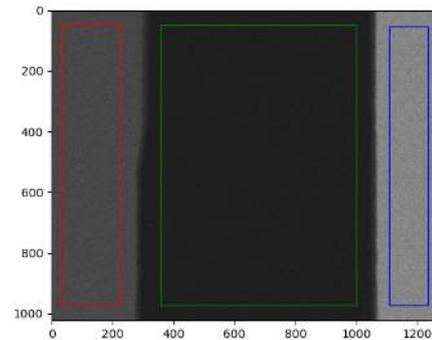


Figura 3. Imagen obtenida al irradiar una muestra de celulosa con una ganancia configurada en 32, tensión de ánodo de 30 kV y una corriente de 40 μ A. En el recuadro de color rojo se observa una zona de espesor constante de 1 mm y en el recuadro verde de 4 mm. El recuadro azul determina una zona libre de material y por lo tanto se observa la intensidad del haz libre.

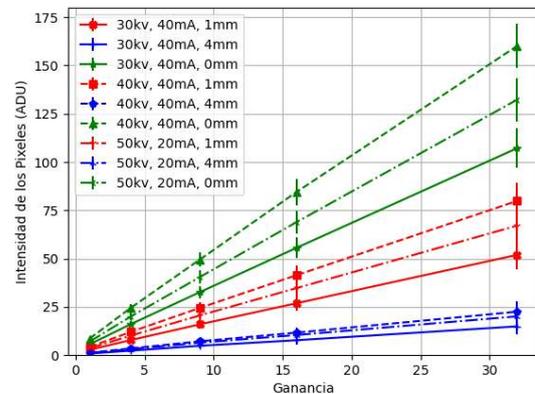


Figura 4. Intensidad media de la imagen para distintas condiciones de irradiación y su varianza (líneas verticales).

observar como la media crece de forma lineal con la ganancia configurada.

Como la intensidad del haz podría no ser uniforme en la superficie del sensor, mayormente debido a efectos geométricos o por la presencia de partículas en la superficie, se debe realizar una nueva corrección utilizando una imagen capturada con el sensor irradiado por el haz sin la presencia de la muestra a radiografiar. Esta corrección es conocida como *flat-field* o de campo plano [9], e implica dividir cada píxel de la imagen original con su correspondiente de la nueva captura para normalizar las intensidades obtenidas.

El valor medio de la intensidad de la imagen corregida corresponde directamente a la transmisión de la muestra radiografiada, siendo la misma el complemento de la atenuación de los fotones que interactúan con el material. Los datos obtenidos de este análisis pueden observarse en la Fig. 5, en la cual se aprecia que el ruido presente permanece constante para distintas ganancias del sensor, lo que permite inferir que el mismo está principalmente compuesto por el *shot-noise* [9].

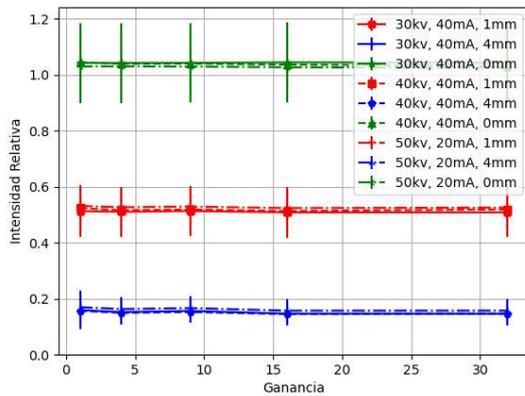


Figura 5. Intensidad relativa de la imagen luego de la corrección de campo plano o flat-field. El valor de la intensidad corresponde directamente con la transmisión de la muestra radiografiada.

Adicionalmente es posible determinar una medida de la resolución del sensor mediante la cantidad de píxeles que corresponden a una longitud conocida, obteniendo valores cercanos al tamaño del píxel definido en la hoja de datos del fabricante.

La gran resolución de las imágenes capturadas y la posibilidad de obtener el valor de la transmisión de la intensidad permiten utilizar los resultados obtenidos en ensayos no destructivos. En la Fig. 6 se puede observar una radiografía tomada a una muestra de madera de 2 mm de espesor y la transmisión a lo largo del eje indicado. Como el espesor de la muestra es constante, la variación de la transmisión corresponde a diferencias en la densidad de la madera debido a los anillos formados por la planta durante su crecimiento.

IV. CONCLUSIONES Y TRABAJO FUTURO

Se pudo concluir que es posible obtener radiografías de alta resolución empleando un sensor de imagen CMOS comercial modelo MT9M001C12STM y un tubo de rayos-X de microfoco de potencial y corrientes variables. Además, es posible obtener la transición asociada a distintos espesores de materiales utilizados como atenuadores.

Como la máxima superficie que es posible radiografiar corresponde al tamaño del área activa del sensor, para obtener radiografías de muestras de un mayor tamaño, se debe desplazar a las mismas sobre la superficie del sensor mientras se realizan capturas parciales. Todas las capturas parciales obtenidas deben ser procesadas y unidas para formar una única imagen final.

REFERENCIAS

[1] J. Lipovetzky, A. Cicuttin, M. L. Crespo, M. S. Haro, F. A. Bessia, M. Pérez, and M. G. Berisso, "Multi-spectral x-ray transmission imaging using a BSI CMOS image sensor," *Radiation Physics and Chemistry*, vol. 167, p. 108244, 2020.

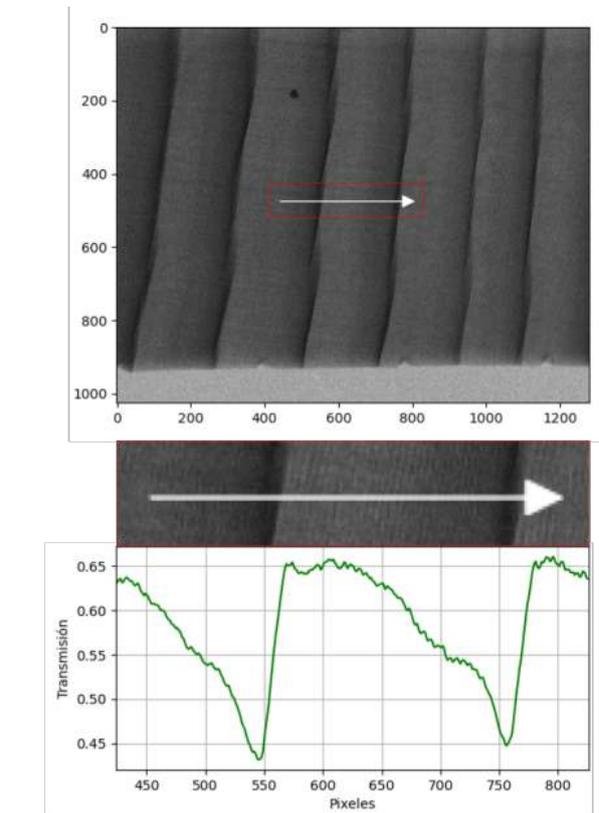


Figura 6. Radiografía de un listón de madera de espesor constante. En la parte inferior se puede observar el valor de la transmisión a lo largo del eje especificado por la flecha de color blanco.

- [2] M. Pérez, M. S. Haro, J. J. Blostein, A. Cicuttin, M. L. Crespo, F. A. Bessia, I. Sidelnik, M. G. Berisso, and J. Lipovetzky, "X-ray spectroscopy up to 17.6 keV using a commercial off the shelf CMOS image sensor," in *2020 Argentine Conference on Electronics (CAE)*. IEEE, 2020, pp. 69–72.
- [3] F. Alcalde Bessia, M. Pérez, J. Lipovetzky, N. A. Piunno, H. Mateos, I. Sidelnik, J. J. Blostein, M. Sofó Haro, and M. Gómez Berisso, "X-ray micrographic imaging system based on COTS CMOS sensors," *International Journal of Circuit Theory and Applications*, vol. 46, no. 10, pp. 1848–1857, 2018.
- [4] P. Rozenberg, T. Chauvin, M. Escobar-Sandoval, F. Huard, V. Shishov, J.-P. Charpentier, A.-S. Sergent, J. J. Vargas-Hernandez, A. Martínez-Meier, and L. Pâques, "Climate warming differently affects larch decidua ring formation at each end of a french alps elevational gradient," *Annals of Forest Science*, vol. 77, no. 2, pp. 1–20, 2020.
- [5] M. Pérez, M. S. Haro, J. Lipovetzky, A. Cicuttin, M. L. Crespo, F. A. Bessia, M. G. Berisso, and J. J. Blostein, "Evaluation of a commercial off the shelf CMOS image sensor for x-ray spectroscopy up to 24.9 keV," *Radiation Physics and Chemistry*, p. 109062, 2020.
- [6] M. Pérez, E. Martínez, J. Lipovetzky, J. Marín, M. S. Haro, F. A. Bessia, M. G. Berisso, and J. Blostein, "High spatial resolution neutron detection technique based on commercial off-the-shelf CMOS image sensors covered with nagdf₄ nanoparticles," *Journal of Instrumentation*, vol. 16, no. 08, p. P08008, 2021.
- [7] F. A. Bessia, M. Pérez, M. S. Haro, I. Sidelnik, J. J. Blostein, S. Suárez, P. Pérez, M. G. Berisso, and J. Lipovetzky, "Displacement damage in CMOS image sensors after thermal neutron irradiation," *IEEE Transactions on Nuclear Science*, vol. 65, no. 11, pp. 2793–2801, 2018.
- [8] F. H. Attix, *Introduction to radiological physics and radiation dosimetry*. John Wiley & Sons, 2008.
- [9] J. R. Janesick, *DN to [lambda]*. SPIE Press, 2007, vol. 170.

Implementación de sistema embebido para adquisición de señales XBT

1^{er} Patricio Bos

Dpto de Propagación Acústica
Dirección de Investigación de la Armada y Dirección de Investigación de la Armada y
UNIDEF (CONICET/MINDEF)
Vicente López, Argentina
pbos@fi.uba.ar

2^{do} Mariano Cinquini

Dpto de Propagación Acústica
UNIDEF (CONICET/MINDEF)
Vicente López, Argentina
mariano.cinquini@gmail.com

3^{er} Christian Galasso

Servicio de Análisis Operativo,
Armas y Guerra Electrónica
Escuela de Oficiales de la Armada
Base Naval Pto.Belgrano, Argentina
christian_galasso81@yahoo.com.ar

4^{to} Walter Correa

Facultad Regional Bahía Blanca
Universidad Tecnológica Nacional
Bahía Blanca, Argentina
waltercorrea86@gmail.com

Abstract—Se presenta la implementación de un sistema embebido para realizar la adquisición y comunicación automáticas de señales analógicas provenientes de sondas batitermográficas descartables que se utilizan en el campo de la oceanografía física en general, con aplicaciones de carácter dual, civil y militar. Se describen la interfaz de hardware que se utiliza para acondicionar las señales de la sonda y características relevantes del firmware de control para el kit de desarrollo NUCLEO-STM32F429ZI. Se utilizó el modelo de Máquina de Estados Finitos para implementar dos tareas manejadoras de eventos en el contexto de un RTOS cooperativo sencillo desarrollado *ad-hoc* para la aplicación.

I. INTRODUCCIÓN

Un batitermógrafo descartable, XBT por sus siglas en inglés (*eXpendable BathyThermograph*), es un instrumento utilizado para medir el perfil de temperatura de la columna de agua en navegación sin afectar las condiciones de operación del buque. El sistema fue desarrollado en la década del setenta por la firma Sippican Inc., hoy subsidiaria de Lockheed Martin y se sigue utilizando en la actualidad tanto en el área civil/científica como militar [1]. El sistema consta de tres componentes:

- sondas descartables;
- un lanzador que puede ser fijo o portátil;
- un adquisidor y registrador.

La sonda se deja caer por la banda del buque. A medida que desciende a través de la columna de agua, mide la temperatura. Pequeños cables transmiten los datos de temperatura al buque, donde se registran para su análisis. Debido a que la sonda cae a una velocidad conocida, la profundidad de la sonda puede deducirse desde el momento del lanzamiento con una ecuación provista por el fabricante.

Los XBTs están compuestos por una sonda con forma balística, un cánister, una bobina de cable dentro del cánister, un pin de lanzamiento y una tapa de cierre protectora para su almacenamiento. El pin de lanzamiento retiene la sonda dentro del cánister y cuando se retira, la sonda cae al agua

por efecto de la gravedad. La sonda contiene un termistor, que es el elemento sensible a la temperatura y que se encuentra conectado a una bobina de cable dentro de la sonda. El otro extremo del cable está enrollado a una segunda bobina dentro del cánister. La técnica de doble bobina permite que el cable quede libre en el agua en el punto de entrada y no se vea afectado por el movimiento del buque ni el descenso de la sonda.

En la figura 1 se pueden observar, a la izquierda, un esquema con los componentes del batitermógrafo junto con una fotografía de un XBT real dentro de su cánister, a la derecha.

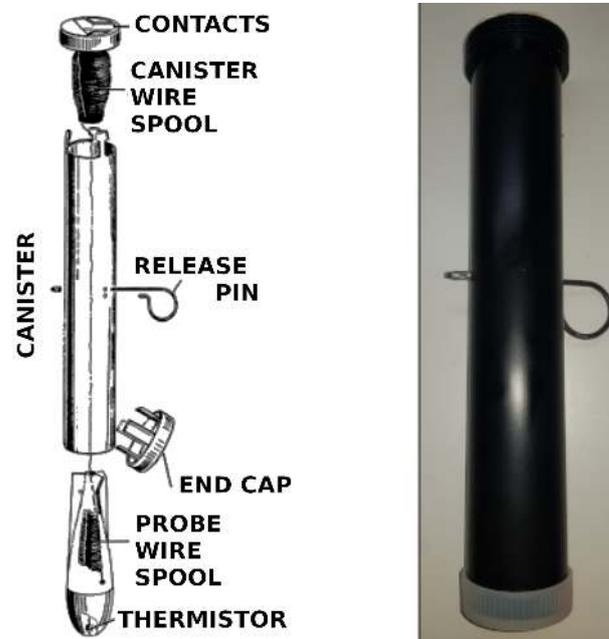


Fig. 1. Vista esquemática de componentes de una sonda XBT, tomada de [2] (panel izquierdo). Fotografía de un XBT real en su cánister (panel derecho).

B. Firmware

La arquitectura del firmware de control se estructuró modularmente en capas y se hizo uso de la *Hardware Abstraction Layer* (HAL) del paquete de STM32CubeF4, provisto por el fabricante del microcontrolador, STMicroelectronics [4]. Se implementó un RTOS sencillo con política de *scheduling* cooperativa. Dada la naturaleza reactiva del sistema, en la que el conjunto permanece en un mismo estado a la espera de un cambio, se implementó un despachador de eventos que lee un cola donde se registran los eventos del sistema. Si no hay eventos para despachar, el microcontrolador puede ingresar en un modo de bajo consumo. Los eventos que pueden reestablecer la actividad del sistema son el ingreso de una sonda al agua (MEASUREev) y/o la recepción de un comando (CMDev) desde un dispositivo externo conectado al microcontrolador.

Se implementaron dos Máquinas de Estados Finitos (MEF), COMM_Handler y XBT_Handler, para manejar la comunicación de la información hacia otros dispositivos y para realizar la adquisición de la información y controlar los modos de operación de la sonda XBT (*reload*, *launch*, *meassure*), respectivamente. Estas máquinas escriben los eventos del sistema en la cola de eventos del RTOS. Los diagramas de estados de las MEFs se pueden observar en la figura 4. Las máquinas se sincronizan a través de un evento MEASUREev que se dispara cuando la sonda XBT ingresa al agua y la MEF XBT_Handler pasa al estado MEASURE. Este evento provoca que la MEF COMM_Handler transicione hacia TX DATA y comience una transmisión de datos.

La máquina COMM_Handler inicia en el estado INIT, donde realiza todas las inicializaciones de hardware y software necesarias para establecer un canal de comunicación hacia otros dispositivos. En esta implementación se utilizó comunicación serie asincrónica sobre un puerto UART. La arquitectura modular del firmware permite emplear otros protocolos de comunicación sin necesidad de alterar la lógica de funcionamiento de la MEF. La máquina permanece en el estado IDLE a la espera de un evento, que puede ser la recepción de un comando por el puerto de comunicación o el lanzamiento de una sonda XBT, en cuyos casos se producirá una transición hacia los estados que darán tratamiento a estos eventos. Cabe destacar tres comandos que puede recibir la MEF: el primero es un *keepalive* para consultar y verificar que el dispositivo se encuentra operativo, el segundo es un pedido de transmisión del último perfil registrado y, finalmente, el tercero es un pedido de información sobre el modo de operación del sistema.

La máquina XBT_Handler inicia en el estado INIT en el que realiza todas las inicializaciones de hardware y software necesarias para la adquisición de datos y control de la sonda XBT. En particular, se inicializan los convertidores AD y DA del kit NUCLEO-STM32F429ZI que se utilizan para interactuar con la etapa de acondicionamiento de las señales XBT. Se emplean cuatro canales del conversor analógico-digital para realizar lecturas de tensión que permitan inferir los valores de corriente, resistencia y el modo de operación

del sistema. Alternativamente, se emplean dos canales del conversor digital-analógico para realimentar tensiones sobre las bases de los transistores Q1 y Q2 del circuito acondicionador de tal manera que las corrientes por sus colectores resulten idénticas. De esta condición depende la posibilidad de descontar los aportes de la resistencia de los devanados, R_{lead} , al cálculo de la resistencia del termistor de la sonda. Luego de la inicialización, la MEF cicla entre los estados RELOAD, LAUNCH y MEASURE en función de si hay o no una sonda en el lanzador y de si esta ingresó o no al agua.

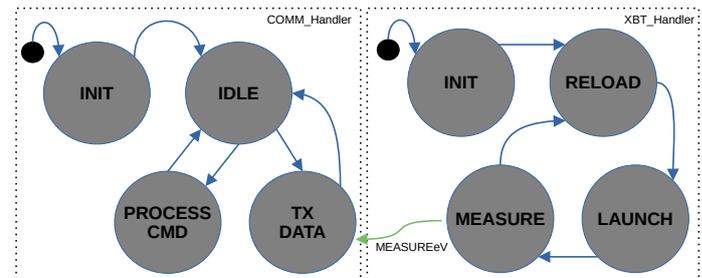


Fig. 4. Diagramas de estado de las Máquinas de Estados Finitos implementadas para la comunicación de datos hacia otros dispositivos (panel izquierdo) y para la adquisición de datos y control de modos de operación de la sonda XBT (panel derecho).

III. CONCLUSIÓN Y DISCUSIÓN

Se ha diseñado e implementado un sistema embebido que permite incorporar a un sistema existente en el ámbito operativo naval, una etapa de adquisición, digitalización y acondicionamiento de señales de sondas XBT, el cual posibilita el ingreso automático de información obtenida *in-situ* como entrada a un software propietario para el cálculo de parámetros acústicos que son de interés para diferentes Comandos de la Armada.

La automatización implícita en el sistema diseñado e implementado conlleva prestaciones con facilidades significativas para el usuario operativo naval. El sistema presentado permite reducir el tiempo desde que se realiza el lanzamiento de una sonda XBT hasta que la información procesada está disponible. Asimismo, se mejora la confiabilidad de la información al eliminar etapas intermedias del proceso de transmisión de la información propensas a introducir ruido en los datos.

REFERENCES

- [1] Lockheed Martin Mission Systems and Training, "Expendable Bathythermograph Expendable Sound Velocimeter (XBT/XSV) Expendable Profiling Systems" Disponible el 23/6/2022 en <https://www.lockheedmartin.com/en-us/products/oceanographic-instrumentation.html>
- [2] The Sippican Corporation, "INSTRUCTION MANUAL for the EXPENDABLE BATHY THERMOGRAPH SYSTEM", R-603G, 1972.
- [3] Patricio Bos y Mariano Cinquini, "Diseño e implementación de etapa de acondicionamiento de señal para sistema XBT", División Acústica Submarina, Informe Técnico AS 7/22, julio 2022
- [4] STMicroelectronics, "Description of STM32F4 HAL and low-layer drivers", User manual, UM1725, Rev 7 - junio 2021.

Wearable device prototype for vital signs monitoring

Leonardo Martínez Hornak, Isabel Morales, Alfredo Solari, and Julián Oreggioni,
 Universidad de la República, Montevideo, Uruguay
 Email: {leonardo.martinez.hornak, imorales, asolari, juliano}@fing.edu.uy

Abstract—Monitoring vital signs is a basic need in healthcare institutions. Due to the limited capacity of these facilities, wearable devices have emerged aiming to aid ambulatory monitoring and doctors' daily activities. This work presents the proof of concept of a wearable device for non-invasive real-time vital signs monitoring. The prototype comprises an MSP432P401R microcontroller, an HC-06 Bluetooth module, and an SEN-15219 biometric sensor board. This device acquires and wirelessly transmits photoplethysmography signals, heart rate, and blood oxygen saturation level, featuring an autonomy of almost 40 hours.

Index Terms—long-term monitoring, photoplethysmography, PPG, heart rate, blood oxygen saturation, SpO2

I. INTRODUCTION

The top global causes of death are associated with three broad topics: cardiovascular (ischaemic heart disease, stroke), respiratory (chronic obstructive pulmonary disease, lower respiratory infections), and neonatal conditions [1]. The treatments for these diseases rely on daily monitoring of vital signs. Moreover, the need for ambulatory low-cost medical equipment to give quick, continuous, and remote access to physiological parameters to improve communication, diagnosis and treatment, is increasing [2], [3].

Monitoring heart rate (HR), blood pressure, temperature, and blood oxygen saturation level (SpO2) are fundamental in the physiological assessment of a patient. The photoplethysmography (PPG) signal is employed to obtain these parameters non-invasively, as stated by Webster [4] for different diagnostic applications. This technique uses a light source capable of transmitting different wavelengths and a photodetector. The light transmitted is absorbed, and then it is spread over human tissue. The reflected light variation corresponds to blood volume changes (known as plethysmography) [4].

The literature reveals several works addressing the development of vital signs monitoring wearable devices using PPG. Such as low-cost devices aiming at non-medical markets like daily activity or sports routine monitoring [2]. Although there are indications that these devices could have a medical grade in the future [5], commercially available solutions are usually not intended for medical use [6], [7]. It is also possible to see mobile healthcare devices from medical device companies [8], [9]. Finally, it is feasible to find open-source alternatives [10] and research efforts [11], [12].

This work introduces the proof of concept of a wearable device that acquires PPG, HR, SpO2 and temperature. The device wirelessly transmits these parameters in real time to a personal computer (PC), where a graphical user interface aimed for research and development runs (see Fig. 1). In the

future, the system will also show the vital signs on a display located in the wearable device.

II. PROPOSED SOLUTION

The system consists of 4 stages (see Fig. 1): data acquisition (HR, SpO2, temperature, and PPG), data processing, data transmission via Bluetooth (BT) link, and data visualization.

A. Hardware

Fig. 1 shows the hardware elements of the solution and their interactions.

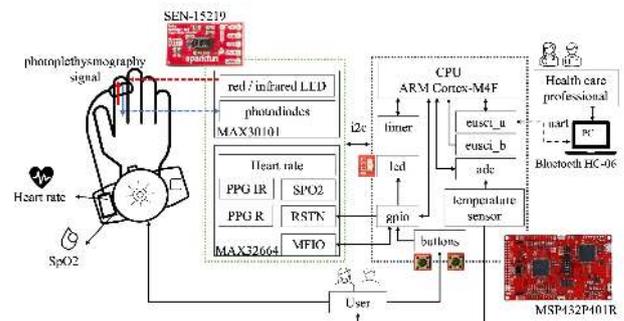


Fig. 1. Block diagram of the vital signs monitoring system.

The SEN-15219 board from SparkFun performs the acquisition of HR, SpO2, and PPG raw data. This board consists of an analog front-end (MAX30101) and a sensor hub (MAX32664), both chips from Maxim Integrated. The MAX30101 has three sensor LEDs: a red one (R), an infrared one (IR), and a green one. When users place their index finger on top of the LEDs, photodiodes capture the reflected light emitted by the LEDs and, therefore, the PPG data. The MAX32664 internally calculates the HR and SpO2 values using the PPG with an accuracy that would comply with the FDA standards [13] and transmits them using the I2C serial protocol.

MSP432P401R from Texas Instruments is a 32-bit ARM-Cortex M4F microcontroller with a hardware floating-point unit. It was configured to work in two modes: i) low power mode 0 (LPM0) to save energy during the wait times in communication with the SEN-15219 board; and, ii) active mode at 48 MHz. It communicates with the SEN-15219 board using I2C at 400 kHz and the HC-06 BT module using UART (Universal Asynchronous Receiver-Transmitter) at 115200 bps. Temperature is obtained using the microcontroller's internal sensor.

B. Embedded software

The embedded software follows a Function-Queue-Scheduling architecture [14]. Interrupt service routines (ISR) add function pointers (tasks) to a queue while the main loop reads these pointers from this queue. Thus, the corresponding queued function pointers is called. This architecture allows the definition of low and high-priority tasks separately (like alarms management and data acquisition).

Fig. 2 illustrates the workflow followed by the embedded software. After system initialization, which includes the configuration of the SEN-15219 board and microcontroller peripherals, the device enters LPM0. Then, the embedded system performs the same tasks periodically. Every 40 ms, it obtains PPG (samples of IR and R), HR, and SpO2 data from the SEN-15219 board. When 200 ms elapsed, it sends PPG signal values to the BT module, that are forwarded to the PC for real-time visualization. Finally, every 1 second, it acquires and sends HR, SpO2, and temperature to the PC.

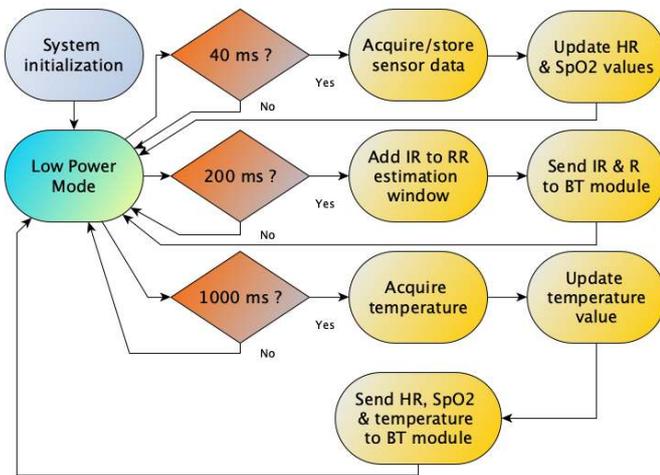


Fig. 2. Embedded software workflow

The configuration of the SEN-15219 board consists of enabling its output mode to acquire the PPG signal (MAX30101) and enabling the internal algorithm for calculating HR and SpO2 (MAX32664). The embedded software initializes the MAX32664 with a data rate of 100 samples per second (in 24-bytes data frames) and parses these frames obtaining the raw data, the parameters, and a status byte.

The system uses ISRs to handle the peripheral data. The MSP432P401R has a nested vector interrupt controller that permits setting priorities for each ISR. EUSCI-B (Enhanced Universal Serial Communication Interface) was configured for I2C communication protocol with the SEN-15219 board with the highest priority. This configuration ensures that the microcontroller obtains the data from the SEN-15219 board timely. The other peripherals that use ISRs are the General-Purpose-Input-Outputs (GPIO) for buttons, the TIMER-A for timing tasks, ADC for temperature acquisition, and the EUSCI-A used in UART mode for interfacing with the BT module. The system configures all these ISRs with low priority.

III. EXPERIMENTAL RESULTS

Fig. 3 shows the test setup used for the experiments.

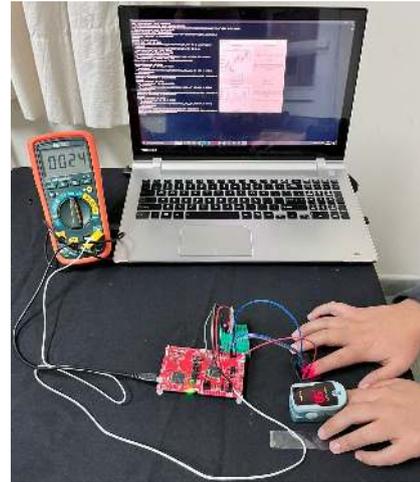


Fig. 3. Wearable device prototype under testing

A. Energy Consumption and autonomy

The energy consumption was measured considering two operation modes: i) *continuous mode*, where the device acquires data and transmits them to a PC; ii) *intermittent mode*, where every 256 seconds, a timer wakes the system up from low power mode 3 (LPM3) to acquire data and transmit them for 30 seconds.

We measure the RMS current drained from the power supply using a standard 3.5-digit multi-meter and *Energy Trace* from Texas Instruments. Table I shows energy consumption and autonomy. The autonomy was determined considering a 235 mAh coin cell lithium CR2032 battery.

TABLE I
ENERGY CONSUMPTION AND SYSTEM AUTONOMY (IF POWERED BY A 235 MAH BATTERY)

Operation Mode	Energy (mWh)	Battery Life (HH:MM)
Continuous	136.3	05:11
Intermittent	17.8	39:37

The most power-hungry component of the system is the SEN-15219 board. In the continuous mode with a finger over the LEDs sensor, the SEN-15219 board consumes 86 mWh (63.1 %). It has a baseline energy consumption of 26.2 mWh (19.3 %) when the chips are off. The LEDs sensor consumes 40 mWh (29.4 %), and the rest of the board consumes 19.6 mWh (14.4 %). On the other hand, the HC-06 module consumes 43 mWh (31.6 %) and the microcontroller 7.3 mWh (5.3 %).

The system allows almost 40 hours of intermittent monitoring which can help to store enough data if needed by the physician.

B. Memory usage

Memory usage was measured using *Code Composer Studio* from Texas Instruments. Table II shows these results.

TABLE II
MEMORY USAGE

	Flash (kB)	Flash (%)	RAM (kB)	RAM (%)
Available	256	100	64	100
Used	6.9	2.7	36.1	56.2

C. Vital signs measurement

Experimental evaluation was performed on a healthy young adult (female, 26 years old) in a quiet room in a home environment. The test subject was instructed to synchronize her respiration rate with a constant rhythm provided by a metronome.

Table III shows results from the experimental evaluation using the prototype (Device) and its comparison with values obtained using the CE-certified pulse oximeter CMS50D from Contec Medical Systems. The Mean Absolute Error (MAE) and Relative Error of the measurements are low and acceptable for the targeted application.

TABLE III
DEVICE MEASUREMENTS

Test	Heat Rate (bpm)			SpO2 (%)		
	Ref	Device	Rel. error	Ref	Device	Rel. error
1	75.0	74.5	0.67%	94.0	94.7	0.74%
2	71.0	70.5	0.71%	94.0	95.4	1.47%
3	68.0	68.4	0.58%	94.0	96.6	2.69%
4	67.0	69.4	3.46%	94.0	93.8	0.21%
5	68.0	68.6	0.87%	95.0	95.6	0.63%
6	73.0	75.1	2.80%	93.0	93.6	0.64%
7	69.0	70.9	2.68%	96.0	96.8	0.83%
8	71.0	71.8	1.11%	96.0	98.0	2.04%
9	68.0	69.4	2.02%	96.0	96.1	0.10%
10	72.0	71.2	1.12%	98.0	97.1	0.93%
MAE	1.14			0.99		

D. Comparison with other solutions

Table IV summarizes the main characteristics of our system compared to other devices (prototypes and commercial products). A fair comparison between such different implementations is difficult. Nevertheless, this qualitative comparison shows the potential of our proposal and locates our prototype within the state-of-the-art. All considered solutions present the sensor inside a fingertip case. The comparison shows that our prototype performs well in line with other systems while providing low cost and a potential FDA approval.

IV. CONCLUSIONS

The prototype of a low-cost device has performed real-time acquisition of relevant vital signs in a non-invasive way. It has also achieved the wireless transmission of these signs to a PC, featuring an autonomy of almost 40 hours. The prototype performs well if compared with similar commercial products. It provides relatively high battery life and could be potentially indicated for clinical usage.

Our following steps include incorporating respiratory rate estimation into our device using already acquired PPG data and conducting more experiments. In addition, we are planning to add a display, decrease power consumption by using

TABLE IV
COMPARISON WITH OTHER WEARABLE DEVICES WITH FINGERTIP CASE

	Nonin [8]	Masimo [9]	Uguz [11]	This work
Parameters	HR, SpO2	HR, SpO2	HR, PPG, Temp	HR, SpO2 PPG, Temp
Connectivity	BLE	BLE	Wired	BT
Autonomy	44 hs	15 hs	N/A	39.6 hs
Power source	Alkaline AAA (x1)	Alkaline AAA (x1)	N/A	CR2032 (x1)
Size	56x74 mm	41x71 mm	N/A	N/A
Weight	71 g	73 g	N/A	N/A
Clin. approval	Yes	No	No	Potential
Cost	\$969	\$299	N/A	N/A
Status	Commercial	Commercial	Prototype	Prototype

BLE (Bluetooth Low Energy) instead of BT, incorporate the MAX30101 and MAX32664 into a custom PCB, and reduce the system's overall size to develop a suitable case.

ACKNOWLEDGMENT

This work was partially funded by CSIC (Comisión Sectorial de Investigación Científica, Uruguay) and IEEE CAS Uruguay Chapter. The experimental procedures involving human subjects described in this paper were approved by the Ethics Committee, School of Psychology, Universidad de la República, Uruguay. The authors would like to thank to Leonel Gómez-Sena and Antonella Arrieta for guidance on requirements and application of the prototype.

REFERENCES

- [1] WHO. (2020, 12) The top 10 causes of death. Accessed September 2021. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>
- [2] M. A. F. Pimentel, A. E. W. Johnson, P. H. Charlton, D. Birrenkott, P. J. Watkinson, L. Tarassenko, and D. A. Clifton, "Toward a robust estimation of respiratory rate from pulse oximeters," *Eng. IEEE Trans. Biomed. Eng.*, vol. 64, no. 8, pp. 1914–1923, 2017.
- [3] J. Oreggioni, A. A. Caputi, and F. Silveira, "Biopotential monitoring," in *Encyclopedia of Biomed. Eng.*, R. Narayan, Ed. Oxford: Elsevier, 2019, pp. 296–304. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128012383641612>
- [4] J. G. Webster, *Design of pulse oximeters*. CRC Press, 1997.
- [5] S. H. Browne, M. Bernstein, and P. E. Bickler, "Accuracy of samsung smartphone integrated pulse oximetry meets full FDA clearance standards for clinical use," *medRxiv*, 2021.
- [6] Apple. (2022, 06) Apple Watch Series 7. [Online]. Available: <https://www.apple.com/apple-watch-series-7/>
- [7] Samsung. (2022, 06) Galaxy Watch3. [Online]. Available: <https://www.samsung.com/latin/watches/galaxy-watch/galaxy-watch3-45mm-mystic-silver-sm-r840nzsalta/>
- [8] Nonin. (2022, 01) WristOx2 Model 3150 with BLE. [Online]. Available: <https://www.nonin.com/support/3150-ble>
- [9] Masimo. (2022, 01) MightySat. [Online]. Available: <https://www.masimopersonalhealth.com/pages/mightysat>
- [10] HealthyPi. (2021, 09) Welcome to HealthyPi v4. [Online]. Available: <https://healthypi.protocentral.com/>
- [11] D. Uguz, "Design of a multipurpose photoplethysmography sensor to assist cardiovascular and respiratory diagnosis," in *21th Int. Student Conf. on Elect. Eng.*, vol. 21, 2017, pp. 1–7.
- [12] F. Braun, P. Theurillat, M. Proença, A. Lemkaddem, D. Ferrario, K. De Jaegere, C. Horvath, C. Roth, A.-K. Brill, M. Lemay, and S. Ott, "Pulse oximetry at the wrist during sleep: Performance, challenges and perspectives," in *42nd Annu. Int. Conf. of the IEEE Eng. in Med. Biol. Soc.*, 2020.
- [13] Maxim Inc., "Wearable optical measurement solution based on MAX32664," App. Note 7400, Tech. Rep., Mar 2021.
- [14] D. E. Simon, "Survey of software architectures," in *An Embedded Software Primer*. Addison-Wesley Professional, 1999, pp. 115–136.

Detector de cajas calientes en material rodante

Fabián Leonardo Sarmiento[†], Gustavo Ramoscelli^{*}, Ariel Lutenberg[†]

[†]Laboratorio de Sistemas Embebidos, CONICET-GICSAFe, Facultad de Ingeniería,
Universidad de Buenos Aires, Paseo Colón 850, Buenos Aires, Argentina

^{*}Departamento de Ingeniería, CONICET-GICSAFe, Avenida Alem 1253, Bahía Blanca, Argentina
lse@fi.uba.ar

Resumen—Las vías ferroviarias pueden sufrir daños severos si uno de los ejes del material rodante se bloquea y las ruedas en lugar de rodar se arrastran por las vías. Esta situación puede detectarse en forma temprana mediante el monitoreo de la temperatura de las cajas de punta de ejes de las formaciones. En este trabajo se presenta un prototipo validado en laboratorio de un sistema destinado a esta finalidad, que se realizó a pedido de Trenes Argentinos, con el objetivo de sustituir importaciones. Se presenta el diseño de un sistema embebido y de una aplicación multiplataforma que permiten la visualización de las temperaturas de las cajas registradas.

Palabras clave—Detector de cajas calientes, seguridad ferroviaria, material rodante

I. INTRODUCCIÓN

Las cajas de punta de eje, o *cajas de grasa*, ver figura 1, contienen los elementos para que las ruedas de las formaciones ferroviarias puedan rotar [1]. El mal funcionamiento de una caja de punta de eje puede bloquear un conjunto de ruedas, de modo que en lugar de rotar las ruedas se arrastrarán por las vías, lo que puede ocasionar incendios y descarrilamientos. En general estas fallas se producen en forma gradual, lo que permite su detección mediante métodos de inspección no destructivos antes de que el daño causado sea muy grave.



Fig. 1. Fotografía de un bogie donde se indican las cajas de puntas de eje.

La detección se puede realizar con la formación en movimiento mediante un “detector de cajas calientes”. El estándar Europeo UNE-EN 15437 [2], define dos métodos para sistemas de monitoreo de cajas de punta de eje:

- Monitoreo realizado por sistema abordo de la formación.
- Supervisión efectuada por detectores ubicados en la vía.

El prototipo que se presenta en este trabajo corresponde a detectores ubicados en vía.

Un parámetro importante a considerar es el tiempo de respuesta del sensor infrarrojo a utilizar, ya que este debe

tomar varias mediciones de la temperatura de la caja de punta de eje aún cuando la formación se desplaza a la máxima velocidad permitida. Si se considera una velocidad de 80 km/h, que es la velocidad máxima permitida en vía en la mayor parte de la Argentina, una caja de punta de eje con 230 mm de diámetro y el requisito establecido en este proyecto de tener al menos tres muestras de la temperatura de la caja, se obtiene que la velocidad de muestreo del sensor infrarrojo debe ser de:

$$3 \text{ ms} = \frac{230 \text{ mm}}{3 \times 80 \text{ km/h}} \quad (1)$$

En la actualidad existen diferentes compañías que se dedican a la fabricación y comercialización de detectores de cajas calientes (*Hot Box Detector*, HBD) para material rodante. Los HBD disponibles en el mercado presentan características similares, por ejemplo el uso de sensores infrarrojos como elemento de medición, pero difieren en cuanto a la tecnología empleada respecto a la transmisión y presentación de la información. A continuación se nombran algunas compañías y las características generales de sus respectivos productos:

- *EKE Electronics* desarrolló un sistema de monitoreo abordo [3], que se encuentra integrado al *Train Control Monitoring System* (TCMS) [4] de la formación.
- *ITSS Tech* desarrolló un equipo de medición infrarrojo (modelo Pegasus [5]) que se instala en vía y cuenta con herramientas de análisis gráfico y diagnóstico.
- *Voestallpine Signalling* ofrece el sistema modular Phoenix HBD [6] que se instala en vía y se comunica mediante redes LAN y GSM.

Los principales trabajos de investigación en la temática son:

- Gampze Garip et al. presentan en [7] el diseño de un HBD basado en sensores infrarrojos ubicados en vía, detectores de la formación y una terminal de comando.
- Shöebel et al. [8] desarrollaron un sistema basado en sensores infrarrojos, una unidad de control, un equipamiento de transmisión de datos y un display.
- Wu et al. [9] desarrollaron un sistema que discrimina los puntos calientes empleando técnicas de clasificación Bayesianas y utiliza una computadora como terminal.

Los trabajos académicos no cuentan con interfaz gráfica ni almacenamiento de datos, y los productos comerciales solo cuentan con aplicaciones para escritorio.

A nivel nacional y latinoamericano no se encontraron productos comerciales ni trabajos académicos relacionados con la temática abordada en este trabajo.

En lo que sigue la organización es la siguiente: en la sección II se presenta el diseño realizado, en la sección III los resultados obtenidos y en la sección IV las conclusiones.

II. DISEÑO E IMPLEMENTACIÓN

La arquitectura desarrollada en este trabajo se muestra en la figura 2. A continuación se describe cada componente:

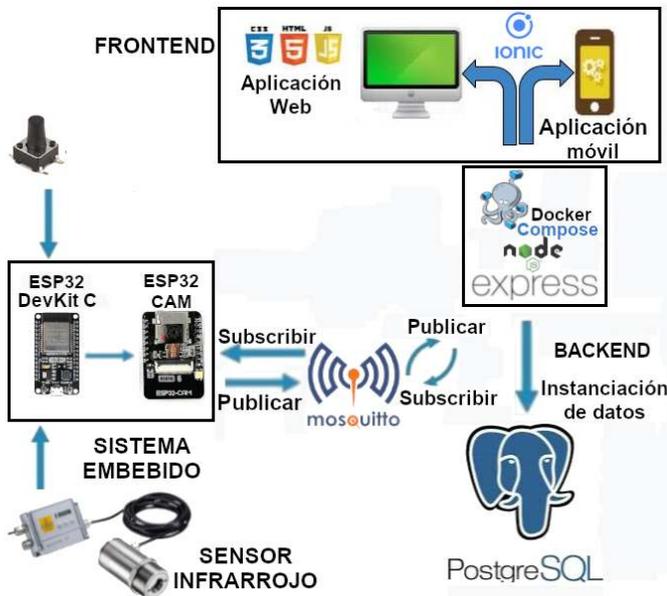


Fig. 2. Arquitectura del sistema.

- Pulsador *on/off*: simula la detección de material rodante en la vía, lo que activa al sistema. En la aplicación en campo esta señal es generada por el sistema ferroviario.
- Sistema embebido: su función es leer la temperatura del sensor cada 1 ms, publicar su valor mediante MQTT cada vez que se excede el umbral establecido, enviar las mediciones a una base de datos a través de API REST y tomar una fotografía para poder constatar que el objeto medido es una formación ferroviaria. Se utilizó una placa ESP32 DevKitC [10] junto con una placa ESP32 Cam [11], como se ilustra en la figura 3, porque la placa ESP32 Cam no soporta de manera simultánea el funcionamiento del conversor analógico digital, la cámara y la red Wi-Fi. Criterios adicionales que explican esta decisión de diseño se explican más adelante en esta misma sección.
- Sensor infrarrojo: realiza la lectura de temperatura de la caja de punta de ejes.
- Broker MQTT: permite la comunicación entre el sistema embebido y la API REST. Se utilizó Mosquitto.
- API REST: se conecta con la base de datos y con el Frontend de la aplicación.
- Frontend Ionic: es la interfaz gráfica de la aplicación mediante la cual se muestran los resultados obtenidos.
- Base de datos Postgres: permite el almacenamiento de forma virtual de las temperaturas medidas.

Otros criterios adicionales considerados para la elección de las placas que componen el sistema embebido fueron:

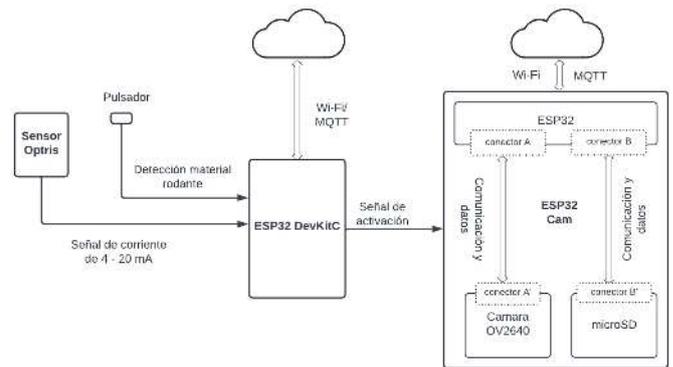


Fig. 3. Diagrama detallado del sistema embebido.

1) *Conectividad Wi-Fi*: el sistema embebido debe contar con la posibilidad de conectarse a Internet a través de una red Wi-Fi para transmitir los datos y las imágenes capturadas. Las placas de desarrollo ESP32 soportan IEEE 802.11 b/g/n.

2) *Consumo de energía*: se consideró este factor ya que el sistema podría tener que funcionar en base a una batería. Las ESP32 cuentan con diversos modos de bajo consumo, que fueron utilizados en el desarrollo.

3) *Captura de imágenes fotográficas*: la placa ESP32 cam cuenta con una cámara y un *slot* para microSD card. Se evaluó utilizar una placa Arduino Uno junto con los módulos de la cámara OV7670, microSD card y el *shield* de Wi-Fi, pero implica un mayor número de módulos.

4) *Costo*: Las placas ESP32 utilizadas cuestan 15 USD, mientras que otras opciones analizadas resultan ser más costosas, como por ejemplo el conjunto Arduino Uno + cámara OV7670 + *shield* de Wi-Fi que cuesta 20 USD.

A partir de las consideraciones presentadas en la sección I se utilizó un sensor infrarrojo Optris CTlaser 3ML [12]. Este sensor fue provisto por Trenes Argentinos y es parte de la serie de alto rendimiento de la firma Optris, que permite mediciones de temperatura en objetos a una distancia óptima de 1100 mm. En la tabla I se pueden observar las principales características del sensor utilizado. Al momento de la elección Trenes Argentinos consideró su tiempo de respuesta, su disponibilidad en el mercado local y su rango de medición, que permite distinguir la temperatura normal de la caja (60 °C), de temperaturas superiores a 100 °C que indican una falla.

Para la lectura de la medición de temperatura se utilizó su salida de 4-20 mA, que se digitalizó mediante el ESP32 DevKitC, como se ilustra en la figura 3.

TABLA I
PRINCIPALES CARACTERÍSTICAS DEL SENSOR OPRIS CTLASER 3ML.

Descripción	Optris CTlaser
Tiempo de respuesta	1 ms
Rango de medición	50 °C a 400 °C
Salida analógica	0/4-20 mA, 0-5/10 V

III. ENSAYOS Y RESULTADOS

Para comprobar el funcionamiento de cada uno de los módulos del sistema se utilizaron diversas herramientas de

hardware y *software*. En particular se empleó un resistor de potencia para tener una fuente de temperatura que permitiera simular un eje caliente y se utilizó un disco ranurado montado sobre el eje de un motor de pasos para interrumpir el haz del sensor infrarrojo durante intervalos de tiempo controlados y así poder simular que la caja caliente pasa por delante del sensor infrarrojo con una determinada velocidad. Por falta de espacio no es posible presentar más detalles sobre el ensayo realizado. Este detalle será pronto publicado en el trabajo final de maestría del primer autor de este documento.

Las herramientas de *software* empleadas en los ensayos se pueden observar en la tabla II junto con una breve descripción de la funcionalidad para la que se empleó cada una de ellas.

TABLA II
RECURSOS DE SOFTWARE EMPLEADOS.

Recursos	Función
Postman	Pruebas del backend, la API REST y el frontend
Google Chrome y Firefox	Pruebas del frontend
PgAdmin	Visualización de la base de datos

La figura 4 muestra los valores de temperatura registrados por el sensor en función del tiempo. Es de interés mostrar la variación de la temperatura en el tiempo, almacenar los datos para su posterior análisis y generar una alarma si se excede el umbral establecido en 100 °C.



Fig. 4. Pantalla sensores de la aplicación.

Se puede afirmar que los requisitos del proyecto fueron alcanzados:

- Las pruebas mostraron que el sistema se encuentra preparado para efectuar mediciones en tiempo real y en el rango de temperaturas establecido.
- Si la temperatura medida supera el umbral establecido los valores son almacenados en una base de datos, junto con una fotografía del objeto medido y se genera una alarma.
- Los valores medidos son presentados en una aplicación de computadora o teléfono móvil según indique el usuario.

- La aplicación desarrollada permite el alta de múltiples usuarios y a cada uno de ellos se le puede asignar acceso a los detectores de ejes caliente que corresponda.

IV. CONCLUSIONES

En este trabajo se abordó el desarrollo de un sistema que puede detectar en forma temprana fallas en las puntas de eje de las formaciones ferroviarias, lo que permite evitar que una rueda bloqueada cause daños en las instalaciones y eventualmente a las personas a partir de descarrilamientos o incendios. El prototipo implementado cumplió con los requisitos del proyecto y sus características son comparables con la de los equipos comerciales disponibles en el mercado. Sin embargo, se destaca por ser el único que cuenta con interfaz gráfica, almacenamiento de datos, y la posibilidad de ser desplegado en teléfonos móviles y computadoras. Por eso se considera que sirve de base para desarrollar versiones posteriores del sistema que permitan sustituir importaciones, que es el objetivo principal propuesto por Trenes Argentinos.

A futuro se espera realizar pruebas en campo y desarrollar una versión apta para su fabricación a escala industrial y su uso en campo. A su vez, se considera prescindir de la placa ESP32 DevKitC y utilizar solamente la plataforma ESP32 Cam. De cumplir estos objetivos el sistema sería transferido a Trenes Argentinos para ser instalado en distintas locaciones del sistema ferroviario argentino.

REFERENCIAS

- [1] A. Peña, "Simulación de la dinámica de un vehículo ferroviario mediante herramientas de ingeniería asistidas por ordenador", Esc. de Ing. Industriales, Univ. of Extremadura, Cáceres, and Badajoz, Spain, 2016.
- [2] "Aplicaciones ferroviarias. Monitorización de la condición de las cajas de grasa. Requisitos de interfaz y diseño. Parte 1: Equipo al lado de la vía y cajas de grasa del material rodante", UNE-EN 15437-1:2010, 2010.
- [3] "Hot Axle Box Detection System (HABD)". <https://www.eke-electronics.com/hot-axle-box-detection-system-habd> (accedido 2 julio 2022).
- [4] S. Tew. "What is TCMS". <https://www.railengineer.co.uk/what-is-tcms/> <https://www.railengineer.co.uk/2015/08/11/what-is-tcms/> (accedido 2 julio 2022)
- [5] "Pegasus HABD/HWD : Hot axle box/ Hot wheel detector." ITSS. <http://www.itss.tech/pegasus-habd-hwd> (accedido 2 julio 2022)
- [6] "Modular Diagnostic System - PHOENIX MDS." Voestalpine one step ahead. <https://www.voestalpine.com/signaling/en/products/Modular-Diagnostic-System-PHOENIX-MDS/> (accedido 2 julio 2022).
- [7] Gamze Garip and Ilker Ustoglu and Tarik Veli Mumcu and Ozgur Turay Kaymakci, "Hot Box Detection System Design for Railway Vehicle Safety", 2014.
- [8] A. Schöbel, M. Pisek and J. Karner, "Hot box detection systems as a part of automated train observation in Austria", 2006.
- [9] Y. Wang and Z. Wu. "A Train Hot Bearing Detection System Based on Infrared Array Sensor", in Proc. of the 2nd International Symp. on Computer, Communication, Control and Automation (ISCCA 2013), Atlantis Press, Ed. Feb. 2013, pp 733-736, doi: <https://doi.org/10.2991/iscca.2013.184>.
- [10] "ESP32-DevKitC V4 Getting Started Guide". ESP-IDF. <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-devkitc.html> (accedido 2 julio 2022).
- [11] "ESP32-CAM camera development board". Ai-Thinker Technology. <https://docs.ai-thinker.com/en/esp32-cam> (accedido 2 julio 2022).
- [12] "Pyrometer optris CT 3M for low-temperature measurements of metal." Optris.global. <https://www.optris.global/optris-ct-3m> (accedido 2 julio 2022).

Diseño e implementación de Data Logger open hardware para estación meteorológica

1^{ro} Lázaro Andrés O’Farrill Nuñez

*Estudiante de pregrado de ingeniería biomédica
Universidad Tecnológica de la Habana CUJAE*

La Habana, Cuba

<https://orcid.org/0000-0002-1146-2775>

2^{do} Alex Manuel Rivera Rivera

*Estudiante de pregrado de ingeniería automática
Universidad Tecnológica de la Habana CUJAE*

La Habana, Cuba

<https://orcid.org/0000-0002-2054-9645>

Resumen—El presente trabajo describe el diseño e implementación del hardware y software de un sistema de adquisición y procesamiento de datos de una estación meteorológica. El hardware del sistema está basado en el ESP32 de Espressif Systems y una serie de módulos comerciales de bajo costo para suplir las funcionalidades de RTC (real time clock), la conexión vía ethernet, el almacenamiento interno y la conexión vía RS485. Aprovechando la disponibilidad de WiFi embebida en el ESP32, el sistema se comunica con una computadora personal, donde un software hecho a medida permite realizar los ajustes al sistema y visualizar gráficamente los datos. Los resultados obtenidos permitieron poner en funcionamiento a la estación meteorológica WS501-UMB de la empresa Lufft que carecía del Data Logger, en la actualidad el sistema está siendo utilizado por el Instituto de Ingeniería Agrícola de Cuba

Palabras Clave—sistema de adquisición de datos, estación meteorológica, ESP32, Data Logger

I. INTRODUCCIÓN

El monitoreo de los parámetros ambientales es de vital importancia en una gran cantidad de aplicaciones y procesos industriales. Hace unas décadas, los sistemas de monitoreo del clima eran instrumentos mecánicos y electromecánicos con errores asociados a la necesidad de intervención humana, el paralaje y durabilidad del equipo. Con la llegada de la automatización a esta rama, los equipos se hicieron más compactos, seguros y económicos; sin embargo, la mayoría de este tipo equipos carecen de un sistema embebido para registro, persistencia y procesamiento de datos. Los Data Loggers (como se conocen estos sistemas de registro de datos) son dispositivos electrónicos que registran, escanean y recuperan automáticamente los datos con alta velocidad y eficiencia durante una prueba o medición [1]. El usuario, normalmente, es el que determina el tipo de información registrada; por lo tanto, puede medir automáticamente la salida eléctrica de cualquier tipo de transductor y registrar el valor.

II. MATERIALES Y MÉTODOS

La mayoría de los Data Loggers existentes en mercado al igual que el descrito en este trabajo se basan en microcontroladores. El microcontrolador es la parte fundamental del sistema, pero también la más costosa.

Para este trabajo se utilizó la placa de desarrollo ESP32. Este microcontrolador es un único chip combinado de Wi-Fi y Bluetooth de 2,4 GHz diseñado para lograr el mejor rendimiento de potencia y RF, mostrando robustez, versatilidad y confiabilidad en una amplia variedad de aplicaciones y escenarios de energía. ESP32 está diseñado para aplicaciones móviles, dispositivos electrónicos portátiles e Internet de las cosas (IoT); cuenta con todas las características de vanguardia

de los chips de bajo consumo, que incluyen desde múltiples modos de alimentación y potencia dinámica, hasta escenarios de aplicación específica, con concentración de sensores IoT de baja potencia activándose periódicamente cuando detecta condiciones específicas.

Para la persistencia de los datos se utiliza una memoria MicroSD de 4GB. La comunicación con la estación meteorológica se realiza vía RS485 utilizando el protocolo de comunicación serial; para la comunicación con el software de la PC el Data Logger hace uso de la tecnología WiFi del microcontrolador.

III. HARDWARE

En la Tabla I se describen los componentes utilizados.

Como se ha descrito anteriormente el Data Logger diseñado gira en torno al microcontrolador ESP32, este se dedica a encuestar los distintos sensores cada un período de tiempo configurable en desde el ordenador. La forma de encuestar los datos es mediante el protocolo serial, el cual expone la estación meteorológica a través de una interfaz RS485. Una vez adquirida la lectura, se realiza un procesamiento mínimo y pasa a ser almacenada en la MicroSD. Debido al tamaño tan pequeño de los datos el dispositivo puede estar almacenando durante toda la vida útil del equipo; estimada en al menos 10 años de uso ininterrumpido. El RTC (TinyRTC), cuenta con el integrado DS1307, capaz de almacenar y llevar la cuenta de la fecha y hora de manera autónoma, parte fundamental del sistema, pues brinda los elementos para almacenar los datos de forma ordenada y analizable. Entre las vías de conexión entre el PC y el dispositivo tenemos: la WiFi, embebida en el microcontrolador y gestionada por este, pero además el sistema cuenta con la posibilidad de conexión por ethernet con el ya mencionado módulo ENC28J60 de Microchip con acceso directo a memoria (DMA), hardware específico para el cálculo de las sumas de control (IP checksums), filtrado de paquetes y más.

Nombre	Función
DOIT ESP32 DEVKIT V1	Microprocesador
Módulo SD de Arduino	Almacenamiento de datos
Tiny RTC	Reloj de tiempo real
Convertor MAX485	Comunicación con la estación meteorológica
Módulo ENC28J60	Comunicación con el PC vía ethernet

Tabla I
PRINCIPALES COMPONENTES DEL DISPOSITIVO.

En la Fig. 1 se muestra un diagrama de la configuración descrita.

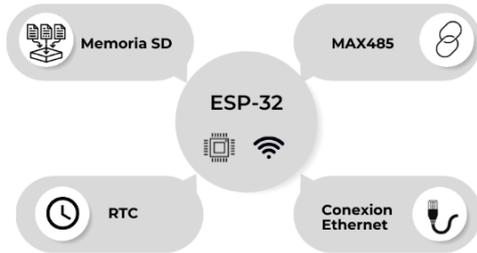


Figura 1. Relación entre los módulos descritos.

Finalmente, el circuito es alimentado con una fuente de 5 V DC.

En la Fig. 2 podemos ver una vista 3D generada en Proteus de todo el circuito.



Figura 2. Render del dispositivo en Proteus.

IV. SOFTWARE

Hay tres procesos fundamentales en el uso del Data Logger : la recolección de los datos, la distribución y su posterior visualización.

A. Recolección de datos

La recolección de datos de la estación ocurre capturando la información de la estación a intervalos pre configurados utilizando el protocolo de comunicación serial expuesto por la misma. Estos datos son almacenados en la memoria del dispositivo de manera secuencial. Se registra el valor de cada variable medida junto con el momento exacto de la medición.

En la Tabla II aparecen las variables registradas por el dispositivo.

B. Distribución de datos

El Data Logger integra capacidades de comunicación inalámbrica y cableada. Está provisto de un servidor web, mediante el cual es posible alterar su configuración y obtener los datos registrados. La comunicación inicial ocurre utilizando WebSockets. Esta es una conexión de baja latencia que permite mantener la comunicación abierta de manera indefinida y la comunicación bidireccional entre dispositivos.

Este protocolo provee una forma estándar para el servidor de enviar contenido al cliente sin que este lo solicite [2].

El acceso al Data Logger está restringido a un usuario y la autenticación a los servicios de este se realiza mediante un token generado de manera aleatoria y con fecha de caducidad. Dados los requerimientos del dispositivo no se consideró necesario implementar métodos más complejos de autenticación como JWT.

El cliente consiste en una aplicación web autónoma con su propia base de datos SQLite. SQLite es una base de datos embebida que no requiere de un servidor ni de dependencias externas [3]. El servidor fue implementado utilizando Nest.js 8.0.1.

C. Visualización de datos

La interfaz de usuario de la aplicación fue desarrollada utilizando el microframework web progresivo Vue.js 2 y Quasar V1.

En la Fig. 3 se muestra la interfaz principal de la aplicación.

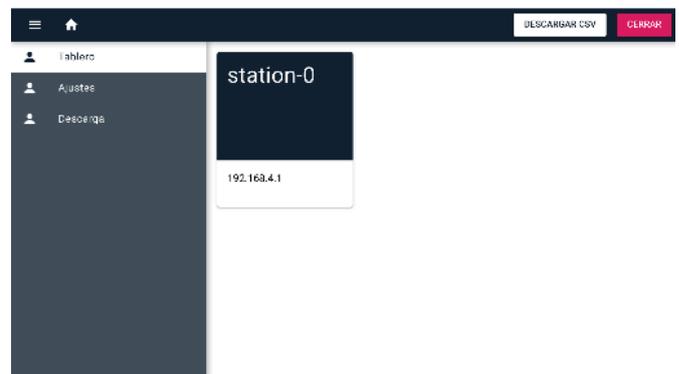


Figura 3. Interfaz del cliente del Data Logger .

La aplicación permite la gestión simultánea de distintos Data Loggers e identifica de manera individual las lecturas de cada uno. Mientras la conexión con el dispositivo permanece abierta se actualiza en tiempo real la información en el terminal del cliente.

Haciendo uso de la biblioteca apexcharts 3.28.3 se muestra la información almacenada en el cliente. Es posible seleccionar rangos específicos de tiempo en el que se desea visualizar una muestra. La aplicación también provee de la posibilidad de descargar los datos en el formato estándar CSV.

En la Fig. 4 aparece reflejado este proceso.

V. CONCLUSIONES

Como resultado final de este proyecto obtuvimos un sistema completo para la adquisición, almacenamiento y envío de

Variable medida	Unidad de medida
Temperatura	°C
Humedad	%
Velocidad del viento	m/s
Dirección del viento	°
Presión atmosférica	hPa
Precipitaciones	mm
Radiación	W/m ²

Tabla II
PARÁMETROS METEOROLÓGICOS REGISTRADOS CON EL DISPOSITIVO.



Figura 4. Flujo de información del sistema.

datos al ordenador, el cual está listo y en su aplicación final, sirviendo como Data Logger en el Iagric, como se aprecia en la Fig. 5.



Figura 5. Data Logger desplegado junto a una estación.

El sistema implementado tiene características esenciales que le permiten ser muy competitivo en el mercado, entre sus principales características se destacan su bajo costo, tamaño y peso reducido, capacidad de ser portátil, el uso de tecnologías abiertas, capacidad de almacenamiento, servidor web, etc.

Todos los componentes del equipo fueron seleccionados según su disponibilidad en el Mercado internacional y su bajo costo, garantizando así una fácil reproducción.

AGRADECIMIENTOS

Agradecemos a la D.Sc. Ivón Oristela Benítez González y a la Mtr. Carmenu Regueiro Busoch. También queremos agradecer al Grupo de Robótica y Mecatrónica de la CUJAE y al Instituto de Investigaciones de Ingeniería Agrícola (IAgric) por darnos la oportunidad de trabajar en este proyecto.

REFERENCIAS

- [1] NATIONAL INSTRUMENTS, “Data Logging - NI,” 2022. [Online]. Available: <https://www.ni.com/en-us/innovations/industrial-machinery/data-logging.html>
- [2] A. S. Abdelfattah, T. Abdelkader, and E.-S. M. El-Horbaty, “RAMWS: Reliable approach using middleware and WebSockets in mobile cloud computing,” *Ain Shams Engineering Journal*, vol. 11, no. 4, pp. 1083–1092, Dec. 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2090447920300861>
- [3] J. Zhang and X. Chen, “Research and Design of Embedded Wireless Meal Ordering System Based on SQLite,” *Physics Procedia*, vol. 25, pp. 583–587, Jan. 2012. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1875389212005457>

Sistema de Adquisición de Datos utilizando una CIAA-Safety

Juan Pablo RUMIE VITTAR
Centro I+D de Tec. Aeronáuticas
Fuerza Aérea Argentina
Las Higueras, Córdoba, Arg.
jvittar@faa.mil.ar

Dario Walter DIAZ
Grupo de Sistemas de Tiempo Real
Universidad Nacional Río Cuarto
Río Cuarto, Córdoba, Arg.
ddiaz@ing.unrc.edu.ar

Gustavo Miguel RODRIGUEZ
Grupo de Sistemas de Tiempo Real
Universidad Nacional Río Cuarto
Río Cuarto, Córdoba, Arg.
grodriguez@ing.unrc.edu.ar

Francisco Martín ESCOBAR
Grupo de Sistemas de Tiempo Real
Universidad Nacional Río Cuarto
Río Cuarto, Córdoba, Arg.
mescobar@ing.unrc.edu.ar

Diego Daniel BADINO
Grupo de Sistemas de Tiempo Real
Universidad Nacional Río Cuarto
Río Cuarto, Córdoba, Arg.
dbadino@ing.unrc.edu.ar

Ariel Cristian PRINCIPI
Centro I+D de Tec. Aeronáuticas
Fuerza Aérea Argentina
Las Higueras, Córdoba, Arg.
aprincipi@faa.mil.ar

Juan Octavio OVIEDO
Centro I+D de Tec. Aeronáuticas
Fuerza Aérea Argentina
Las Higueras, Córdoba, Arg.
joviedo@faa.mil.ar

Damian Hector PRIMO
Grupo de Sistemas de Tiempo Real
Universidad Nacional Río Cuarto
Río Cuarto, Córdoba, Arg.
dprimo@ing.unrc.edu.ar

Diego Salvador FUSARI
Fuerza Aérea Argentina
Las Higueras, Córdoba, Arg.
salvadordiegofusari@gmail.com

Resumen—En el presente trabajo se expone el análisis, diseño, desarrollo e implementación de un sistema de adquisición de datos provenientes de sensores varios, a través de un micro controlador ARM® CortexTM-R4F CPU que satisface normas de seguridad crítica, tales como IEC61508, ISO26262 y DO-178 (a nivel Software) correspondiéndose al factor de forma LQFP modelo TMS571LS12227LS sobre una placa CIAA-Safety[1]. Los sensores son: un GNSS (Sistema Global de Navegación por Satélite), una IMU (Unidad de Medición Inercial) y sensores de presión atmosférica y temperatura. El micro controlador ARM fue montado sobre una placa de desarrollo denominado CIAA-Safety y su programación fue realizada bajo el formato BAREMETAL, es decir, sin Sistema Operativo en Tiempo Real

Palabras Claves—Seguridad Crítica, GNSS, IMU, UML, MISRA-C, RTOS, Normas ISO

I INTRODUCCIÓN

En el marco de un proyecto de investigación propuesto por Fuerza Aérea Argentina en conjunto con la Dirección General de Investigación y Desarrollo mancomunados en proyectos I+D, y el Grupo de Sistema de Tiempo Real (GSTR), pertenecientes a la Facultad de Ingeniería de la Universidad Nacional de Río Cuarto (UNRC) surge la necesidad de desarrollar un sistema de software para realizar la *captura* y *transmisión* de información provenientes de sensores de *posicionamiento* y *ubicación global*, (GNSS), presión atmosférica e información de actitud y aceleraciones provista por una unidad denominada *Inertial Measurement Unit* (IMU). Para lograr este objetivo, se utilizó un GNSS fabricado por la Empresa SparkFun Electronics® modelo *Venus638FLPx1*, una IMU de Adafruit® modelo BNO055 que posee un micro controlador marca BOSCH y un sensor de temperatura y presión atmosférica modelo BMP180. El sistema de software desarrollado, esta alojado en una placa electrónica prototipo, denominada CIAA-Safety (Figura 1), este desarrollo sigue las buenas practicas¹ recomendadas por los estándares de seguridad funcional para aplicaciones críticas. El proyecto CIAA-Safety esta fuertemente orientado a la industria y a las PyME industriales (entre otros) de nuestro país que puedan acceder libremente a casos de

estudio que les permita comprender los pasos para elaborar un sistema electrónico capaz de certificar estándares internacionales de seguridad funcional. Para alcanzar este objetivo, el proyecto CIAA-Safety esta impulsado por casos reales de uso industrial a partir de casos de uso de sectores aeronáuticos, ferroviarios, aeroespaciales, automotrices, industria de procesos, etc.



Figura 1: Versión final CIAA-Safety

El proyecto abierto CIAA-Safety, tiene como desafío establecer e impulsar una metodología de desarrollo y una plataforma tecnológica de referencia para el desarrollo de sistemas de seguridad crítica, en Argentina. En otra arista, el desarrollo, evolución y funcionamiento de sistemas embebidos de seguridad crítica se enfrentan a cuatro retos principales:

- Cumplir con los requisitos y expectativas del usuario, tanto funcionales como no funcionales.
- Gestionar la complejidad (cada vez mayor).
- Proporcionar fiabilidad demostrable (seguridad crítica, disponibilidad e integridad).
- Encontrar métodos eficientes para el desarrollo y la evolución de los sistemas.

¹ <http://polux.unipiloto.edu.co:8080/00001881.pdf>

La CIAA-Safety [1] posee un micro-controlador ARM-CortexR4F® (modelo TMS5701227LS®) en el cual (y es el único que esta certificado por el fabricante, el resto del hardware utilizado, debe someterse a auditorías de certificación) cumple las normas de seguridad crítica (IEC61508, ISO26262) mínimas y necesarias. El análisis y diseño del software embebido en el micro-controlador se realizo utilizando lenguaje unificado de modelado UML[4] y bajo pautas en materia de desarrollo de software relacionadas a “*buenas practicas de programación*” acentuadas sobre el lenguaje de programación C, denominada MISRAC².

II OBJETIVOS

La placa de desarrollo y múltiple propósito CIAA-Safety, se diseñó [3] con el objetivo de brindar una plataforma para desarrollos que sean canalizados en materia de seguridad funcional y también, seguridad crítica. De esta manera, se utilizó como plataforma de hardware base para ensamblar electrónica (Poncho) con componentes (GNSS, IMU y Sensor de presión atmosférica y temperatura) para lograr un prototipo denominado Sistema Registrador de Adquisición de Parámetros Portátil; SIR-APP. Este sistema puede ser configurado de forma autónoma mediante la utilización de baterías de tipo LIPO o energizado, en contrapartida, a la instalación eléctrica del vehículo móvil que se lo instalese.

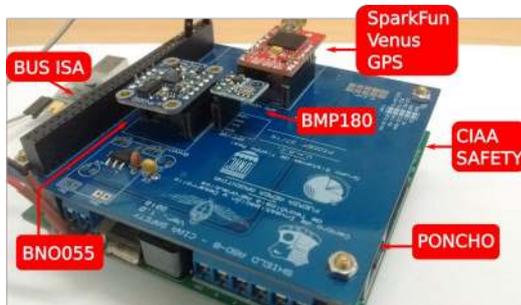


Figura 2: CIAA-Safety + Placa Poncho

El desarrollo de este prototipo, surge de la necesidad de contar con un dispositivo que permita, entre otros usos, registrar vuelos de aeronaves, ejercicios sobre vehículos terrestres, también en el ámbito ferroviario, entre otros diversos usos; con la finalidad de mostrar y representar lo registrado en un sistema visual sobre diferentes tipos de topologías de mapas como así también en componentes en 3D con el propósito de otorgar valor agregado en áreas de simulación, entrenamiento y aprendizaje, tanto en la capacitación de personal de la FAA, como así también a los integrantes del GSTR, como los usuarios finales.

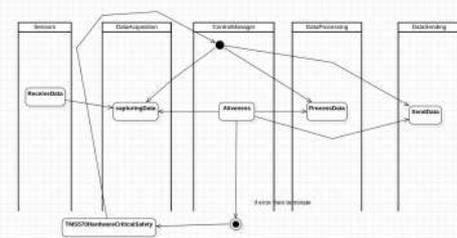


Figura 3: Diagrama de componentes

Lógicamente, en cuanto el Hardware (a través de la placa Poncho) como Software mediante el diseño adaptado de la placa poncho como así también del desarrollo de Software a las necesidades de futuros usuarios finales.

El proceso de especificación y desarrollo de software se realizó siguiendo los punto ordenados a continuación:

- 1 Estudio de Artefactos de Análisis definiendo así los componentes de alto nivel.
- 2 Estudio de Artefactos de Componentes de Diseño, a partir de los artefactos de Análisis.
- 3 Desarrollo del Software sobre cada componente de diseño.
- 4 Aplicación de lineamientos MISRA-C.
- 5 Integración al hardware CIAA-Safety.

III DESARROLLO

I.A Desarrollo del Software

Habiendo realizado un análisis de todos componentes involucrados y tomar medidas en base al funcionamiento de cada sensor, como así también, los componentes de diseño desarrollados por la expansión de los componentes de casos de uso y análisis, respectivamente, de manera modular y adaptable, computacionalmente hablando de una forma sencilla, el desarrollo completo se llevó a cabo con la librería correspondiente a cada sensor involucrado en el sistema (BNO055, BMP180 y GNSS) sobre el lenguaje C utilizando a la IDE CCS (*Code Composer Studio*) sobre el micro controlador TMS5701227LS®. Para ello, se explicará como se desarrolló cada sección que interviene cada sensor, de forma modular en librerías:

- 1 Registros propios (de Hardware) del sensor.
- 2 Inicialización, bus de comunicación y calibración del Sensor.
- 3 Obtención de datos en unidades Euler y Conversión de datos Euler en Unidades de Ingeniería (EU).

A.1 *Registros de Hardware*: En la presente sección, se muestran algunos registros definidos para la unidad inercial, utilizando para este caso, los registros Euler.

Dichos registros propio de la electrónica de IMU BNO055, son los necesarios para escribir en ellos, y obtener como respuesta los valores de, por ejemplo, actitud, aceleración, magnetómetro, etc.

A.2 *Inicialización, bus de comunicación y calibración del Sensor*: El bus de comunicación seleccionado para cada uno de los sensores involucrados son:

- 1 BNO055 y BMP180 – Bus I2C
- 2 GNSS – Bus Serial

Para ejemplificar un caso muy particular, y siguiendo con el ejemplo anterior de la BNO055, el bus de comunicación I2C envía escrituras de palabras de datos de 8 bits (llamada función *write8*), de tipo *unsigned int*.

La inicialización y calibración se realizan en conjunto, primero se establece la comunicación inicial con el sensor, modo de configuración normal, esto es, envío de datos de actitud, aceleraciones y magnetómetro en los tres ejes (x,y,z), se reinicializa la placa (para que tome la configuración previamente enviada) y se realiza una espera de segundos, hasta que el sensor responda con un protocolo predeterminado para dar respuesta a que esta en funcionamiento, luego se procede a deshabilitar el funcionamiento de bajo consumo enviando o configurando al BNO055 bajo *power mode: normal*, y se obtiene el ID de dirección de paginado y por último el modo de funcionamiento a través de respuestas de datos a consultas por direcciones de registros. Además, se configura la utilización del sistema de disparo *Trigger* de datos sobre direcciones de memoria, a partir de los registros de datos propios del sensor.

² <https://www.misra.org.uk/>

A.3 Obtención de datos en unidades Euler y conversión de datos Euler en unidades de Ingeniería: Continuando con el ejemplo de la BNO055 (Para el resto de los sensores el proceso es análogo por lo que no se expondrá en este documento para evitar extender demasiado el mismo) los datos en unidad Euler, se obtienen a partir de la solicitud de los registros de orientación (recordar que hay tres tipos: Orientación, Aceleración y Magnetómetro) la información de estos registros, debe dividirse en los correspondientes 3 ejes (x,y,z) y luego se los almacena en unidades de Euler en un único vector. Posterior a ello, se transforma dicho vector en EU convirtiendo los datos de Euler a grados en los tres ejes (x,y,z). De forma similar, se realiza las conversiones correspondientes de los datos de registro de aceleración del sensor, a aceleración en EU y datos de registro de magnetómetros a intensidad de campo magnético en EU.

Cabe aclarar que para el sensor GNSS se implementó un componente genérico para el modelo que se dispone GNSS SparkFun que no dependiera del modelo escogido, es decir, puede ser reemplazado por otros modelos.

I.B MISRA-C

Es un conjunto de recomendaciones o directrices para el desarrollo de software en lenguaje C desarrollado por The Motor Industry Software Reliability Association (MISRA). Su finalidad es proveer portabilidad, seguridad y fiabilidad al código fuente en el contexto de software embebido. Si bien, MISRA-C no es un estándar abierto y los documentos con las directrices se adquieren mediante la compra del mismo, cuando se actualiza la documentación, queda a libre disposición la versión anterior a la última vigente, siendo esta versión libre la utilizada para su análisis y aplicación en el presente trabajo. MISRA-C, básicamente, es una guía de reglas para clasificar un código fuente según los siguientes escalafones:

- Reglas/directivas preceptivas (*mandatory*).
- Reglas/directivas requeridas (*required*).
- Reglas/directivas recomendadas (*advisory*).

En base a estos lineamientos, se procedió a ejecutar las reglas de software desarrollado. Se mostrará a continuación, algunos de los resultados encontrados, a modo de ejemplo:

- MISRA-C:2004 6.3/A: Los *typedefs* que indican tamaño y signo se deben utilizar en lugar de los tipos numéricos básicos.
- MISRA-C:2004 5.2/R: Los identificadores en un ámbito interno no deben usar el mismo nombre que un identificador en un ámbito global, y por lo tanto ocultar ese identificador *scale*.
- MISRA-C:2004 10.1/R: El valor de una expresión de tipo entero no se convertirá implícitamente en un tipo subyacente diferente si no es una conversión a un tipo entero más amplio con el mismo signo.
- MISRA-C:2004 9.1/R: Todas las variables deben ser inicializadas con un valor antes de ser utilizadas (*variable quat*).
- MISRA-C:2004 17.4/R: La indexación de matrices será la única forma permitida de aritmética de punteros.

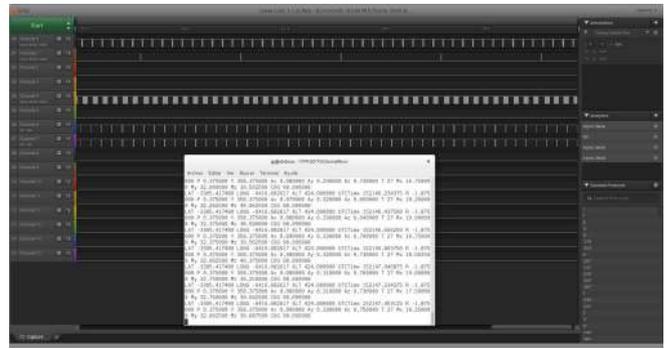


Figura 4: Ejemplo de Funcionamiento - Ensayo

IV CONCLUSIÓN Y TRABAJO A FUTURO

El objetivo de este trabajo de desarrollo de Software sobre una plataforma de Hardware desarrollada, por cierto, en nuestro Centro I+D, da el puntapié inicial para nuevos desafíos de desarrollo y mejoras completamente bajo normas de seguridad crítica y con la posibilidad de realizar la gestión para la certificación. Este Hardware, cumplirá con las prestaciones estipuladas [2] en los lineamientos iniciales y el Software implantado es el mismo que el que se viene desarrollando en otros productos, por ejemplo, utilizando hardware de la empresa TI (*Texas Instruments*), que inicialmente se utilizaron en paralelo mientras se modelaba y fabricaba la placa multi propósito CIAA-Safety. Esta buena adaptación de Software desde los productos de TI a la placa CIAA-Safety se logro gracias a las técnicas de desarrollo de Software y buenas practicas de programación (captura, análisis, diseño, desarrollo, implantación y test) utilizadas a lo largo de todo el proyecto. Para esto, se realizó inicialmente una versión *Baremetal* (sin Sistema Operativo) cumpliendo en ambos casos la Guía de MISRA-C. A Futuro, se prevé una versión sobre un sistema operativo RTOS certificable, para lo cual, se debe gestionar su compra.

Como fortaleza, entre los ya mencionados, cabe aclarar que no se utilizó manejo de interrupciones (síncronas o asíncronas), y tampoco manejo de excepciones, dado que estas técnicas y/o metodologías de software y hardware van en contra de la seguridad crítica.

Los objetivos a futuro, se prevé modificar, en primera instancia, el frame de salida (trama) de datos, para enviar un frame que corresponda al formato SBS (*Surveillance Broadcast Services*) que se utiliza por el ATC (*Aircraft Traffic Control*) para intercambiar información proveniente de un transponder ADS-B hacia otros ATC sobre la red de satelites de posicionamiento global. Y en segunda instancia, transformar el prototipo en formato ADS-B, es decir, frames de (tramas) de 112 bits como lo exige el estándar ADS-B con el objetivo de convertirlo en un transponder ADS-B. En ambos casos, se eliminan completamente los sensores IMU, temperatura y presión atmosférica, porque el estándar ADS-B se basa en la utilización del sensor que brinda GNSS.

REFERENCES

- 1 Computadora Industrial Abierta Argentina para Aplicaciones de Seguridad Crítica - CIAA Safety, Proyecto CIAA, ACSE, CADIEEL. Marzo 2022. [Proyecto CIAA](#).
- 2 D.Diaz, G. Rodriguez, R. Manno, J. Amor, D. Fusari, 2018. Análisis de falla del Modelado de la Computadora Industrial Abierta Argentina de Seguridad Crítica bajo AADL, 5to Congreso Argentino de Ingeniería Aeronáutica. pag 210-216.
- 3 D.Diaz, G. Rodriguez, R. Manno, J. Amor, D. Fusari, 2018. Modelado de la Computadora Industrial Abierta Argentina de Seguridad Crítica bajo AADL, Simposio Argentino de Sistemas Embebidos. 2018. ISBN: 978-987-46297-4-6.
- 4 Rumbaugh, Jacobson, Booch, 2005. El Lenguaje Unificado de Modelado Manual de Referencia Segunda Edición, Addison-Wesley.

Herramientas de apoyo al estudio de sistemas embebidos utilizando uModelFactory

Almaraz, Nicolás Tobías
Estudiante de Ing. Electrónica
UTN FRBA
CABA, Argentina
nalmaraz@frba.utn.edu.ar

Nirino, Felipe
Estudiante de Ing. Electrónica
UTN FRBA
CABA, Argentina
fnirino@frba.utn.edu.ar

Resumen — A partir del trabajo realizado desde el departamento de ingeniería electrónica de la UTN FRBA este equipo de trabajo desarrolló un software llamado uModelFactory. El mismo permite el diseño, depuración y simulación de máquinas de estados finitas tipo Moore. Dicha herramienta fue concebida para ser utilizada a la hora de diseñar sistemas embebidos. Es por ello que la cátedra de la asignatura Informática II decide implementarla como herramienta didáctica. Sin embargo, los datos provenientes de encuestas realizadas al alumnado revelan que este software no tenía una interfaz lo suficientemente intuitiva o cómoda para desarrolladores noveles. Es por ello, que el equipo de desarrollo decide lanzar una sección de ayuda dinámica que facilite el acceso a la información acerca de conceptos teóricos. Para esto se implementa un sistema con tres niveles de ayuda que muestran información acerca de lo que el cursor del mouse esté apuntando. Los resultados de esta nueva funcionalidad fueron positivos y motivan a futuros desarrollos.

Palabras Clave — Máquinas de estados, Sistemas embebidos, Enseñanza y Aprendizaje.

I. INTRODUCCIÓN

El software “uModelFactory” [1] se presenta como una herramienta para asistir en la elaboración de diagramas y Máquinas de Estado Finitas (MEF) tipo Moore gobernados por eventos. A partir de trabajos anteriores, se ha analizado el uso de la misma en el ámbito educativo [2], particularmente en la cátedra de Informática II de la Universidad Tecnológica Nacional - Facultad Regional Buenos Aires, como una tecnología que facilite la enseñanza y el aprendizaje del proceso de diseño y diagramado de MEF con complejidad creciente.

De la retroalimentación obtenida por parte del alumnado general, recolectada a partir de instrumentos como encuestas cerradas o de las opiniones expresas por parte de los respectivos docentes, se realizó un análisis de las falencias, defectos y limitaciones a la funcionalidad del software. Los resultados revelaron en esta instancia una prominente manifestación de falta de intuitividad y de facilidad de uso que perjudica la apropiación de conocimiento y causa frustraciones en el conjunto de los estudiantes.

Teniendo en cuenta la dificultad, se optó por incluir un apartado de ayuda dinámica que favorezca la interacción con el programa y presente al usuario diferentes niveles de información relativa a los elementos que componen los diagramas.

II. MARCO TEÓRICO

A. Concepción de máquinas de estados

Una máquina de estados es una estrategia de modelizar el comportamiento de cualquier tipo de sistema (por ejemplo, una cafetera, un proceso de fabricación industrial, las condiciones de aprobación de un examen, etc).

El proceso de diseñar una máquina de estados consiste en dividir el sistema en diferentes instancias llamadas “estados” que cada uno de ellos representa una etapa del proceso. Debido a las excitaciones o estímulos de entrada podremos pasar de un estado a otro. A este cambio de estado se lo denomina transición y aquí se deben aclarar qué salidas del sistema deben cambiar frente a la excitación que la produce.

B. Componentes de una máquina de estados

Las máquinas de estados pueden descomponerse en cuatro pilares esenciales (Fig. 1):

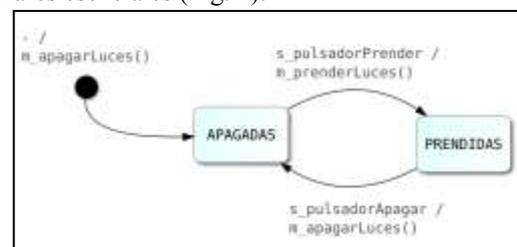


Fig. 1. Diagrama con los pilares de una máquina de estados elaborado en “uModelFactory”

- Estados: Son las diferentes etapas por las que tiene que pasar un sistema para completar un proceso. Para avanzar de un estado a otro se hace a través de las transiciones. En los diagramas son representados por recuadros (estados “prendidas” y “apagadas”).
- Transiciones: Es el paso de un estado a otro. Estas se dan cuando se cumplen ciertas condiciones de entrada (eventos) y son representadas por flechas que van desde un estado de origen a un estado de destino. Se dice que una transición se dispara frente a un evento determinado. Por ejemplo: Pasar de “prendidas” a “apagadas” al recibir el evento “s_pulsadorApagar”. En estas transiciones se modifican las salidas o alguna variable interna en caso de ser necesario (acciones). A la hora de representarlas en el diagrama se debe detallar “eventos / acciones”, es decir, qué evento la disparó y qué acción debe ejecutarse. Por ejemplo, si se presiona el pulsador de apagado, entonces apagar las luces.
- Eventos: Representan los estímulos externos o internos que recibe la máquina de estados, y a los que la misma responderá. Por ejemplo, la pulsación de un botón que resulta en la generación de un evento “s_pulsadorApagar”.
- Acciones: Determinan los comportamientos al producirse una transición de un estado a otro. Por ejemplo, encender o apagar las luces descritas en los ejemplos anteriores.

C. Estados compuestos y máquinas de estados concurrentes

Los sistemas a describir pueden tener un alto nivel de complejidad, por lo que suelen agregarse a los pilares fundamentales herramientas más elaboradas, como ser los estados compuestos (estados que poseen máquinas de estados dentro de sí) y las máquinas concurrentes (aquellas que coexisten en un mismo período temporal, procesando eventos en paralelo).

Estas dos configuraciones son soportadas por “uModelFactory”, permitiendo no solo el uso sino la simulación de las mismas, presentándose entonces como un software que goza de un gran poder de aplicación.

D. Áreas de aplicación

Uno de los campos en los que se utiliza la metodología de máquinas de estado es el de los sistemas embebidos, donde deben describirse sistemas que respondan a estímulos y lleven a cabo determinadas tareas. Este método permite ordenar el procedimiento de cada tarea a realizar, facilitando la implementación del sistema y reduciendo la posibilidad de fallos o comportamientos erráticos.

El software “uModelFactory” ha sido diseñado especialmente para dicho campo, contando con la posibilidad tanto de simular la ejecución como de generar el código necesario para la correcta implementación en un microcontrolador programable [3].

III. MATERIALES Y MÉTODOS

Las opiniones del alumnado han sido registradas por los docentes de cada curso a partir de encuestas anónimas, de manifestaciones expresas de los alumnos, e inferencias basadas en consultas y dudas recurrentes durante el periodo de clases. La encuesta anónima incluía tanto aspectos personales (tiempo dedicado a estudio, facilidad de seguir las explicaciones del docente, posesión de hardware necesario para correr el software, etc.) como aspectos relativos a facilidad de uso y efectividad en la comprensión de conceptos a partir del uso de la plataforma. Ejemplos de

algunas de las preguntas utilizadas se presentan a continuación:

- ¿Cómo calificarías el uso del software para el modelo de MEF?
- ¿Cómo calificarías su interfaz gráfica y experiencia de usuario?
- ¿Cómo calificarías la generación de código brindada?

Con esta información, se procedió a elaborar un sistema de ayuda que brinde información sobre conceptos clave del marco teórico asociado a los diagramas o las MEF. Para determinar su contenido, el cuerpo docente relevó información detallada sobre los diferentes cursos. A esta se aplicó un filtro con el fin de obtener las críticas hechas pura y exclusivamente a las características y las formas de presentación del software.

Analizando los resultados se optó para proceder un desarrollo segmentado en torno a tres conceptos principales elaborados a continuación: División cualitativa de los temas de ayuda en acotados grupos o “niveles”, fácil traducción y/o alteración, y una interfaz efectiva e intuitiva.

Para el desarrollo del proyecto se utilizó el framework de Qt. La elección fue dada por contar con una amplia variedad de bibliotecas fácilmente adaptables para resolver las diferentes problemáticas que se describen. Por otro lado, tiene una muy detallada documentación y ejemplos que facilitan la tarea del programador.

IV. RESULTADOS

En la captura de pantalla mostrada en la Fig. 2 se puede ver que cuando el mouse se encuentra sobre los diferentes ítems de la máquina en la ventana de la derecha nos mostrará la información referida a lo que se encuentre en la posición del cursor. A su vez, en la Fig. 3 se detalla la ayuda correspondiente a los elementos seleccionados del diagrama.

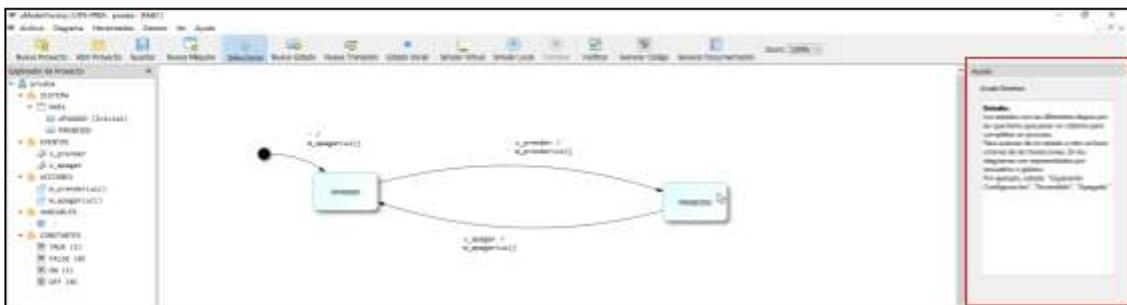


Fig. 2. Captura de pantalla de la aplicación con el panel de ayuda desplegado y resaltado en rojo.

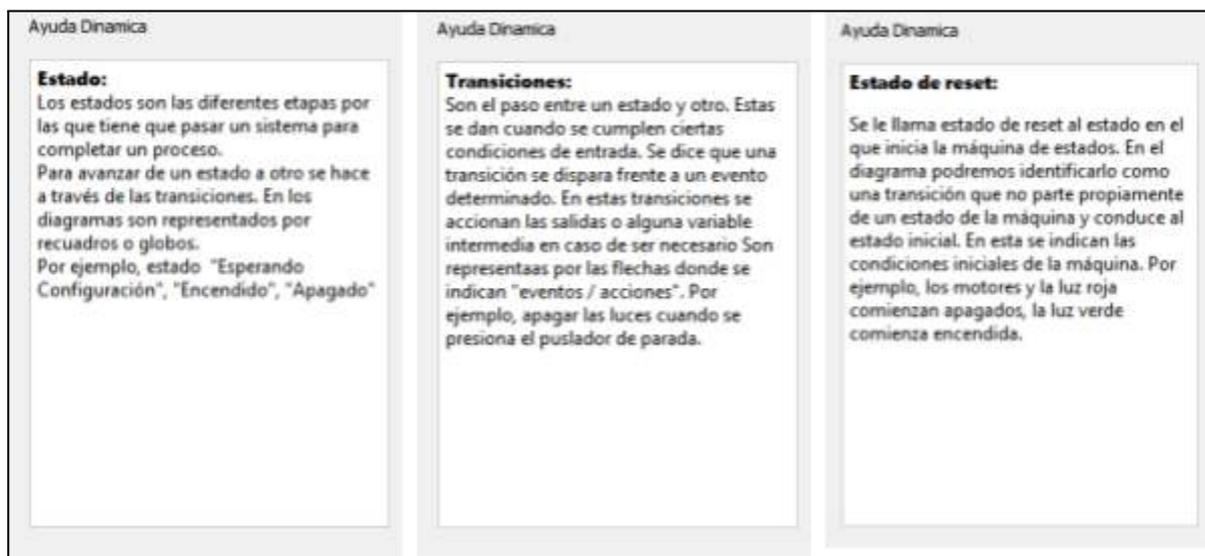


Fig. 3. Capturas de diferentes textos de ayuda. De izquierda a derecha: Estado, Transiciones y Estado de reset o estado inicial

A. División cualitativa de la ayuda

Desde el inicio se planteó la necesidad de poder alterar la cantidad de ayuda mostrada, con el principal objetivo de permitir un avance progresivo en términos educativos del alumno y permitir a usuarios intermedios o avanzados obtener una referencia rápida de conceptos clave que pudiesen ser olvidados.

Se confeccionaron tres categorías o niveles de visualización de ayuda nombrados de acuerdo a la clasificación del usuario: “Básico” (el nivel más abarcativo, útil para los primeros usos), “Medio” (reduce ciertos aspectos del nivel Básico, apropiado para usuarios intermedios) y “Avanzado” (provee una referencia rápida y limitada). Además, se incluye una opción para desactivar completamente el subsistema de ayuda, lo que brinda mejores posibilidades en cuanto a organización del entorno de trabajo.

Esta configuración podremos hacerla desde la barra de herramientas en la opción de ayuda (Fig. 4).

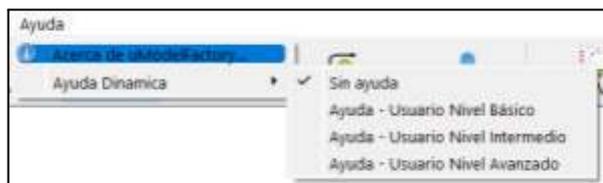


Fig. 4. Menú desplegable con opciones de ayuda dinámica

B. Fácil traducción y/o alteración del contenido

Con este punto se buscó favorecer el proceso de modificación y mejora de las recomendaciones mostradas por la ayuda dinámica, incluyéndose en el diseño la posibilidad de llevar a cabo una traducción del software a otros idiomas. En pro de ello, se dictaminó el uso de un archivo con sintaxis XML (eXtensible Markup Language) [4][5] para almacenar tanto los textos de ayuda como metadatos correspondientes a versionado y lenguaje en el que se encuentran los mismos.

Se decidió utilizar la presente estructura dada por la sintaxis XML por la facilidad de implementación en cuanto a código (la complejidad agregada por otros mecanismos de codificación opaca las ventajas de los mismos respecto a un mejor aprovechamiento de recursos como espacio de almacenamiento), la claridad intuitiva con la que permite organizar la información, y la facilidad en términos de edición (los archivos XML se agrupan dentro de los archivos de caracteres, donde la información se encuentra codificada en forma de texto y son legibles por cualquier persona sin la necesidad de utilizar programas especializados).

C. Interfaz intuitiva y efectiva

Luego de probar con varios diseños se optó por implementar la ayuda en un panel (una sección vertical presente dentro de la ventana del programa principal) pasible de ser movida y redimensionada por el usuario a gusto, apelando al confort del mismo y evitando frustraciones innecesarias por una interfaz no adaptable.

En función de la posición del mouse en el diagrama se presenta un mensaje de ayuda dinámica con información del objeto señalado por el mouse. En nuestro caso podremos encontrar el estado de reset, estados y transiciones. El mensaje cambiará en función del nivel de ayuda seleccionado.

D. Devolución por parte de los alumnos y ex-alumnos

Los resultados de las encuestas a los ex-alumnos y/o alumnos recursantes de la materia fueron que con la funcionalidad de ayuda dinámica podría haber sido mucho más fluida la etapa de aprendizaje.

Por otro lado, los alumnos que cursaron la asignatura con esta nueva versión de uModelFactory no manifestaron problemas o sugerencias de accesibilidad a la información en las encuestas.

V. CONCLUSIONES Y TRABAJOS A FUTURO

Dado que el impacto de estas nuevas funcionalidades fueron positivos se llega a la conclusión de que cuanto mayor sea la cantidad de herramientas que el alumno tenga, mejor será su rendimiento.

En consecuencia, esto motiva a desarrollar nuevas herramientas que faciliten el proceso de aprendizaje. De hecho, actualmente, se está trabajando en un menú de ayuda que cuente con un buscador de tal manera que allí se pueda acceder fácilmente a una descripción de cada funcionalidad de la aplicación. Así como diferentes aspectos teóricos.

Al igual que en la ayuda dinámica se está implementando con un archivo XML para añadir la posibilidad de configurar el idioma en versiones posteriores, además de contar con un sistema de hipervínculos que puedan facilitar información acerca de herramientas que podrían llegar a ser útiles según la funcionalidad buscada.

Al igual que en las herramientas de ayuda ya desarrolladas, estas nuevas herramientas serán configurables

para que se adapten en función de las necesidades del usuario y no estorben en caso de no ser requeridas.

REFERENCIAS

- [1] L. Sugezky, M. Prieto, N. González, M. Giura, Y. Kuo, M. Trujillo, J.M. Cruz. "Desarrollo e implementación de herramientas de simulación de modelos para sistemas embebidos". Congreso Argentino de Sistemas Embebidos, 2016.
- [2] N. González, L. Sugezky, M. Prieto, M. Giura, Y. Kuo, M. Trujillo, J.M. Cruz. "Evaluación del software uModelFactory como herramienta didáctica". IEEE Argencon, 2016.
- [3] N. Gonzalez, J. Cruz, L. Sugezky, M. Giura, M. Trujillo, M. Prieto. "Analysis of a UML-based embedded system modeling software application". Congreso Argentino de Sistemas Embebidos, 2014.
- [4] B. Schnabel. "Enabling Language Translation with XML Tools and Standards". The Center For Information-Development Management, 2015. Disponible en: http://www.oasis-open.org/committees/xliff/documents/translation_w_xml_tools.pdf
- [5] A. Zydron, (2004). "Translating XML Documents with xml" Disponible en: <https://www.xml.com/pub/a/2004/01/07/xmltm.html>

Implementación de una maqueta de sistema de control con aplicaciones didácticas

Enrique Sergio Burgos
Laboratorio de Informática
U.T.N. Facultad Regional Paraná
Paraná, Entre Ríos, Argentina
sergioburgos@frp.utn.edu.ar

Francisco Alberto Sala
Laboratorio de Control
U.T.N. Facultad Regional Paraná
Paraná, Entre Ríos, Argentina
franciscosala@frp.utn.edu.ar

Carlos Alberto Cappelletti
Cátedra de Teoría de los Circuitos II
U.T.N. Facultad Regional Paraná
Paraná, Entre Ríos, Argentina
ccappelletti@frp.utn.edu.ar

Resumen—En este trabajo se presentan los avances realizados en la construcción de un objeto didáctico diseñado a fin de ser utilizado para la realización de experiencias en temáticas de Sistemas de Control en carreras de grado. Es gestionado a través de una red *WiFi* e incorpora interfaces de control a través de un *API Rest* y *web sockets*, permitiendo interactuar con *GNU Octave* como herramienta de cálculo.

Palabras claves:—*embeded systems, control systems, GNU Octave, IoT*

I. INTRODUCCIÓN

El uso de experimentos físicos como objetos didácticos, como una forma de aplicación de conceptos teóricos durante el proceso de enseñanza, ha generado antecedentes de éxito. Entre ellos, en [1] se presenta el uso de un kit basado en *Arduino* y un *shield* dedicado, utilizado para tratar tópicos propios de Control de Procesos. En [2] se presenta una variante, donde el objeto controlado es un cabezal de impresora 3D. A partir de estos y otros antecedentes similares, y considerando el aporte que produce la aplicación de conocimientos teóricos a casos de estudio prácticos, se ha dado origen al desarrollo de un sistema embebido para la realización de actividades prácticas relacionadas a tópicos de control de procesos en ingeniería. El dispositivo propuesto, no busca ser de tipo portable, sino más bien de laboratorio y para ser accedido utilizando redes de datos. La dirección tomada en este trabajo sigue las líneas consideradas en el desarrollo del sistema piloto de [3]. Dejando la posibilidad de ser integrado en el *framework* allí propuesto.

El objeto de estudio es un sistema térmico, donde una lámpara incandescente actúa como calefactor, un ventilador introduce perturbaciones al sistema y se realiza la realimentación utilizando una sonda de temperatura.

El trabajo se estructura del siguiente modo, en la sección II se presentan los objetivos particulares del desarrollo, en la sección III la estructura del *hardware* desarrollado, en la sección IV los elementos de software y finalmente en la sección V se presenta una breve discusión sobre algunos resultados encontrados.

II. OBJETIVOS DEL DESARROLLO

El objetivo principal del desarrollo es un equipo que permita la realización de actividades prácticas de identificación y control de sistemas utilizando diferentes técnicas. Para esto

la interfaz provista por el dispositivo permite la conexión con *GNU Octave* [4], siendo este software la herramienta de análisis utilizada en el abordaje de aplicaciones de control. Surge de aquí un objetivo secundario en el que se está trabajando actualmente, y que consiste en el desarrollo de paquetes utilizando *guiEditor* [5], mejorando la experiencia de usuario al incorporar interfaces gráficas.

Dentro de los objetivos considerados también se encontraba que el control de potencia se realice con alguna de las técnicas clásicas de control de potencia para cargas de 220 [V]. Esto también genera un objetivo indirecto, que el sistema sea accesible totalmente de modo inalámbrico a fin de eliminar cualquier potencial riesgo al ser manipulado por estudiantes.

El dispositivo ha sido diseñado de modo que, una persona por vez, pueda interactuar y realizar las experiencias de interés. Esto requiere que al ser utilizado en cursos de grado, se disponga de varios de estos dispositivos para permitir que cada estudiante pueda realizar las prácticas que correspondan. Para lograr esto, es significativo que el dispositivo completo sea fácilmente reproducible y de bajo costo.

III. HARDWARE

El dispositivo ha sido diseñado de modo de poder ser construido utilizando impresión 3D y elementos accesibles en el mercado. En la Fig. 1 puede observarse la estructura del mismo. Consta de 3 placas unidas a través de 4 barras roscadas. La placa inferior obra de sustento del porta lámpara y, de modo similar, la placa superior fija el ventilador. La placa intermedia, fue diseñada originalmente para contener el sensor de temperatura dentro de un bloque de aluminio, optándose posteriormente por vincular directamente el sensor a la superficie de la lámpara, manteniendo la placa intermedia para aumentar la resistencia mecánica al dispositivo.

En la parte posterior se incorpora un gabinete que, adosado a las barras roscadas, contiene los dispositivos electrónicos del sistema. Estos incluyen un *SoC Esp32* de la empresa *Espressif*, un *Cortex M3* de la empresa *ST Microelectronics* (*STM32F103C8T6 / blue pill*), un módulo mosfet basado en el *IRF520* y un módulo de control de potencia por ángulo de disparo.

El *SoC Esp32* es responsable de la interacción con el usuario a través de la conexión a la red *WiFi* y se comunica con el

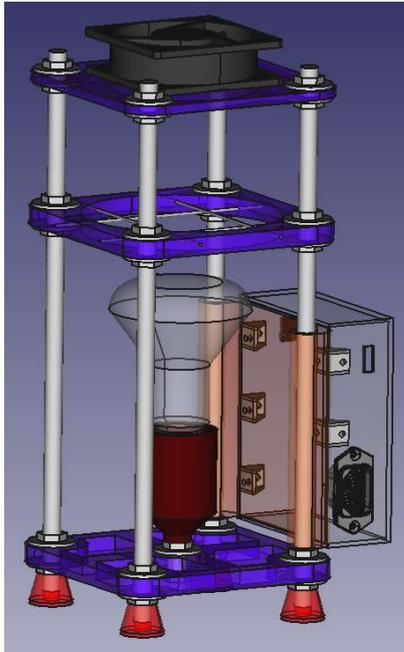


Figura 1. Aspecto del sistema de control desarrollado

Cortex M3 utilizando una interfaz *USART*, estando a cargo de este último la gestión de los actuadores y el sensor de temperatura. El control de potencia se realiza haciendo la detección de cruce por cero en la línea de energía alterna, comenzando la cuenta de un temporizador para determinar el momento en que debe dispararse un *triac*. Esta detección se realiza utilizando una interrupción, dependiendo de la potencia a entregar a la carga es el momento en el que se realiza el disparo. Desde el *Esp32* la potencia a entregar es comunicada como un valor entero, entre 0 y 100, que al ser recibido por el *Cortex M3* es ajustado para obtener una respuesta lineal y traducido a un valor de tiempo de disparo. El control del ventilador se realiza utilizando una salida de *PWM*, lo que permite modificar su velocidad de giro. Al igual que en el caso anterior se utiliza un valor entre 0 y 100 para configurar su velocidad de funcionamiento. La realimentación es lograda utilizando una sonda *PT100* de dos conductores conectada a un *MAX31865*, este dispositivo incorpora la interfaz analógica requerida por la sonda y un conversor analógico/digital de 15 bit. El control de este es realizado utilizando una interfaz *SPI*. Con el objetivo de lograr un mejor tiempo de respuesta, se optó por utilizar el modo de conversión continua enviando el último valor registrado cuando sea requerido. En favor de lograr un dispositivo fácilmente reproducible, se utilizaron módulos estándares para la construcción del dispositivo que se integraron en una placa pre perforada.

IV. SOFTWARE

El software del sistema está constituido por cuatro aplicaciones, que corresponden con el *firmware* de cada microcontrolador, la interfaz de configuración/monitoreo web y el desarrollo de las aplicaciones de comunicación desde *GNU Octave*.

La aplicación utilizada en el módulo *blue pill* fue desarrollada en lenguaje C, utilizando el entorno de desarrollo propuesto por el fabricante, *STM32CubeIDE* en su versión 1.8. La tarea a realizar por el *firmware* es controlar los actuadores, interactuar con la interfaz de la sonda de temperatura y comunicarse con el *SoC Esp32*. Para esta tarea se utilizó una máquina de estados implementada a través de un vector de punteros a funciones.

En *firmware* del *Esp32*, se desarrolló también en lenguaje C, utilizando *Visual Studio Code* junto al complemento *PlatformIO*. La implementación se realizó con los siguientes objetivos:

- Trabajar en modo mixto (*access point* y *station*), para proveer una red *WiFi* abierta que permita el acceso a dispositivos, más allá de si se configuró o no una red a la que conectarse.
- Implementar un *API REST* que permita interactuar con el dispositivo. Por compatibilidad con *GNU Octave*, la implementación se realizó utilizando los verbos *POST* y *GET*.
- Implementar comunicación basada en *web sockets*. La implementación se realizó de modo de proveer funcionalidades similares a las del *API REST*, pero agregando la posibilidad de acceder a los valores de temperatura a través de un flujo.

Se utilizó una parte de la memoria de programa como un sistema de archivos *spiffs* de modo que permita almacenar algunas opciones de configuración tales como la descripción del equipo y el uso de servicio de *DNS* dinámico.

Considerando que puede resultar de interés la reprogramación del *SoC*, se utilizó una variante de *OTA* (*Over The Air Updates*) a través del repositorio que contiene el proyecto. Para esto, el repositorio fue configurado utilizando *CI/CD* (*Continuous Integration / Continuous Delivery*) a fin de que, al realizarse modificaciones en el código fuente del *SoC*, este sea compilado en la nube. Durante el proceso de construcción de la aplicación, además de la imagen binaria del *firmware*, se genera un archivo *json* incluyendo información de la versión, fecha y hora de la compilación. Finalmente a través de la interfaz de gestión web incluida en el *Esp32* es posible disparar el proceso de reprogramación, descargando la nueva imagen generada en el repositorio.

La aplicación web, fue desarrollada utilizando *Bootstrap*, *JQuery* y *Chart.js*, incluyendo todos los archivos requeridos dentro de la imagen binaria del *firmware*. Su interfaz está conformada por 4 secciones a saber:

- Estado: presenta información relativa a la conexión de red, valor actual del sensor de temperatura, estado de los actuadores y una descripción del equipo.
- Configuración cliente: permite elegir la red *WiFi* a la que conectarse y da la posibilidad de configurar un servicio de *DNS* dinámico, indicando la *URL* de actualización y el certificado de seguridad para establecer la comunicación con el servidor. También incluye la posibilidad de agregar una descripción del equipo.



Figura 2. Aspecto de la interfaz web de gestión, presentando los valores de temperatura registrados.

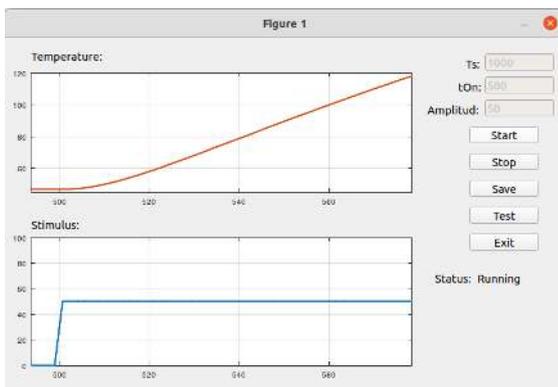


Figura 3. Interfaz de control desarrollada en guiEditor para GNU Octave

- Actualizar Firmware: presenta la versión de firmware utilizada por el SoC así como la disponible en el repositorio, dando la posibilidad de realizar la actualización.
- Monitor: desde aquí es posible observar los valores registrados de temperatura así como manipular los diferentes actuadores. En la Fig. 2 puede observarse su aspecto.

Finalmente, en la Fig. 3 puede observarse la interfaz de control desarrollada para *GNU Octave*. Desde aquí es posible acceder a la información registrada por el sistema así como modificar el estado de los actuadores. En esta interfaz solo se presentan un conjunto reducido de valores de temperatura, posibilitando su almacenamiento para un posterior análisis. El resultado de un experimento de excitación con un escalón del 50% de amplitud puede observarse en la Fig. 4.

Tanto la aplicación web como la de *GNU Octave* fueron desarrolladas en dos versiones, utilizando el *API REST* y *web sockets*. Para esto último, en el caso de *GNU Octave* se utilizó el paquete de *websockets* [6].

V. CONCLUSIONES

Un primer análisis que puede resultar de interés es el consumo de recursos que se obtuvo en los microcontroladores. La variante del *Esp32* utilizada incluye 4 MB de memoria *flash* y 327.680 bytes de *RAM*, de los cuales se utilizó el 87.9% (1.731.541 bytes) y el 12.1% (39564 bytes) respectivamente.

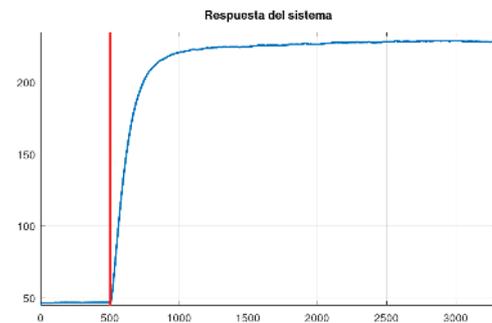


Figura 4. Resultados obtenidos en una experiencia una vez logrado estabilidad

Aquí debe considerarse que la *flash* fue particionada para utilizar *OTA*, teniéndose particiones de 1.970.176 bytes para almacenar el *firmware*. La partición utilizada para almacenar datos de configuración (*spiffs*) ocupa 64KB. En lo que a la memoria *RAM* respecta, el valor indicado corresponde con el espacio utilizado por variables globales y estáticas, no siendo un indicador completo ya que la implementación utiliza asignación de memoria dinámica. En el *STM32F103* se utilizó el 9.73% de memoria *RAM* (1.95 KB) y 21.72% de memoria *flash* (13.9 KB). De los valores observados puede concluirse que aún se cuenta con espacio suficiente para ampliar las funcionalidades desarrolladas o agregar nuevas.

Durante las pruebas realizadas se encontró que al utilizar el dispositivo como se lo presentó en la Fig. 1 se registraban fluctuaciones en las mediciones. Esto se resolvió confinando el sistema en un recipiente, con lo que se logró disminuir el impacto de las perturbaciones externas en las mediciones.

La primera iteración de diseño utilizaba solo el *SoC Esp32*, pero el control de potencia por ángulo de disparo no tenía la estabilidad deseada. Esto se debió a la latencia del sistema al momento de atender las interrupciones. Se intentó utilizar otros periféricos disponibles en el *SoC* (*MCPWM* y *LEDC*), pero siempre se requirió sincronizar alguna señal con la detección de cruce por cero, por lo que no se obtuvieron resultados satisfactorios y se incluyó un segundo microcontrolador.

Todo el material asociado al trabajo aquí presentado se encuentra disponible en su repositorio [7].

REFERENCIAS

- [1] P. M. Oliveira and J. D. Hedengren, "An apmonitor temperature lab pid control experiment for undergraduate students," en *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2019, pp. 790–797.
- [2] G. Takács, M. Gulan, J. Bavlina, R. Köplinger, M. Kováč, E. Mikuláš, S. Zarghoon, and R. Salíni, "Heatshield: a low-cost didactic device for control education simulating 3d printer heater blocks," en *2019 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2019, pp. 374–383.
- [3] M. L. Miretti y E. Bernardi, "Infraestructura para el desarrollo de laboratorios remotos," *Elektron*, vol. 6, no. 1, 2022.
- [4] J. W. Eaton, D. Bateman, S. Hauberg *et al.*, *Gnu octave*. Network theory London, 1997.
- [5] E. S. Burgos y E. J. Adam, "Graphical user interface editor for octave applications," *Engineering Reports*, vol. 2, no. 11, p. e12269, 2020.
- [6] M. Miretti. (2020) Octave websockets. [Online]. Disponible en: <https://gnu-octave.github.io/packages/websockets>
- [7] E. S. Burgos. (2022) Repositorio del proyecto "tempcontrol". [Online]. Disponible en: <https://gitlab.com/labinformatica/tempcontrol>

Categoría

FORO TECNOLÓGICO

Inteligencia Artificial

Área Temática

People Behavior Tracking (PBT)

Hernán Contigiani^{†1}, Pasquinnell Urbani^{*2}

[†]Facultad de Ingeniería - Universidad de Buenos Aires (FI-UBA)
Universidad Tecnológica Nacional (UTN)

¹hernan4790@gmail.com

*Universidad de Chile (UCh)

¹purbanir@gmail.com

Resumen—El presente documento describe el diseño de un sistema de monitoreo de personas. El sistema permite monitorear mediante técnicas de inteligencia artificial basadas en visión por computadora los desplazamientos que realiza una persona en un espacio y obtener métricas de las zonas que visitó. Se presentan los resultados y mejoras de la arquitectura propuesta respecto a las soluciones que se pueden hallar en la bibliografía, y cómo esta propuesta de trabajo se transformó en una solución real llevada a cabo en conjunto con la empresa Globant S.A.

Palabras claves—seguimiento de personas, aprendizaje profundo, re-identificación de objetos

I. INTRODUCCIÓN

El propósito de este trabajo fue el desarrollo de un sistema de monitoreo de personas. Tiene como principal objetivo estudiar los movimientos que realiza un individuo al ingresar y transitar un espacio a fin de obtener métricas sobre las zonas que visitó. Para cumplir este objetivo es necesario poder detectar a las personas que aparecen en el video tomado por una cámara, realizar el seguimiento de cada una e identificarlas aún cuando desaparecen del espacio de visión por unos minutos. La solución debe ser capaz de realizar el seguimiento en espacios muy transitados, como puede ser una tienda, un negocio o un supermercado.

Existen distintas soluciones abiertas basada en inteligencia artificial que resuelven parte de este problema, tanto en librerías de visión por computadora como OpenCV [1], proveedores de *hardware* como NVIDIA [2] o Intel [3] y en proveedores de soluciones *cloud* como Amazon [4] o Google [5]. Los algoritmos y modelos de estos proveedores que pueden utilizarse para el seguimiento de personas carecen de robustez por sí solos, dado que solo funcionan en espacios abiertos libres de obstáculos o en donde la densidad de personas en el espacio es baja. Mantener a cada persona correctamente identificada sin perderla o confundirla (mantener el identificador único de seguimiento, su ID) es muy complejo, debido que las personas en movimiento forman y disuelven grupos, cambian su pose constantemente, pueden salir y volver de escena, etc. Este trabajo aborda esta problemática.

II. SISTEMA DE SEGUIMIENTO DE PERSONAS

A. Tecnologías empleadas

Para desarrollar el sistema de seguimiento se emplearon técnicas de visión por computadora del campo del aprendizaje profundo (*Deep Learning*) [6] entre las cuales se encuentran:

- Detección de objetos: técnica que permite detectar múltiples objetos [7] (personas, vehículos, animales, etc) en una imagen indicando la ubicación de cada objeto y el tipo de objeto detectado. Las detecciones son representadas con una caja de detección (*bboxe*) [8], la cual posee las coordenadas en píxeles para ubicar al objeto en la imagen. En este trabajo se utilizó *Yolo (You Only Look once)* [9] como modelo detector de objetos configurado para detectar personas en imágenes y conocer su ubicación.
- Seguimiento de objetos: técnica que permite seguir múltiples objetos en imágenes [10] asignando a cada objeto un identificador único (ID) de seguimiento. Este identificador es un valor numérico incremental que permite distinguir de forma unívoca a cada objeto encontrado en la imagen. Se espera que se mantenga el mismo número para el mismo objeto mientras este permanezca a la vista. En este trabajo se utilizó *Deepsort (Deep Simple Online Tracking)* [11] el cual consume las detecciones (*bboxes*) del detector y asigna a cada objeto un ID único.
- Extraer características visuales: técnica que permite extraer vectores de características de una imagen de una persona que la representan y diferencian de otra. El extractor toma las detecciones que arroja el detector y genera un vector numérico por cada persona que representa las características observadas en ella. Las características que arroja el extractor son números, no representan ninguna cualidad física específica sino más bien un espacio numérico que la representa y diferencia de otras. En este trabajo se utilizó *Osnet (Omni-Scale Network)* [12] como modelo extractor de características.

B. Desafíos en el seguimiento de personas

Los desafíos que se detallan a continuación son problemas que presentan en este tipo de sistemas de monitoreo a la hora de seguir un objeto con el mismo identificador:

- Pérdida del identificador de seguimiento (ID): ocurre cuando una persona pasa por detrás de otro objeto que la ocluye. El objeto que la ocluye puede ser estático (un mueble, un cartel, etc) o dinámico (otra persona o grupo de personas en la escena).
- Intercambio de identificador de seguimiento (ID): ocurre cuando dos personas se cruzan en movimiento. Al momento que ambas personas se cruzan, sus *bboxes* poseen

una ubicación geométrica y un vector de movimiento similar. Estos factores producen que el modelo seguidor no pueda diferenciar las detecciones, aumentando significativamente la probabilidad de que el modelo intercambie sus identificadores por error.

Estos desafíos son el principal problema de los sistemas que se mencionaron en la introducción, que se basan únicamente en modelos de detección y seguimiento de objetos. Estos sistemas requieren un ajuste fino de los parámetros de configuración de la cámara y del área de visión a fin de reducir los errores al mínimo [13]. En este trabajo se busca superar estos desafíos agregando a la ecuación un modelo de extracción de características y técnicas de re-identificación por aprendizaje automático.

C. Métricas de evaluación del sistema

Los sistemas de monitoreo o seguimientos de objetos se los evaluará según cuantos objetos pudo el sistema detectar y seguir correctamente. Para este trabajo se definieron los siguientes requerimientos de seguimiento:

- Se considerará que una persona es correctamente monitoreada si al menos se mantuvo su seguimiento el 80% del tiempo que circuló en el recinto.
- Se considerará que el sistema funciona dentro de los parámetros aceptables si entre el 80% y 100% de las personas en el video fueron correctamente monitoreadas.

III. DISEÑO E IMPLEMENTACIÓN

A. Arquitectura

En la figura 1 se observa un diagrama general de la solución y las etapas que la componen:



Fig. 1. Diagrama general de la solución.

- Cadena de procesamiento: etapa compuesta por los tres modelos de inteligencia artificial (detector, seguidor, extractor). Cada modelo en la cadena realiza un aporte al sistema de detección y seguimiento de personas. El resultado de esta etapa son las detecciones y características de cada persona observada en el video suministrado al sistema.
- Motor de seguimiento (Engine): etapa encargada de mejorar la precisión de seguimiento mediante re-identificación y manejo de oclusiones. Recibe los datos de la etapa de cadena de procesamiento y calcula las métricas de monitoreo que luego se presentan en la interfaz de usuario disponible en la aplicación web.
- Interfaz de usuario: aplicación web que utiliza el usuario para configurar las zonas de interés en el recinto y

observar la evolución de las métricas y el monitoreo de personas.

B. Re-identificación

El Engine utiliza los vectores de características de cada objeto para agrupar los datos (*clustering*) que corresponden a una misma persona. Clustering es el proceso que permite establecer relaciones entre elementos que poseen características en común conformando grupos de datos (*clusters*). En la figura 2 se observa cómo las distintas capturas tomadas de una persona en diferentes ángulos se agrupan para formar un cluster.



Fig. 2. Segmentación de personas en clusters.

El proceso de re-identificación de personas busca relacionar una nueva persona por su vector de características con un cluster ya conformado. El objetivo es mantener el seguimiento de una persona con el mismo cluster, es decir con el mismo identificador, a pesar de que se haya salido de escena por unos segundos. Para poder calcular la similitud entre el vector de una persona y los diferentes clusters se utiliza la similitud coseno [14], que devuelve un valor entre -1 y 1 siendo 1 la expresión para máxima similitud.

Cuando el modelo seguidor pierde el identificador de seguimiento de una persona o se intercambia por uno erróneo, el Engine ejecuta el proceso re-identificación para asignar a esa persona su cluster correspondiente. El proceso de re-identificación está conformado por una serie de *thresholds* y *timers* a fin de evitar generar falsos positivos. Estos valores de configuración se obtuvieron de forma empírica utilizando videos de diferentes recintos y diferentes personas, analizando la calidad de los clusters en cada ensayo mediante una matriz de similitud [15].

IV. RESULTADOS

A. Ensayos

Dado que para ensayar todas las condiciones se necesita involucrar varias personas a lo largo del video, se optó por utilizar el videojuego “Los Sims 3” [16], en el cual es posible crear un ambiente totalmente a medida y personajes que interactúan en él. El juego permite crear personajes con diferentes personalidades por lo que cada uno interactúa de forma singular con el espacio pudiéndose obtener distintas métricas por el tiempo que se desee que dure el ensayo. En la figura 3 se observa el entorno creado para ensayar el sistema.

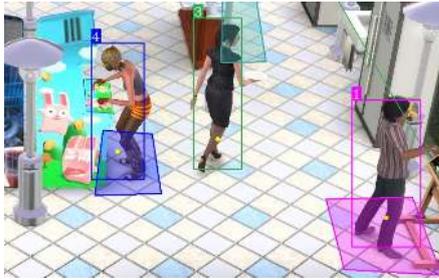


Fig. 3. Escenario creado en el video juego "Los Sims 3".

Se generó un video de cinco personas que interactuaron con el espacio durante diez minutos, el cual permitió ensayar y validar todas las funcionalidades del sistema. Se ensayaron tres sistemas diferentes a fin de comparar las técnicas incluidas en cada uno:

- Yolo y Deepsort: sistema basado unicamente en el detector Yolo y el seguidor Deepsort, tal cual se encuentra disponible en la web.
- Engine10: sistema basado en el detector Yolo, el seguidor Deepsort y el postprocesado del Engine v1.0, este último utiliza los vectores de características para el manejo de oclusiones y re-identificación de personas.
- Engine16: sistema basado en el detector Yolo, el seguidor Deepsort y el postprocesado del Engine v1.6, este último incorpora la información agregada de las las zonas de interés (información del recinto) al manejo de oclusiones y re-identificación de personas.

La precisión del sistema de monitoreo se obtiene como la cantidad de personas correctamente monitoreadas sobre el total de personas en el video. Se considera que una persona es correctamente monitoreada si mantiene su identificador de seguimiento durante al menos el 80% del tiempo que la persona permanece en el espacio. En la tabla 1 se observa la precisión alcanzada por los sistemas sometidos al mismo ensayo.

TABLA I
PRECISIÓN DE SEGUIMIENTO POR SISTEMA

Métricas	Sistemas		
	<i>yolo deepsort</i>	<i>Engine10</i>	<i>Engine16</i>
precisión	12.5%	75%	87.5%

En conclusión, la precisión del sistema aumenta con cada nueva funcionalidad que se incorpora. A medida que evolucionan los sistemas también se reducen los intercambios de identificador entre personas y aumenta la precisión.

B. Implementación

Una vez validado el sistema y el potencial del mismo, se continuó trabajando en su implementación como un sistema de procesamiento de datos en tiempo real. Para ello, se incorporó el uso de las herramientas de desarrollo de NVIDIA para sistemas embebidos [17], que permiten optimizar los modelos de inteligencia artificial para el *hardware* del dispositivo candidato.

Se utilizó la familia de dispositivos NVIDIA Jetson [18], que poseen una placa de video integrada (GPU) optimizada para inteligencia artificial. Son computadoras de alto rendimiento que permiten ejecutar procesamiento en el borde (*edge computing*).

Un factor importante a tener en cuenta en la implementación de un sistema de visión por computadora en tiempo real es su velocidad de ejecución, la cual se mide en *frames per seconds* (FPS) [19]. El sistema sin optimización (Engine v1.6) no cumplía el requerimiento mínimo de 10 FPS establecido para esta solución, por lo que se optimizó el sistema con técnicas de cuantización y optimización por hardware [20]. El sistema definitivo (Engine v2.0) desplegado en una Jetson Xavier NX alcanzó una velocidad de ejecución de 20 FPS.

V. CONCLUSIÓN

Se alcanzó a cumplir el objetivo propuesto, el cual consistía en estudiar los movimientos que realiza una persona al ingresar y transitar un espacio durante al menos el 80% del tiempo que la persona permanece en el espacio. La precisión del sistema superó la precisión que puede alcanzarse utilizando los modelos disponibles de detección y seguimiento en la web.

La experiencia adquirida en la realización de este trabajo permitió que el proyecto no quedará enmarcado unicamente como proyecto de investigación, sino que también se continuó trabajando en conjunto con Globant S.A. [21] en soluciones con clientes reales.

REFERENCIAS

- [1] OpenCV, "Tracking by detection". <https://rebrand.ly/z7aniy2> (May. 2022).
- [2] NVIDIA, "Object detection and Tracking". <https://rebrand.ly/r8sj94o> (May. 2022).
- [3] Intel, "Retail Analytics". <https://rebrand.ly/cdspxp3> (May. 2022).
- [4] Amazon, "Object Detection". <https://rebrand.ly/5k5lqrh> (May. 2022).
- [5] GCP, "Track objects". <https://rebrand.ly/ycc5tb5> (May. 2022).
- [6] J. Brownlee. "What is Deep Learning?". <https://rebrand.ly/2u5to9w> (Jul 2019).
- [7] Tensorflow, "Object Detection". <https://rebrand.ly/1hg81pu> (May 2021).
- [8] OpenCV, "Bounding box". <https://rebrand.ly/n61zkzw> (May 2021).
- [9] J. Redmon, A. Farhadi, "YOLOv3: AN Incremental Improvement". <https://rebrand.ly/060cdb> (Ago. 2018).
- [10] VISO AI, "Object Tracking". <https://viso.ai/deep-learning/object-tracking/> (Dic. 2020).
- [11] D. Paulus, N. Wojke, A. Bewley, "DeepSORT". <https://rebrand.ly/37fb38> (Mar. 2020).
- [12] K. Zhou, Y. Yang, A. Cavallaro and T. Xiang, "Omni-Scale Feature Learning for Person Re-Identification," 2019 IEEE/CVF ICCV, 2019, pp. 3701-3711, doi: 10.1109/ICCV.2019.00380.
- [13] Y. Wu, J. Tsotsos, "Active Control of Camera Parameters for Object Detection Algorithms". <https://rebrand.ly/gxl9na9> (May. 2021).
- [14] Scikit learn, "pairwise cosine similarity". <https://rebrand.ly/wme8zh6> (May. 2021).
- [15] Universidad de Valencia, "Análisis Cluster". <https://rebrand.ly/2gln1pg> (May. 2022).
- [16] Electronic Arts, "Los Sims 3". [DVD-ROM]. 2019.
- [17] NVIDIA, "NVIDIA SDK Manager". <https://rebrand.ly/0uodqin> (May. 2022).
- [18] NVIDIA, "Sistema Integrado con Jetson". <https://rebrand.ly/nvem13n> (May. 2022).
- [19] A. Fagg, "Frame Rate Matters for Embedded". <https://rebrand.ly/1bdf6> (May. 2021).
- [20] NVIDIA, "Using Quantization with NVIDIA". <https://rebrand.ly/0bcf1c> (May. 2021).
- [21] Globant S.A, "Somos Globant". <https://www.globant.com/es/about> (May. 2022).

B-VGG16: Red Neuronal de Convolución binarizada para el reconocimiento de Objetos

1^{er} Nicolás Urbano Pintos
 Grupo TAMA - UTN FR Haedo
 DRL - CITEDEF
 Villa Martelli, Argentina
 urbano.nicolas@gmail.com

2^{do} Héctor Alberto Lacomí
 Grupo ASE - UTN FR Haedo
 DRL - CITEDEF
 Villa Martelli, Argentina
 hlacomí@citedef.gob.ar

3^{er} Mario Blas Lavorato
 Grupo TAMA
 UTN FRH
 Haedo, Argentina
 blamar2004@yahoo.com.ar

Resumen—En este trabajo se propone implementar una red neuronal binarizada (BNN) de convolución para reconocer objetos a partir de imágenes RGB. Las BNN reducen la cantidad de recursos computacionales y de memoria, y permiten inferirlas en sistemas embebidos como las FPGA logrando respuestas en tiempo real. El modelo se basa en la red VGG16 y se entrena con el conjunto de datos CIFAR10. La red se cuantiza de forma binaria con la técnica de cuantización consciente del entrenamiento (QAT). Se logró una precisión cercana al 88% con el conjunto de evacuación de CIFAR10.

Palabras Clave—Aprendizaje profundo, Redes Neuronales de convolución, Cuantización binaria

I. INTRODUCCIÓN

Las redes neuronales de convolución (CNN – Convolutional Neural Network) son algoritmos de aprendizaje profundo [1], los cuales tienen como entrada una imagen, y se les asignan pesos y bias a los aspectos u objetos de la imagen, de modo de poder diferenciarlos entre sí.

Este tipo de redes utilizan pesos y activaciones de punto flotante, generalmente de 16 o 32 bits, por lo tanto, dependen de dispositivos dedicados como los GPU's (Graphics Processing Unit), ya sea para realizar el entrenamiento o para inferir. En aplicaciones como la navegación autónoma, la robótica y la realidad aumentada, es necesario implementar inferencias en dispositivos capaces de tener un alto rendimiento, con niveles de energía menores, y a un costo menor al de las GPU. Es por ello, que diversos investigadores se han enfocado en los últimos años en la implementación de inferencias de redes neuronales en sistemas embebidos [2]. Cómo es el caso de las FPGA (Field Programmable Gate Array), que son dispositivos que tienen un gran rendimiento en operaciones de bits, por lo tanto, permiten realizar inferencias en tiempo real a un bajo costo energético. Para utilizarlas, es necesario cuantizar los pesos y las activaciones a bits. Este trabajo realiza las siguientes contribuciones:

- Implementa una red VGG16 [3] cuantizada de forma binaria.

- Obtiene con el modelo cuantizado una precisión comparable a la del modelo completo entrenando y evaluando con el dataset CIFAR10 [4].
- Utiliza activaciones bipolares, para que el modelo pueda ser inferido en FPGA con compuertas XNOR.

II. ANTECEDENTES

En la última década, se ha puesto el foco en el aprendizaje profundo para el reconocimiento de objetos a partir de imágenes, en este campo se han logrado resultados favorables [5], pero estas redes utilizan como datos variables de puntos flotantes (FP- Floating Points) de 16 o 32 bits. Lo cual hace que en redes profundas, por la cantidad de parámetros implicados, la cantidad de memoria necesaria para inferir el modelo sea alta.

Una red neuronal es una estructura redundante, y si se determinan las neuronas que tienen poca injerencia en la salida, se puede optimizar la red. Por tal motivo, se han desarrollado diferentes técnicas de cuantización para poder inferir estas redes en dispositivos como las FPGA [2], las cuales tienen un mejor rendimiento que las GPU y las CPU en las operaciones de bits.

La cuantización consciente del entrenamiento QAT (Quantization Aware Training) compensa los errores mientras se realiza el entrenamiento, pero esto trae aparejados tiempos de entrenamiento más largos. Durante el entrenamiento todos los cálculos son en FP, con módulos que modelan el efecto de la cuantización aplicando clamping y redondeando.

Dentro del estudio de las NN cuantizadas, se encuentran las redes binarizadas, en donde todas las operaciones de la red se resuelven con el cómputo de salidas que están restringidas únicamente a bits, como el caso de BinaryNet [6]. Si bien, las activaciones y los pesos son binarios, los gradientes de valor real son acumulados en variables de valor real de alta resolución. Estos valores son utilizados por el gradiente descendiente estocástico, que explora el espacio de parámetros en pasos pequeños y ruidosos, este ruido se promedia por las contribuciones de los gradientes estocásticos acumulados en cada peso. El hecho de sumar ruido al cálculo

de los gradientes, no es más ni menos que una forma de regularización, por lo tanto, contribuye a una mejora de la generalización. Esto es similar a lo que ocurre con Dropout [7], aunque en lugar de colocar en 0 la mitad de las activaciones de forma aleatoria, se binarizan los pesos y las activaciones. En BinaryNet se binariza con la ecuación 1.

Al derivar la función sign, se observa que en la mayoría de los casos es cero, por lo tanto, no es compatible con la retro propagación. Para solucionar este inconveniente, Courbariaux et al. [6] utilizan la estimación directa STE, pero tomando en cuenta el efecto de la saturación. La estimación STE establece que los gradientes entrantes a la función sean igual a las salientes, sin tener en cuenta la derivada de la función.

En la actualidad, el equipo de Xilinx ha desarrollado el repositorio BNN-PYNQ [8], que es parte del proyecto FINN, está basado en la red BinaryNet. Los autores utilizan un modelo inspirado en VGG11, al que denominan CNV, y obtienen una precisión en el dataset CIFAR10 del 84.22% con una cuantización de 1 bit de pesos y 1 bit de activación.

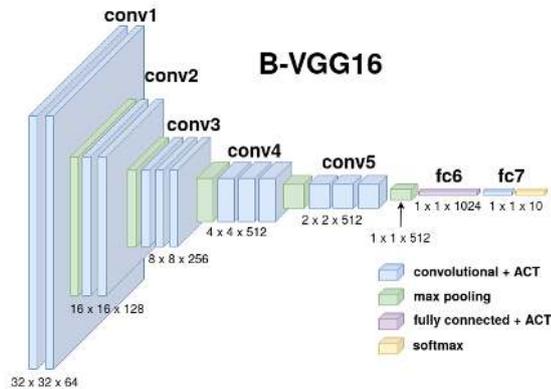


Fig. 1. Arquitectura del modelo B-VGG16 propuesto

III. EXPERIMENTOS

Este trabajo se basa en la variación de la red VGG16 propuesta por Lui et. al [9], la cual consiste de pilas de capas formadas por convolución, activación, y pooling. En la fig. 1 se observa la combinación de capas utilizadas y la cantidad de filtro asociada a cada una. Incrementando la cantidad de filtros en cada pila de capas, siendo estos 64, 128, 256 y 512.

Por último, se utiliza una red neuronal totalmente conectada para relacionar los 512 elementos con las 10 clases de salida de la red.

Se utiliza una cuantización tipo QAT, se implementa en Pytorch [10] a partir de la librería Brevitas [11] con pesos y activaciones de 1 bit, se representa a partir de -1 y +1, utilizando la función de la ecuación 1, la normalización se realiza con BatchNorm.

$$\text{Sign}(x) = \begin{cases} +1 & \text{if } x \geq 0, \\ -1 & \text{if } x < 0. \end{cases} \quad (1)$$

Para el entrenamiento se utilizó un Learning rate de 0.02, un tamaño de batch de 100, como función de pérdidas Cross Entropy, como optimizador ADAM, y se cuantizó con 1 bit tanto para los pesos como para las activaciones.

Se entrenaron 1000 épocas en casi 5 horas. Vale la pena aclarar, que si bien se obtienen pesos y activaciones binarios, para realizar el entrenamiento se utilizan valores reales que luego son cuantizados en cada capa, por ese motivo se entrenó en una GPU de NVIDIA compatible con CUDA [12].

IV. RESULTADOS

Con el modelo planteado, denominado VGG16 Binarizado o B-VGG16, se obtiene una precisión con el dataset de evaluación CIFAR10 de un 87.97%. La misma es calculada como el cociente de las predicciones correctas con el número total de predicciones. Se denomina a esta métrica TOP1. Además, se evaluó con la métrica TOP5, la cual, toma la predicción como correcta si la clase verdadera se encuentra dentro de las primeras 5 clases predichas. Se obtuvo una precisión TOP1 del 87.97%, y una precisión TOP5 del 98.92%. En la tabla I se observa la cantidad de memoria requerida para el modelo. La misma es calculada con el tamaño de activación y los parámetros de cada capa.

En la figura 2 se observa la evolución de la precisión de validación durante el entrenamiento de acuerdo a cada época.

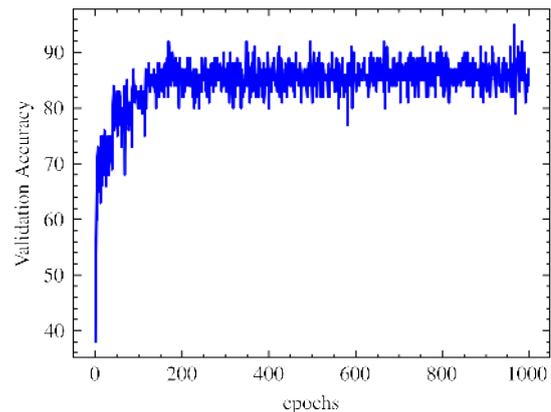


Fig. 2. Precisión de Validación vs. Épocas

Además, de la evaluación con imágenes pertenecientes al dataset se evaluó con fotografías, como es el caso de la 3, donde en primer lugar se recortó la imagen para centrar al vehículo, y luego se la redujo a 32x32 píxeles. Se pueden observar las predicciones obtenidas en 4.

Se compara el modelo propuesto cuantizado con la red CNV y la red VGG16 con BatchNormalization y Dropout, se observa en la tabla I las precisiones de evaluación TOP1 obtenidas.

V. CONCLUSIONES Y TRABAJO A FUTURO

En este trabajo se implementa una red de convolución basada en la arquitectura VGG16 a partir de las herramientas

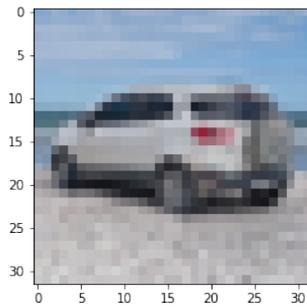


Fig. 3. Imagen reducida a 32x32 píxeles

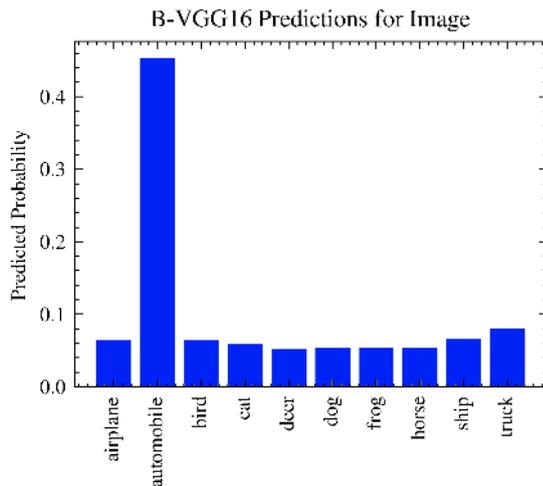


Fig. 4. Probabilidad de la predicción

de cuantización tipo QAT que ofrece la librería Brevitas. Se utiliza una cuantización binaria de 1 bit tanto para los pesos como para las activaciones. Se entrena y evalúa con el dataset CIFAR10. El modelo propuesto obtiene una mayor precisión de evaluación respecto al modelo CNV [8], esto se debe a que se aumentan la cantidad de pilas de capas de convolución, activación y pooling. En la tabla I se observa una mejora de casi 4 puntos porcentuales. Al utilizar un modelo más profundo, con mayor cantidad de activaciones y parámetros, también aumenta la cantidad de memoria en juego. Esto podría implicar un problema a la hora de implementar la inferencia de la red en sistemas tipo FPGA, pero el modelo CNV está implementado para un chip de Xilinx ZYNQ 7020, que se encuentra dentro de la generación anterior de FPGA.

TABLA I
COMPARACIÓN CON OTROS MODELOS

Red	Variables	TOPI	Memoria
VGG16 [9]	FP32	~ 90%	466 MB
CNV [8]	1 bit (Quant 1W1A)	84%	1,87 MB
B-VGG16 (propio)	1 bit (Quant 1W1A)	87,97%	14,57 MB

Actualmente, la nueva generación de Zynq Ultrascale+ es capaz de albergar dichos modelos.

Al cuantizar los pesos y las activaciones se pierde precisión, pero gracias a la cuantización tipo QAT, a medida que el modelo se entrena, estas pérdidas se compensan. Se puede ver que el modelo cayó poco más de 2 puntos en porcentaje en comparación con el modelo VGG16-BN-DROPOUT entrenada con FP32.

Por lo tanto, se puede concluir que con el modelo B-VGG16 propuesto se obtienen precisiones comparables con las del modelo VGG16 tradicional. Esto implica que es posible lograr resultados similares utilizando menos memoria, ya que, en comparación con las redes de punto flotante de 32 bits, requiere un tamaño en memoria 32 veces menor como se observa en I. También se reducen la cantidad de accesos en memoria, 32 tiempos menos de acceso a memoria.

El modelo planteado se puede implementar con una cantidad de hardware menor, ya que al estar los pesos y las activaciones binarizadas de forma bipolar, con +1 y -1, puede ser resuelto con compuertas XNOR en dispositivos tipo FPGA. A partir del framework experimental FINN, se implementará esta red en una FPGA de Xilinx Ultrascale+.

REFERENCIAS

- [1] N. Aloysius and M. Geetha, "A review on deep convolutional neural networks," *Proceedings of the 2017 IEEE International Conference on Communication and Signal Processing, ICCSP 2017*, vol. 2018-Janua, pp. 588–592, 2018.
- [2] A. Shawahna, S. M. Sait, and A. El-Maleh, "FPGA-Based accelerators of deep learning networks for learning and classification: A review," *IEEE Access*, vol. 7, pp. 7823–7859, 2019.
- [3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–14, 2015.
- [4] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.
- [5] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object Detection with Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212–3232, 2019.
- [6] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1," *arXiv*, 2016. [Online]. Available: <http://arxiv.org/abs/1602.02830>
- [7] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.
- [8] Y. Umuroglu, N. J. Fraser, G. Gambardella, M. Blott, P. Leong, M. Jahre, and K. Vissers, "FINN: A framework for fast, scalable binarized neural network inference," *FPGA 2017 - Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, no. February, pp. 65–74, 2017.
- [9] S. Liu and W. Deng, "Very deep convolutional neural network based image classification using small training sample size," *Proceedings - 3rd IAPR Asian Conference on Pattern Recognition, ACPR 2015*, pp. 730–734, 2016.
- [10] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimeshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, no. NeurIPS, 2019.
- [11] A. Pappalardo, "Xilinx/brevitas." 2021. [Online]. Available: <https://doi.org/10.5281/zenodo.3333552>
- [12] NVIDIA, P. Vingelmann, and F. H. Fitzek, "Cuda, release: 10.2.89," 2020. [Online]. Available: <https://developer.nvidia.com/cuda-toolkit>

Genre Classification with Deep Learning Techniques

Marco Carnaghi and María C. Cebedio
 ICYTE - UNMDP
 Mar del Plata, 7600, Argentina
 {mcarnaghi, celestecebedio}@fi.mdp.edu.ar

Abstract—Music recommendation systems are aimed at improve the listening and search experience of music consumers. The increased access to digital content has turned the algorithmic recommendation systems into a necessity in order to save the time of the users. A common approach is to employ metadata obtained from the audio track to generate the different recommendations Music genre is one of the most important descriptors in the decision process. Therefore, in this paper, the performance of three neural networks architectures for automatic genre classification is compared. The dataset, preprocessing of audio data, training process and concepts employed for models combination is also presented. The results indicate that to combine the local pattern representation ability of CNN models with the time relationship overview of RNN models leads to an improvement in the performance of the system. The proposed models and their analysis focus on improving classification accuracy while maintaining a reduced number of layers to allow an easy implementation, whether in a embedded systems or as flexible module within another application.

Index Terms—Convolutional Recurrent Neural Networks, Music Recommendation Systems, Content-based, Genre Classification

I. INTRODUCTION

Music streaming services are now more accessible than ever before, and, consequently, so is music in all its variety. However, nowadays, the music industry offers just too much options for the user to explore, turning the song selection into a time-consuming process which causes in the user information fatigue. To this fact is added the tendency of users to accept recommendations or to prefer a guided experience in their music choice and discovery of new artist [1]. Therefore, music recommendation systems (MRS) have recently became into a extensive field of study in order to improve the user experience [2], [3].

MRS can mainly be classified into three approaches:

- Collaborative Filtering: This approach bases on user information and propose that users might like what similar users listen to.
- Context-aware: This approach considers user's situation, activity, and circumstances that constitute the context in which the music is listened to. The context elements can be environment-related, i.e, links between the user's geographical location and music listened there in the past; or user-related, i.e, aspects associated with the activity the user is carrying out at the moment, the mood and similar circumstances.
- Content-based: This approach bases on the metadata that describes a track and the assumption that a user is likely to like tracks with similar metadata to tracks he already likes.

Regarding the latters, efforts are been made in the area of Music Information Retrieval (MIR) to develop automated algorithms for descriptive metadata (e.g. mood, energy, and genre) extraction. In this sense, the proposal is to develop a genre classification system based on a hybrid architecture constituted by convolutional layers and recurrent layers for the feature extraction stage and feed-forward layers for the genre classifier. This system can be considered a module within a complete content-based or context-aware music recommendation application or any music recommendation system that take into consideration sound sources as input references. The model takes as inputs the Mel-Frequency Cepstrum Coefficients (MFCCs) obtained from music frames which are reshaped into a grey-scale image. Then, the model output is the predicted genre for the track.

II. MODEL DEVELOPMENT

In this section, the followed steps to develop the Neural Network based model are described. The presented process includes a description of the data collection stage, the preprocessing of the audio data and description of the neural network architectures. All the step that are explained below have been implemented in a Python notebook [4].

A. Data Collection

The collection of meaningful data has a remarkable impact on the final prediction capacity of the model. As neural networks models learn from the initial data, it is a crucial aspect to train the model with a high quality and equally distributed dataset. Additionally, as a supervised training algorithm is employed, the dataset needs to be also well labeled [5]. The GTZAN Music Genre Dataset is used to fulfil the mentioned requirements. This set is constituted of 1000 audio tracks each 30 seconds long. There are 10 genres represented, each containing 100 tracks. All the tracks are 22050 Hz monophonic 16-bit audio files in .wav format [6].

B. Preprocessing of the Audio Data

Before being fed into the neural network model, the audio data needs to be transform into a more suitable representation. With this aim in mind, MFCCs are obtained from the audio signal data and, after that, converted into grey-scale images. To do this conversion, each audio track is divided into frames, according to the window length and hop length, for each of which is calculated the MFCCs. Then, the vectors of coefficients for each frame are concatenated to create a spectrogram.

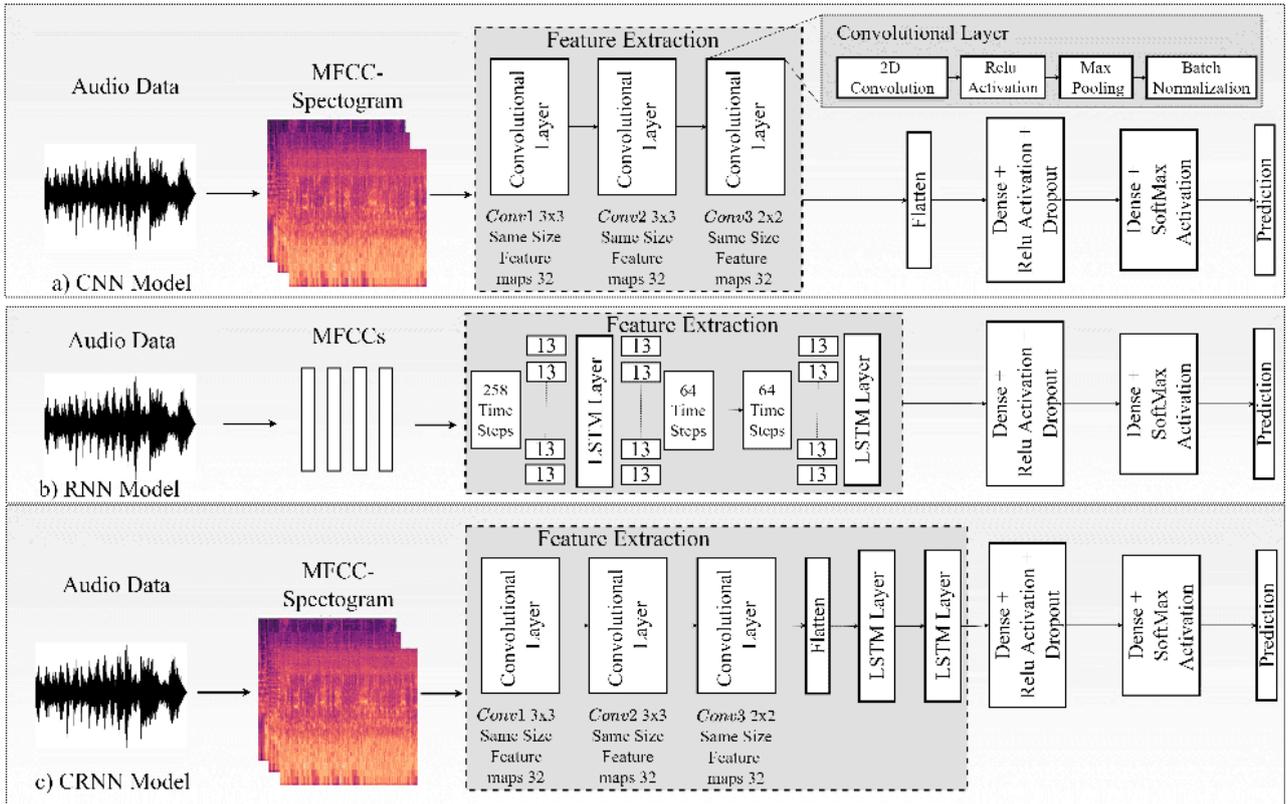


Fig. 1: Neural Network architectures and data processing chain: a) CNN model, b) RNN model and c) CRNN.

C. Neural Network model

1) *Architectures*: To classify the music genres, the three architectures shown in Fig. 1 were used. For all the alternatives, the last stage is in charge of the classification task and consist on a full-connected feed-forward 2-layers network. An a priori analysis based on a literature survey, indicates that:

- CNN take 3D tensors as input data and the convolutional layers are stacked to get representations of local patterns. Then, a hierarchical proceeding occurs by adding more layers to combine local patterns and detect more complex structures by looking over wider contexts. Convolutional layers are also employed in order to preserve spatiality in time and frequency. This architectures incorporate non-linear subsampling layers to provide translational invariance to the model and reduce the size of intermediate representations [7].
- RNN take time-series data and LSTM or GRU layers are stacked to model time relations in the data. Recursive layers can be set in sequence configuration or single output configuration. Sequence configuration generates an output value for each cell, which, generally, refers to the prediction of the following value. Whereas, a single output configuration is employed in classification applications [7], [8].
- CRNN seek to take the best of both worlds. First, convolutional layers are stacked to find local patterns and combine them into more complex structures while preserving spatiality. Then, recursive layers are added to summarise information over time [9].

According to above mentioned, MFCCs coefficients need

Table I: Neural Network training parameters

<i>Audio_samplerate</i>	22050 Hz
<i>Track_Duration</i>	30 s
<i>Hop_length</i>	512
<i>N_samples_FFT</i>	2048
<i>N_MFCCs</i>	13
<i>Input_spectrogram_shape</i>	258 x 13
<i>Batch_size</i>	32
<i>Epochs</i>	20
<i>Optimizer</i>	ADAM
<i>Loss_function</i>	Sparse Categorical Cross-Entropy

to be reshaped into grey-scale images or 3D tensors before being fed into the CNN and CRNN models, but need to be kept as time-series data to be compatible with the RNN model. Lastly, note that Batch Normalization and Dropout layers are introduced into the model to speed the training process and, together with the early stop method, avoid overfitting [10].

2) *Training*: In order to train the models, the MFCC-spectrograms generated from the audio data are split into training, validation and testing datasets, which proportion is 3:1:1 respectively. Regarding the labelling codification, to adapt the representation to the softmax output a one hot encoding is used. This means, that a value of 1 is given to the chosen music genre and a value of 0 is given to the other genres.

III. RESULTS

In the current study, the parameters listed in Table. I were employed for the training process of the three architectures. Fig. 2 shows the evolution of training and validation error and accuracy of the three models. As it can be appreciated, the CNN model presents a smoother evolution in comparison with the other two models. This

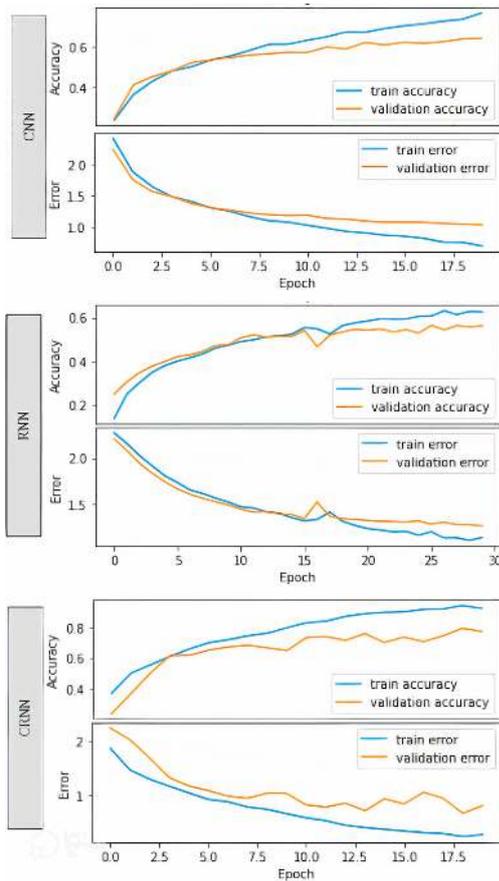


Fig. 2: Train and validation error and accuracy for the three compared models.

correspond to an expected behavior as RNNs are known to be harder to train than CNNs. Additionally, all models present a reduction in the learning slope near the 30th epoch, that is why, it was considered an appropriate condition to stop the training process in order to avoid overfitting. Finally, the inclusion of LSTM layers in the CRNN model has led to a more erratic learning curve but it also entailed an improvement in the accuracy.

The Area under a Receiver operating characteristic (ROC) of the three proposed models is compared in the Table. II where, accordingly with the observations made about the training process, the performance of the CRNN model overcomes the one of the other two models by at least 16%. In the same Table, the performance of the proposed model is compared with other proposals in the literature. On the one hand, the proposed CRNN has better AUC score than the CNN model presented in [11]. On the other hand, the performances is worst than the achieved by the models presented in [12], [13]. However, note that the proposed architecture is simpler and, last but not least, the available dataset (GTZAN) has 25 and 214 times less data samples than the MagnaTagATune dataset and Million Song dataset, respectively. The latter means that those models were trained with richer datasets. Therefore, a straightforward performance comparison among them is not entirely correct.

IV. CONCLUSIONS

In this paper, the performance of three neural network architectures was compared for their application to genre

Table II: Model's AUC^a

Model	AUC ^a	Architecture summary	Dataset
Proposed CNN	0.65	96 feature maps	GTZAN
Proposed RNN	0.60	128 LSTM cells	GTZAN
Proposed CRNN	0.81	96 feature maps + 128 LSTM	GTZAN
FCN-4 [12]	0.894	3968 feature maps	MagnaTagATune ^b
Bag of features and RBM [13]	0.888	Manual feature extraction and Fully connected layers	MagnaTagATune ^b
CNN [11]	0.77	400 feature maps	Million Song Dataset ^c

^a Area Under Receiver Operating Characteristic.

^b MagnaTagATune dataset consists of 25,856 audio tracks of 29.1s, 16 kHz-sampled mp3 files.

^c Million Song Dataset consists of 214,284 audio tracks (201,680 for training and 12,605 for validation).

classification applications. The analysis of these three alternatives was aimed at improving accuracy but keeping a reduced number of layers, thus promoting an easier implementation and, therefore, a more flexible integration as a module within a complete system., e.g, a music recommendation application. The latter turns to be an appealing alternative to implement this models in mobile devices becoming in a useful tool to improve context-aware systems, e.g, for travelers discovering traditional music from the visited regions. Finally, combining the strengths of CNN models in local pattern representation and their integration into complex structures, together with the capability of RNN models to get and summarise time relations proved to be the superior architecture.

REFERENCES

- [1] S. A. Stafford, "Music in the digital age: The emergence of digital music and its repercussions on the music industry," 2010.
- [2] C. C. Aggarwal, *Recommender Systems: The Textbook*. Springer, 2016.
- [3] M. Schedl, "Deep learning in music recommendation systems," *Frontiers Appl. Math. Stat.*, vol. 5, p. 44, 2019.
- [4] M. Carnaghi and M. C. Cebedio, "Genre classification with deep learning techniques," GitHub, 2022. [Online]. Available: <https://n9.cl/asrra>
- [5] K. Choi, G. Fazekas, and M. B. Sandler, "Automatic tagging using deep convolutional neural networks," *CoRR*, vol. abs/1606.00298, 2016. [Online]. Available: <http://arxiv.org/abs/1606.00298>
- [6] G. Tzanetakis, G. Essl, and P. Cook, "Automatic musical genre classification of audio signals," 2001. [Online]. Available: <http://ismir2001.ismir.net/pdf/tzanetakis.pdf>
- [7] K. Choi, G. Fazekas, K. Cho, and M. Sandler, "A tutorial on deep learning for music information retrieval," 2018.
- [8] A. Ycart and E. Benetos, "A study on lstm networks for polyphonic music sequence modelling," in *ISMIR*, 2017.
- [9] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-y. Chang, and T. Sainath, "Deep learning for audio signal processing," *IEEE Journal on Selected Topics in Signal Processing*, vol. 13, 04 2019.
- [10] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015. [Online]. Available: <https://arxiv.org/abs/1502.03167>
- [11] A. van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *NIPS*, 2013.
- [12] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks," 2016. [Online]. Available: <https://arxiv.org/abs/1606.00298>
- [13] J. Nam, J. Herrera, and K. Lee, "A deep bag-of-features model for music auto-tagging," 2015. [Online]. Available: <https://arxiv.org/abs/1508.04999>

Espectrogramas de registros de Ballenas Barbadas, sintetizados a partir de Autoencoders

Marco Carnaghi y María C. Cebedio

ICYTE, Depto. de Electrónica y Computación, Facultad de Ingeniería - UNMDP

Mar del Plata, 7600, Argentina

{mcarnaghi,celestecebedio}@fi.mdp.edu.ar

Resumen—En este trabajo se presenta la síntesis de espectrogramas de registros de ballenas barbadas, a partir de utilizar Autoencoders Convolucionales simples. El estudio abarca el acondicionamiento de los datos, el diseño de diferentes arquitecturas, el entrenamiento y la evaluación, el análisis de la mejor opción y la generación de los espectrogramas ficticios. Los resultados obtenidos demuestran que es posible generar espectrogramas de elevada correspondencia con sonidos provenientes de ballenas barbadas, a partir de una arquitectura de pocas capas convolucionales.

Palabras Clave—Autoencoders convolucionales, espectrogramas, sonidos subacuáticos, síntesis.

I. INTRODUCCIÓN

El estudio de los registros vocales de los mamíferos subacuáticos es un área de interés e importancia biológica. Los sonidos que emiten representan el lenguaje natural de estas especies y reflejan su comportamiento. En la actualidad, el creciente interés por la conservación de las especies marinas ha propiciado un avance científico en esta área. En este sentido, las nuevas técnicas de Machine Learning resultan prometedoras y se han desarrollado trabajos muy interesantes [1], [2], [3]. Prueba de esto, en [4] se utilizan para detectar llamadas de ballenas en peligro de extinción. La ausencia del volumen requerido de datos de audio submarino, es uno de los principales problemas que se enfrentan. Esta baja disponibilidad de recursos condiciona en gran medida la utilización de técnicas, para las cuales la cantidad y calidad de los datos juega un papel central.

La generación automática de espectrogramas ficticios con las características de una señal de audio de interés puede ser beneficioso para expandir la cantidad de datos. Estos datos, generados artificialmente, podrían solucionar problemas de desbalanceo en el conjunto de entrenamiento [5].

Investigaciones actuales han desarrollado modelos generativos basados en técnicas de aprendizaje profundo no supervisado, como el Autoencoder Convolutivo (AE-CNN) y el Autoencoder Variacional (VAE), para reconstruir señales. Estos Autoencoders (AE) se caracterizan por generar una representación compacta de los datos de entrada denominada “espacio latente”, la cual es posteriormente utilizada para generar imágenes de salida similares a las de entrada. En este sentido, el análisis realizado y los resultados obtenidos en [6] para la síntesis de la risa pueden ampliarse y aplicarse a sonidos proveniente de fuentes biológicas.

Por otro lado, la generación ficticia de espectrogramas en tiempo real puede ser útil en bancos de prueba destinados al análisis de sonido submarino. En este sentido,

sería de mayor utilidad aún, contar con un equipamiento dedicado a la generación de espectrogramas de manera de independizarse de la PC. Previo a una implementación física sobre algún sistema embebido, es necesario desarrollar un modelo de arquitectura simple y de pocas capas, de manera que el costo computacional asociado a las operaciones matemáticas y a la cantidad de parámetros se reduzca. Trabajos en el área han demostrado que con grandes redes convolucionales se pueden obtener excelentes resultados, pero estas redes poseen una gran cantidad de parámetros y las operaciones matemáticas se incrementan con cada capa que se añade. El problema reside actualmente en minimizar el modelo de red para que luego puedan ser implementados en sistemas embebidos.

En este trabajo se propone el análisis y evaluación de diferentes arquitecturas sencillas de AE, para el proceso de síntesis de espectrogramas de sonidos de ballenas barbadas. El fuerte hincapié en obtener un modelo simple radica en la posible implementación del mismo sobre sistemas que disponen de una potencia computacional muy limitada. Esta limitación no es un gran impedimento en redes neuronales sencillas sin mucha complejidad o sin muchos cálculos, pero en redes más complejas sí.

II. METODOLOGÍA

La metodología de síntesis consiste en transformar el espectrograma logarítmico de alta dimensión, en un vector latente de baja dimensión. Este vector contiene información valiosa que se utiliza para reconstruir un espectrograma de magnitudes sintéticas. El proceso se desarrolla siguiendo los siguientes pasos: adecuación de los datos de entrada, entrenamiento del modelo y obtención de métricas, selección del mejor modelo a entrenar y generación de espectrogramas ficticios. Los pasos enumerados anteriormente no son estáticos. La selección de características y evaluación es iterativa.

II-A. Adecuación de los datos de entrada

El objetivo de esta etapa es obtener una matriz bidimensional que represente el espectrograma de magnitud logarítmica para cada registro de audio. Esta representación matricial de la energía del contenido frecuencial de una señal puede pensarse como una imagen en escala de grises.

El conjunto de todos los espectrogramas normalizados, correspondiente a registros de audio de diferentes tipos de ballenas barbadas, se agrupa en un tensor. Originalmente, estos datos provienen de diferentes fuentes y para su tratamiento se normalizan bajo las siguientes características:

Frecuencia de remuestreo= 44.1KHz, Duración temporal del registro de audio=1s, Tipo de ventana para espectrogramas=Tukey, Cantidad de puntos por bloque para STFT =256, Normalización de Amplitud: -150 a 150 dB.

Para automatizar el proceso de normalización, se diseña un programa [7] que permite customizar estas características. De esta manera se obtiene un tensor de 3 dimensiones (S_x, S_y, N), siendo 'N' la cantidad de muestras de 1 seg., S_x y S_y los valores de la matriz de espectrograma. En este punto, los datos aún no son adecuados para el entrenamiento de los AE. Por lo tanto, se realiza: la adecuación de la dimensión de los datos de entrada a ($N, S_x, S_y, 1$); la división de los datos en entrenamiento (D_{Train}), validación (D_{Val}) y test (D_{Test}); y una normalización entre 0 y 1.

II-B. Arquitecturas de Autoencoders

Se generan dos tipos de arquitecturas: AE-CNN y VAE. Como se muestra en la Fig. 1, el AE-CNN posee una sección de codificación basada en redes convolucionales, en la que se realiza una sección de extracción de características, una capa flatten (utilizada para uni-dimensionalizar la entrada) y una última red densamente conectada. A la salida se obtiene un vector representativo (espacio latente), a partir del cual y replicando a la inversa el proceso de codificación se intenta recuperar la imagen de entrada.

La arquitectura VAE es similar al AE-CNN con la diferencia de la capa de parámetros probabilísticos después de la capa Flatten [8]. Por lo tanto, en lugar de aprender directamente las características latentes de las muestras de entrada, aprenden la distribución de las características latentes. Los VAE aseguran que los puntos que están muy cerca uno del otro en el espacio latente representan muestras de datos muy similares [9].

Las arquitecturas propuestas poseen en la etapa de extracción de características 4 capas. En todas las etapas la función de activación es tipo RELU y el Padding (asociado a la etapa de filtrado) se re-acomoda de forma automática. Se realiza la operación de Batch Normalization como capa Pooling. Esta capa acelera el proceso de entrenamiento, mejora las propiedades de normalización de la red y la vuelve más robusta a diferentes esquemas de inicialización y tasas de aprendizaje [10]. Resumiendo la arquitectura empleada: 1° capa: 32 Filtros de $3 \times 3 \times 1$ con stride= 1; 2° capa: 64 filtros de $3 \times 3 \times 1$ con stride= 2; 3° capa: 64 filtros de $3 \times 3 \times 1$ con stride= 2; 4° capa: 64 filtros de $3 \times 3 \times 1$ con stride= 1.

Para el caso del AE-CNN, se diseña una sola capa densa en la sección de capa profunda y para el VAE dicha capa está representada por capas densamente conectadas de varianza y media. Respecto a la dimensión del espacio latente, esta es una consideración de diseño que hace a la arquitectura de la red. En este caso se decide obtener

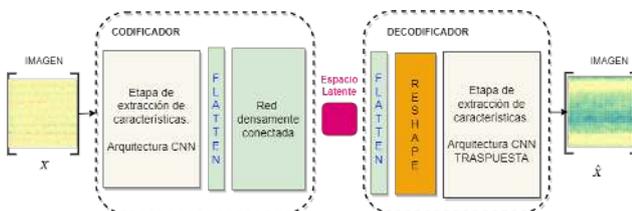


Fig. 1: Arquitectura de un Autoencoder CNN genérico.

diferentes arquitecturas, a partir de cambiar únicamente esta dimensión.

II-C. Entrenamiento

El objetivo de entrenar la red es obtener un Autoencoder que reconstruya el espectrograma a su entrada. Los Hiperparámetros asociados al entrenamiento se establecen en: función de pérdida=MSE; optimizador=Adam; tasa de aprendizaje: 0.0005; épocas=60(CNN) y 48(VAE); tamaño de Minibatch=100; D_{Train} =6043 con $\%D_{Val}$ =0.2.

II-D. Elección de las arquitecturas

Para la elección de la dimensión del espacio latente, se entrenan la AE-CNN y la VAE, con el set de datos correspondiente a las ballenas Barbadas, variando la dimensión entre tres valores posibles: 2, 3 y 4. A partir del análisis de los resultados del entrenamiento que se muestran en la Fig. 2, se seleccionan las arquitecturas AE-CNN con $D=4$ y VAE con $D=3$.

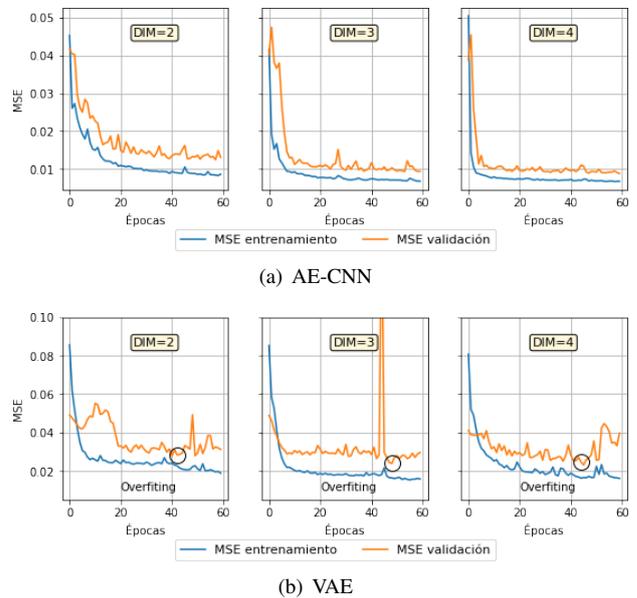


Fig. 2: Datos de Entrenamiento para arquitecturas AE-CNN y VAE con diferentes dimensión del espacio latente.

En la Fig. 3 se observa que, aún en el mejor caso de VAE, AE-CNN posee un orden de magnitud menos de error y éste cae notablemente con pocas épocas de entrenamiento.

El MSE es un indicativo de buen desempeño, pero es conveniente evaluar con una imagen de Test para corroborar el significado de dicho valor. Además, a modo de obtener otro parámetro de medición, se calcula el error de similitud estructural (SSIM), que ofrece buena precisión de evaluación y simple formulación [11]. En la Fig. 4 se presentan los resultados, indicando que el mejor modelo a implementar es el AE-CNN.

III. GENERACIÓN DE ESPECTROGRAMAS FICTICIOS

Para generar espectrogramas ficticios se utiliza el bloque decodificador del AE-CNN. El problema radica en definir los valores del espacio latente de entrada. Con el fin de obtener muestras representativas, se realizan los siguientes pasos:

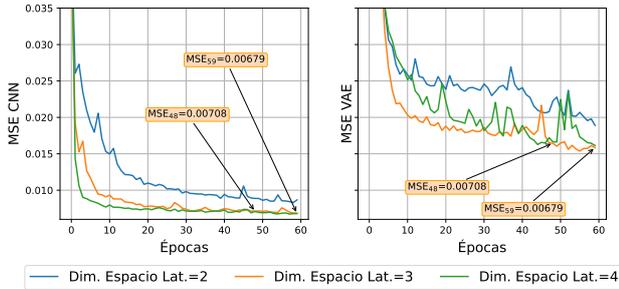


Fig. 3: MSE de entrenamiento para diferentes dimensiones del espacio latente en las arquitecturas AE-CNN y VAE.

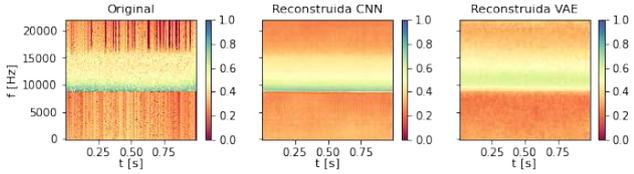


Fig. 4: Imagen reconstruida utilizando un AE-CNN y VAE. Datos: $MSE_{CNN}=0.0052$, $SSIM_{CNN}=0.580$, $MSE_{VAE}=0.011$, $SSIM_{VAE}=0.497$.

1. Con los vectores de espacio latente generados en el entrenamiento, se obtiene una matriz de dim. $[N_{Train}, 4]$.
2. Se analiza la interdependencia entre los datos de cada vector de espacio latente (fila de la matriz). En 1, la matriz de correlación muestra que no existe una marcada relación intra-vector.

$$\begin{bmatrix} 1 & 0,125 & -0,071 & -0,074 \\ 0,125 & 1 & 0,073 & 0,143 \\ -0,071 & 0,073 & 1 & -0,142 \\ -0,074 & 0,143 & -0,142 & 1 \end{bmatrix} \quad (1)$$

3. Se obtiene el valor medio y la desviación estándar de la distribución de los valores de cada vector.
4. Se genera un vector de código aleatorio $[X_0, X_1, X_2, X_3]$ siguiendo una distribución Gaussiana con la media y desviación estándar encontradas en el paso 2.

De esta manera se busca que el vector generado sea representativo. Las Figs. 5 y 6 muestran ejemplos de imágenes reales y sintetizadas, con los modelos entrenados en [12].

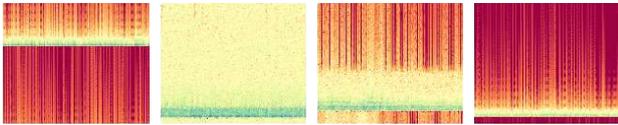


Fig. 5: Espectrogramas Reales, obtenidos a partir de registros aleatorios de ballenas Barbadas.

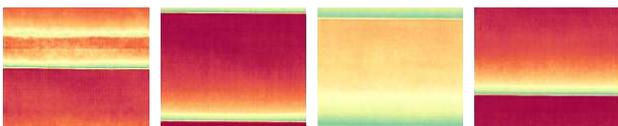


Fig. 6: Espectrogramas sintetizados aleatoriamente, obtenidos con un decodificador entrenado, a partir de registros de ballenas Barbadas.

En ambos casos, se generan imágenes aleatorias, no hay correspondencia entre ellas.

IV. CONCLUSIÓN Y TRABAJO A FUTURO

En esta trabajo se demuestra la posibilidad de generar registros de espectrograma ficticios, a partir de un codificador automático reducido y a partir de pocas muestras. Se observa que un modelo general simple permite aprender características latentes efectivas para diferentes espectrogramas de sonidos subacuático de baja frecuencia.

En general, las implementaciones de redes profundas se realiza sobre CPU's debido a la gran carga computacional que conlleva realizar la cantidad de operaciones asociadas a cada capa de la red. Un modelo simple como el mostrado puede ser entrenado en una PC personal (sin recurrir a GPU o entrenamiento en la nube) y permite, a futuro posibles implementación sobre sistemas embebidos que posean limitada potencia computacional.

Dado que el modelo propuesto posee una cantidad reducida de capas convolucionales y de parámetros asociados, se reducen las operaciones involucradas y los errores asociados a ellas debido a la limitación física en la cantidad de bits que presentan las arquitecturas de sistemas embebidos simples.

V. AGRADECIMIENTOS

Al Dr. Diego Comas y al Dr. Gustavo Meshino por los conocimientos impartidos sobre la temática.

REFERENCIAS

- [1] E. Tejero, "Aplicaciones de Machine Learning a la Bioacústica Marina," Ph.D. dissertation, 07 2020.
- [2] D. Tuia and E. Al, "Perspectives in machine learning for wildlife conservation," *Nature Communications*, vol. 13, no. 792, 2022.
- [3] A. Lamba, P. Cassey, R. Raja Segaran, and L. Koh, "Deep learning for environmental conservation," *Current Biology*, vol. 29, pp. R977–R982, 10 2019.
- [4] A. Ibrahim and et. al, "A multimodel deep learning algorithm to detect North Atlantic right whale up-calls," *The Journal of the Acoustical Society of America*, vol. 150, 08 2021.
- [5] H. Ali, M. Salleh, R. Saedudin, K. Hussain, and M. Mushtaq, "Imbalance class problems in data mining: A review," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, 03 2019.
- [6] N. Mansouri and Z. Lachiri, "Laughter synthesis: A comparison between Variational autoencoder and Autoencoder," in *5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, 2020, pp. 1–6.
- [7] M. C. Cebedio and M. Carnaghi, "Matriz de Espectrogramas," GitHub, 2022. [Online]. Available: <https://acortar.link/MKnoYk>
- [8] Q. Xu, Z. Wu, Y. Yang, and L. Zhang, "The difference learning of hidden layer between autoencoder and variational autoencoder," in *29th Chinese Control And Decision Conference*, 2017, pp. 4801–4804.
- [9] C. Doersch, "Tutorial on Variational Autoencoders," 2016. [Online]. Available: <https://arxiv.org/abs/1606.05908>
- [10] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," 2015. [Online]. Available: <https://arxiv.org/abs/1502.03167>
- [11] A. León-Batallas, J. Bermeo-Paucar, Paredes-Quevedo, and H. Torres-Ordoñez, "Una revisión de las métricas aplicadas en el procesamiento de imágenes," *RECIMUNDO*, pp. 267–273, 2020.
- [12] M. Carnaghi and M. C. Cebedio, "Autoencoders," GitHub, 2022. [Online]. Available: <https://acortar.link/YgWCOY>

Detección de eventos y estimación de volumen de captura en buques pesqueros mediante visión artificial

Nicolás E. Horro

*Departamento de Ingeniería de Software - Gerencia de Ingeniería y Producción
INVAP S.E.*

San Carlos de Bariloche, Argentina
nhorro@invap.com.ar

Resumen—Se describe el prototipo de un sistema para detección de eventos y estimación de volumen de captura en buques pesqueros que utiliza técnicas de visión artificial. La solución consiste en un software que puede ejecutarse en una PC industrial o también en un dispositivo *Edge* con GPU para complementar un sistema de CCTV existente. Se utilizan detectores y *trackers* de objetos, y otros algoritmos para identificar los momentos en que se realizan algunas maniobras de pesca o se manipula la captura.

Palabras clave—pesca industrial, visión artificial, YOLO, DeepSORT, inteligencia artificial

I. INTRODUCCIÓN

La pesca industrial es un tipo de pesca que tiene como objetivo obtener un gran número de capturas. En Argentina el sector primario pesquero, que es el que se ocupa de la captura, está compuesto por una flota de buques fresqueros de altura, de costeros grandes y costeros chicos y una flota de buques procesadores. La flota fresquera de altura está conformada por barcos arrastreros con bodegas refrigeradas que cuentan con equipamiento de navegación, detección y utilizan redes de arrastre [1].

A fin de garantizar que la pesca a esta escala sea sostenible, el Consejo Federal Pesquero tiene como función establecer la política pesquera nacional. Esto incluye la planificación del desarrollo pesquero nacional, las políticas de investigación del recurso, la determinación de la Captura Máxima Permisible (CMP) por especie, la aprobación de los permisos de pesca comercial y experimental y el establecimiento de cánones para el ejercicio de la pesca, entre otros.

El monitoreo y la vigilancia deben permitir conocer las características del esfuerzo en la actividad pesquera y asegurar que las capturas se realicen dentro de los cánones admitidos. Existen también regulaciones que establecen el modo en que debe emplearse la técnica, por ejemplo, exigiendo una permanencia mínima de las redes sumergidas para disminuir la captura incidental de mamíferos marinos.

Actualmente existe una creciente tendencia a la utilización de sistemas electrónicos de seguimiento (SE) y un mayor grado de automatización, que representan una alternativa más

eficiente y rentable que la elaboración de reportes manuscritos [3]. Estos sistemas pueden incluir cámaras que registran la actividad en puntos estratégicos del buque, pero aún así un esfuerzo significativo sigue siendo requerido para la inspección de las imágenes registradas.

Este reporte, que está basado en un proyecto de finalización de carrera [2], describe la implementación de un software que puede ejecutarse en una computadora del barco y automatizar parte de la tarea de detección de eventos o conteo de piezas capturadas mediante técnicas de inteligencia artificial. A modo de ejemplo, la figura 1 muestra una escena de un buque tangonero en la que se han identificado objetos y definido regiones poligonales para reconocer eventos de interés e identificar algunas maniobras que incluyen la manipulación de redes o clasificación de captura, entre otras.



Fig. 1. Ejemplo de escena de buque tangonero en la que se han identificado operadores, delimitado regiones de interés, y en la que se clasifican de manera automática las actividades realizadas.

II. ARQUITECTURA

La figura 2 muestra el flujo de datos entre los principales bloques funcionales del sistema, que son:

- **Adquisición de video:** consiste en obtener los cuadros de una cámara o un archivo de video y aplicar una corrección de imagen para reducir las diferencias entre cámaras.

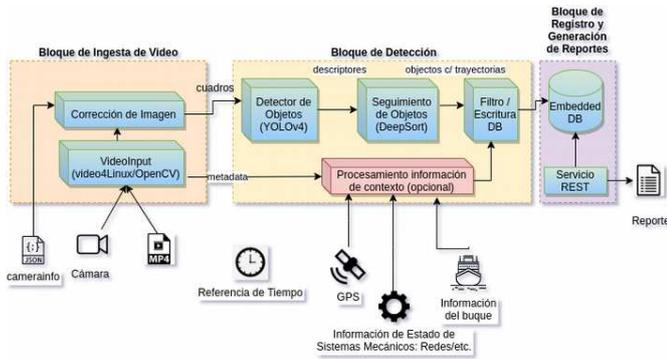


Fig. 2. Descomposición funcional del sistema identificando tecnologías y flujo de datos.

Esta corrección puede incluir: contrarrestar distorsión de lente de pez, aplicar transformaciones de coordenadas cromáticas o ecualizar la imagen, entre otras.

- Detección de objetos: relevar posiciones de objetos en escena con el algoritmo YOLOv4 [4].
- Seguimiento: construir trayectorias de los objetos detectados con los algoritmos SORT [5] y DeepSORT [6].
- Incorporación de información de contexto (opcional): obtener velocidad del buque de sus GPS, accionar de malacates u otra información.
- Filtrado, registro y generación de reportes: procesar la salida de las etapas anteriores aplicando filtros y reglas para extraer sólo los eventos de interés o aumentar la confianza de las predicciones. Un ejemplo es eliminar las predicciones por debajo de un umbral de confianza, o retener únicamente la actividad de algunos objetos específicos en algunas regiones. La salida es un reporte con horarios, tipos de capturas o cantidad de descartes, que se publica en una base de datos.

III. MODELOS DE APRENDIZAJE SUPERVISADO

Para este trabajo se han utilizado modelos que tienen como entrada dos tipos de datos: imágenes de cuadros de video y posiciones de objetos generadas por un detector. En ambos casos se ha omitido la dimensión temporal.

A. Detectores

Se entrenaron dos tipos de detectores: uno para detección de operadores, redes y capturas en la pesca de langostinos [7] y otro para presas grandes utilizando datos de Kaggle [8].

Detectar piezas individuales de manera automática con un margen de error bajo es posible cuando las presas captadas por la cámara exhiben características morfológicas evidentes como por ejemplo aletas, tamaño o color -que permiten distinguirlas y delimitarlas-, pero no resulta factible para piezas pequeñas donde el límite entre un objeto y otro no es claro o hay una alta superposición entre los objetos, como ocurre en la pesca de langostinos.

En casos como este último, en cambio sí es posible detectar los operadores a bordo, las redes cargadas y los pescados

grandes, que sin importar su tipo, siempre serán considerados descartes. Detectar los operadores permite estudiar su movimiento y establecer patrones de comportamiento nominal, y esto a su vez luego permite automatizar la detección de patrones anómalos. Detectar las redes es un paso previo para estimar el volumen de su carga o para aislar los cuadros de video en que es visible. La tabla I presenta la matriz de confusión para la evaluación de los modelos de detectores entrenados para el caso de aplicación de escenas de pesca de langostinos y para detección de piezas grandes.

TABLA I
DESEMPEÑO DE MODELOS DE DETECCIÓN

Detector	Clase	Nombre	maP	TP	FP
Langostinos	0	Operador	93.48	73	10
	1	Red	78.85	5	0
	2	Pescado	63.03	22	7
Presas grandes	4	Tiburón	94.63	129	11
	3	Lampris guttatus	92.08	73	3
	6	Otros	90.95	233	46
	5	Thunus albacores	90.59	495	91
	0	Thunnus alalunga	90.13	1590	198
	2	Coriphaena hippurus	76.72	68	10
1	Thunnus obesus	66.96	150	35	

maP: Mean Average Precision, TP: True Positives, FP: False Positives

B. Eventos activados por regiones de interés

Las regiones de interés (abreviadas ROIs por *Regions of Interest*) son regiones poligonales que pueden ser sometidas a evaluaciones geométricas de contención e intersección, como por ejemplo si incluyen un punto o se intersectan con otra forma geométrica. En video analítico se utilizan para evaluar la presencia de objetos, actividad en un área, u otros eventos de interés, como por ejemplo:

- Si un objeto ingresa o egresa de una ROI.
- El tiempo que un objeto permanece en una ROI.
- La cantidad de objetos de un determinado tipo en una ROI.

Combinadas con la salida de un detector de objetos, un algoritmo de seguimiento o un estimador de movimiento, pueden proporcionar información sobre lo que está ocurriendo en una escena.

C. Clasificador de actividades

Luego de observar el video y de sospechar que existía una relación entre la forma en que se desplazaban los operadores en la cubierta del barco para realizar distintas actividades, se consideró conveniente explorar la opción de reemplazar las detecciones individuales por variables estadísticas, de modo que para cada cuadro del video exista una única observación que contenga momentos estadísticos de interés, quedando los datos de entrada formados por las siguientes columnas:

- n : cantidad de detecciones.
- $mean_x, mean_y, mean_{area}$: media muestral de las posiciones y área de todas las detecciones.

- s_x, s_y, s_{area} : desvío estándar muestral de las posiciones y área de todas las detecciones.
- var_x, var_y, var_{area} : varianza muestral de las posiciones y área de todas las detecciones.
- $skew_x, skew_y, skew_{area}$: oblicuidad de las posiciones y área de todas las detecciones.
- $kurt_x, kurt_y, kurt_{area}$: kurtosis de las posiciones y área de todas las detecciones.

Las figuras 3 y 4 muestran la evolución temporal y distribución de tres de las quince variables agregadas. En la observación parece existir una correlación entre los valores que adquiere una determinada combinación de estas variables y el tipo de actividad que se está realizando.

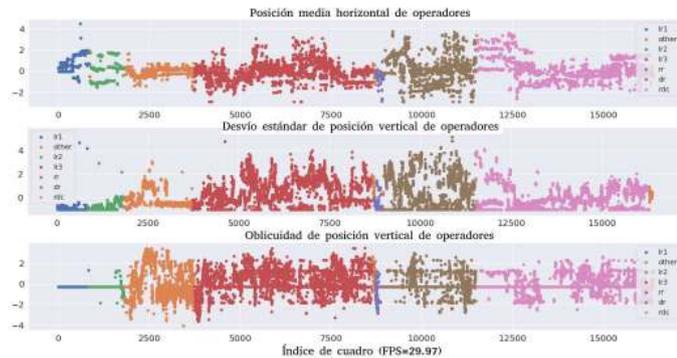


Fig. 3. Evolución en el tiempo de media de posición horizontal y desvío estándar y oblicuidad de posición vertical de operadores. Se indican los tres momentos de levado de redes (lr), recogimiento de redes (rr), descarga de redes (dr) y recolección y clasificación (rdc).

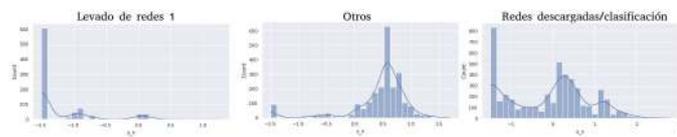


Fig. 4. Histogramas para desvío estándar de posición horizontal de operadores en distintos momentos del video.

Los modelos entrenados para clasificación de actividades incluyeron regresión logística con y sin transformación de cuantil, AdaBoost, Random Forest, Gradient Boost y K nearest neighbors (KNN).

En todos los casos se implementó un paso de estandarización de las variables de entrada. En algunos casos se agregaron otros tipos de preprocesamiento como la transformación de cuantil para evaluar si tenían un impacto positivo en el desempeño.

Para cada modelo se registró el puntaje sobre la partición de entrenamiento y de evaluación con la métrica AUC y el tiempo de inferencia promedio por muestra (obtenido sobre la partición de evaluación).

Como se puede observar en la figura 5, en general todos los modelos tienen un buen desempeño, superando el 70%. Con excepción de KNN, el resto de los modelos tienen un tiempo de inferencia por debajo de $2 \mu s$ en un microprocesador i7-6700HQ, por lo que son aptos para inferencia en tiempo real.

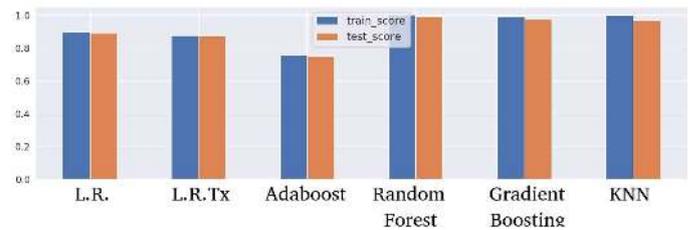


Fig. 5. Desempeño de los modelos de clasificación (AUC).

IV. CONCLUSIONES

Las técnicas de inteligencia artificial consideradas muestran ser de utilidad para la automatización total o parcial de algunos de los procesos requeridos para la fiscalización de la actividad pesquera. En el caso de piezas grandes o para la extracción de imágenes de redes de pesca, el uso de detectores, *trackers* y *ROIs*, presentó un desempeño cercano al de un observador humano en los videos de prueba. En el caso de la pesca de langostinos, tener la capacidad de identificar los intervalos en los que ocurren actividades críticas como la manipulación de la red o el tratamiento del descarte, reduce significativamente el esfuerzo de análisis posterior y puede aplicarse para activar el registro del video a bordo sólo cuando es relevante.

La decisión de adoptar una arquitectura de software que separe un problema complejo en bloques que realizan tareas específicas sobre los datos disponibles permite definir cadenas de procesamiento adecuadas para distintas aplicaciones que se benefician de la reutilización de modelos y código, reduciendo así el costo de implementación.

Este trabajo puede continuar extendiéndose para la detección y conteo de otros tipos de pescados e incorporar técnicas de estimación de volumen, pero a medida que se incorporen capacidades automáticas, deben también considerarse mecanismos para garantizar la robustez del sistema y detección de fallas o posibles comportamientos anómalos.

REFERENCIAS

- [1] Andrea Pagani y Patricia Gualdoni, "2do Informe de Monitoreo Ciudadano - Sector Pesquero". URL: <http://nulan.mdp.edu.ar/3004/1/pagani-gualdoni-2018.pdf>. Visitado el 2021-09-05.
- [2] Nicolás Horro, "Detección de eventos y estimación de volumen de captura en buques pesqueros mediante visión artificial". Trabajo final de Carrera de Especialización de Inteligencia Artificial de FIUBA, 2021.
- [3] Erik Bergh y Sandy Davies, "A Fishery Manager's Guidebook - Management Measures and Their Application", capítulo 8 "Fishery Monitoring, Control and Surveillance", FAO (Food and Agriculture Organization of the United Nations). URL: <http://www.fao.org/3/y3427e/y3427e0a.htm#bm10.1>. Visitado el 2021-09-05.
- [4] Alexey Bochkovskiy and Chien-Yao Wang and Hong-Yuan Mark Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection", 2020.
- [5] Bewley, Alex and Ge, Zongyuan and Ott, Lionel and Ramos, Fabio and Upcroft, Ben, "Simple online and realtime tracking", 2016.
- [6] Wojke, Nicolai and Bewley, Alex and Paulus, Dietrich, "Simple Online and Realtime Tracking with a Deep Association Metric", 2017.
- [7] "Maniobra realizada para la pesca de langostino en el buque tan-gonero Mirta R en la zona de Playa Unión, en la provincia de Chubut". URL: https://www.youtube.com/watch?v=zGqXKR_wvFU. Visitado el 2021-09-05.
- [8] "Kaggle - The Nature Conservancy Fisheries Monitoring. Can you detect and classify species of fish?". URL: <https://www.kaggle.com/c/the-nature-conservancy-fisheries-monitoring>. Visitado el 2021-09-05.

Computational Creativity through AI modeling

Fernando Furundarena
CI2S Labs
Buenos Aires, Argentina
fernandof89@gmail.com

Daniela López De Luise
CI2S Labs
Buenos Aires, Argentina
ORCID 0000-0003-3130-873X

Micaela A. Veiga
CI2S Labs
Buenos Aires, Argentina
micaela.a.veiga@gmail.com

Abstract— The main goal of this paper is to introduce a novel Artificial Intelligence creativity model based on sensor's information, a set of art pieces, user emotions, and real world data. There is a short introduction to the state-of-the-art artificial intelligence' models that can analyze and produce creative audiovisual expressions. Then the proposal combines several models to create an artistic experience of interaction between a computer and a human observer. As a final step a case study is presented and analyzed. The first results indicate that despite the controlled context the observer presents an animism belief, a human brain bias, and it has a strong influence in the interaction of the individual with the system.

Keywords: Art; Artificial Intelligence; Human and Machine Interaction; Computational Creativity.

I. INTRODUCTION

Artificial Intelligence is a field that keeps adding useful solutions to our everyday life, from home applications to industrial processes. As complexity and capabilities of these systems grow, a common question comes out: Can Computational Intelligence replace human intelligence? Can it create art, in similar fashion to human artistic expressions? This paper explores how, through non deterministic modeling, it is possible to create audiovisual expressions based on Artificial Intelligence, how it would affect human's mind and behavior. The question if Art could be or not part of human subjectivity and bias has been the source of plenty of news and articles ([1] [2], [3], etc.), but is out of the scope of this work.

The present article explains the preliminary part of project Lumiere whose main goal is to introduce an AI model that takes certain interesting advanced tools for manipulating images, and detecting human emotions to produce a new multimedia dynamical version of a piece of art.

The art produced this way constitutes a novel expression following Bourriaud conception [4]. He suggests that art can be an exchange between two subjects, or a subject and an object, where one part shows something, the other receives it, transforms it and shows it back. He compares this exchange with a dialogue when he cites Daney: "all form is a face looking at me".

But there are many ways to collaborate with the observer. For instance telematic or telecommunication, where artists utilize interactive media like telephones, computers, e-mails, television, radio, satellites, etc. It is thought that these media are not legitimate, but Kac suggests getting rid of these prejudices. He suggests that telecommunication art can be the culmination of the dematerialization process of the artistic object, as exemplified by Duchamp and as pursued by concept artists. Getting rid of the object and the artist, the aesthetic debate can be found in the interactions between members of a network.

These telecommunication art pieces tend to not be experienced as unidirectional messages. Interaction replaces contemplation with a participative experience. Unlike linear communication, which privileges the artist as a message coder, telematics utilize multidirectional communication, where the artist creates a context for facilitating interactions. Provides the observer, transformed into an user or participant, with the

codes and tools the artist utilizes, allowing the work's meaning to be negotiated by both. [5]

Back to Bourriaud, the artist explains that contemporary art consists, frequently, of pieces that can only be seen in a single place and time frame, forging a unique bond with the public. He makes sure to differentiate between the piece and its recording as completely different experiences.

When talking about art produced, completely or partially, within galleries or museums, he says that "[...] art is made in the gallery [...]", referring to performances or interactive art.

He says there has been an increase in the amount of participative artistic projects that take the public into account, transforming it into part of its aura [4]. Lumiere is a research project looking for adding humans in the artistic conception produced with the dynamic interaction of the expositive object and the visitor. There is a subprocess in the prototype that watches the human faces as a clue to infer his mood, and this new information is expressed as a brushstroke in the current artistic performance.

Simon Penny considers an interactive system as successful depending on how much is the user capable of interpreting its responses as related to their conduct, as if engaged in a dialogue [6]. Part of the research considers if this interaction could be thought of as a successful expression of visitor's interpretation of the image being exposed.

It is important to note that the artwork in this context introduces a new and unique aesthetic and converts the traditional static canvas into a dynamic experience. According to Burnham, AI aesthetics will evolve with time, as technology's influence on art will end up with art showing true intelligence and establishing a two-way relationship with humans, making the word "observer" obsolete [7]. People working on installation art, robotic art, or involved in computer-based multimedia forms refer to 'users' or 'visitors' as opposed to 'observers'. The experience ceases to be a passive contemplation and becomes interaction.

This project belongs to CI2S Labs, in collaboration with the National Academy of Sciences in Buenos Aires (Argentina). The rest of this paper is organized as follows: Section II speaks of the relation between digital systems and Art, Section III is some of the state of the art in AI models of interest; section IV describes the proposal and the model creating an interaction with a human observer; section V is a case study and section VI has conclusions and future work.

II. COMPUTATIONAL INTELLIGENCE AND MULTIMEDIA

This section presents several tools that make use of intelligent algorithms in order to manipulate images, video and/or sound. The list is not complete but focused on the approaches of interest as background for the proposal.

A. 3D inferred Scenes and View Synthesis

The transformation of a set of 2D images into a 3D video is a machine resource demanding but profitable task. Several authors have proposed approaches to perform this task using intelligent algorithms [8]. For instance with Neural Networks (NN) [9], depth-map generation followed by a depth image-

based rendering [10], a micro-system interposer based on screen coordinate conversion to OpenGL coordinates [11], or wavelets combined with a mathematical estimation of figure's edges [12]. It is of special interest here the project NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis [13], an algorithm that represents a scene using a fully-connected (non-convolutional) deep network. Figure 1 presents a summary of the workflow.

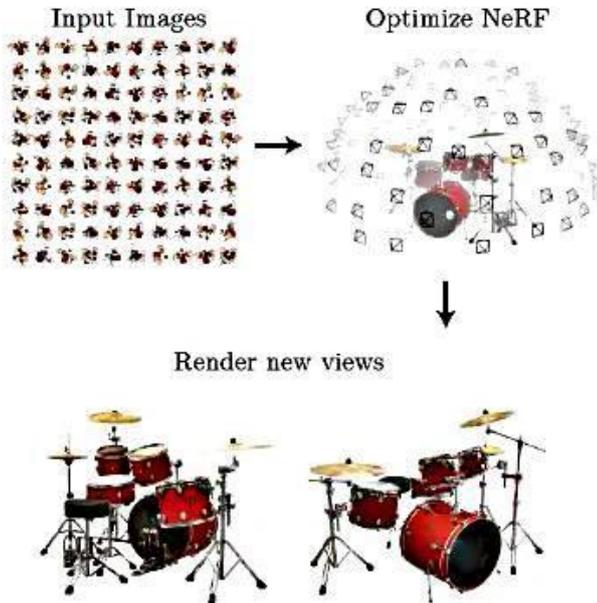


Fig. 1. Example of a figure before (left) and after (right) DeOldify processing

B. Photo manipulation with Machine Learning

Most of the applications in this field are related to image segmentation for object detection and tracking, but in the current context the focus is on color and contour detection and manipulation. Some proposed solutions are Direct Color Imaging overlapping channels to improve the quality of filters application [14], image enhancement and space-variant color reproduction by an adaptive sigmoid function [15], total Variation Minimization Methods for Color Image Restoration [16], Clifford algebras to define Color Monogenic Signal associated to an image [17], etc. For the current project it is important to detect, change and combine colors and brightness levels, like in [18] that uses a kind of calibration pattern based on image geometry, or [19] that applies a Nonsubsampled contourlet transform and $L\alpha\beta$ color space [20]. A simpler but less effective solution is based on pixel similarities comparison [21], and a two-step proposal with Fuji Intelligent Color Enhancement and entropy [22]. In this project DeOldify [23] is used: a photo-retoucher based on two Generative Networks. Antagonistic Algorithms, implemented by a system of two neural networks that compete in a kind of game zero sum. Figure 2 is an example of a photo and the inferred colors.



Fig. 2. Example of a figure before (left) and after (right) DeOldify processing

C. Video Depth Estimation

Depth estimation is important to rebuild 3D video in modern TV technology, and for virtual 3D synthesis. In this pro-

ject, it is considered as a feature used to distort the user's visualization. Among others, there are adaptive motion-estimation-mode selection for depth video [23], adaptive support-weight block matching to extract disparity information from consecutive video frames [24], Sparse-to-Dense Depth Estimation in Videos via High-Dimensional Tensor Voting [25], View-consistent multi-view depth estimation for three-dimensional video generation [26], models with a unified energy function applied to depth estimation from asymmetric stereoscopic videos [27], etc. In the virtual 3D synthesis an interesting proposal is in [28]: authors build a smooth and accurate depth 3D map with a special video coding that preserves details in the depth map and simultaneously smooth depths in object boundary occluded area, less texture and discontinuity regions, to avoid blurring in the synthesized views. In [29], [30], and [31] the estimation is for compensation of 3D video compression. Here, in this project Siggraph is used: Consistent Video Depth Estimation [32], that uses a Convolutional NN (CNN). Figure 3 shows an input photo and its derived depth map.

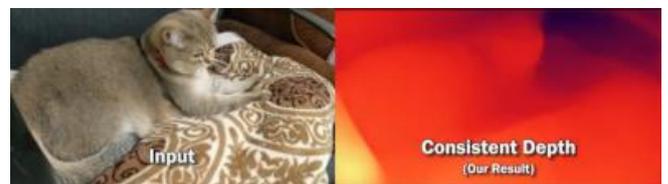


Fig. 3. An input image (left) and its estimated depth (right) using Siggraph

D. Video-Driven Speech Reconstruction

Speech-reading infers phonetic information from visually observed articulatory facial movements. It is useful to restore audio of silent videos automatically and to help deaf people. There is a strong relationship between speech-reading and single-word reading, which is fully mediated by phonological awareness [33] [34] [35]. There is very little research in this area. One interesting proposal is in [36] that presents a model based on a CNN. For this project the option is Speech: Video-Driven Speech Reconstruction using Generative Adversarial Networks [37], based on GANs, capable of producing natural sounding, intelligible speech which is synchronized with the video. Figure 4 shows a summary of its workflow.

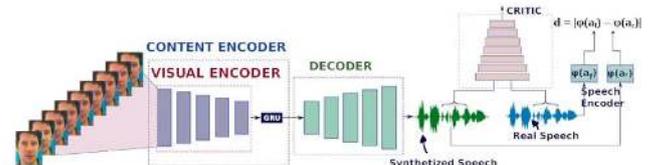


Fig. 4. Main workflow of GAN for speech inference

E. Face Recognition and Facial Attribute Analysis

The most popular algorithms for this type of task are based on NN [38] and are able to detect and track faces, eyes, nose, mouth, etc. Many can also detect emotions and certain individual's peculiarities [39]. DeepFace is used here: A Lightweight Face Recognition and Facial Attribute Analysis Framework [40], modeling with a nine-layer DNN.

III. THE PROPOSAL

The proposal consists of two notebooks connected via direct cable. Figure 5 shows the architecture. Both computers are located one in front of the other in such a way that both observers (O_1 and O_2) are back to back and can't see the counterpart activity (with different animations and music). Every computer uses a set of predefined features (D_f) to change the output of the counterpart's computer. The set of Dfs is still open but at present are: Observer's mood (by means of

DeepFace), distance (with Siggraph), and lip movement (using VDSR), and environment sounds, taken from a microphone. Among other tools, the prototype is expected to use DeOldify to dynamically alter the colors of the presented images according to the active-features. A grid of 3 cameras (see Figure 4) is for 3D synthesis from 2D images on the screen, when the prototype considers it relevant to change 2D to 3D. The conversion is performed by NeRF. VDSR is used to alter image and mic information of spoken words and images being presented. The current prototype version (a beta version) uses Style Transfer [41] software to perform most of the visual effects.

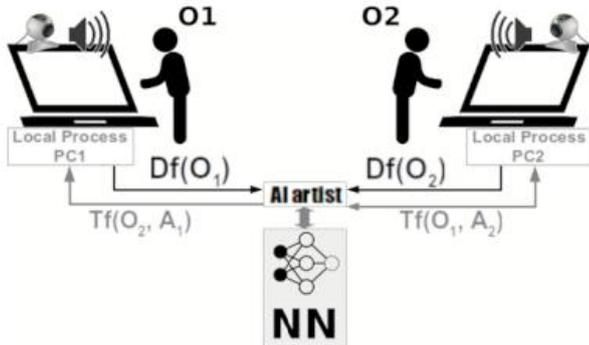


Fig. 5. Global configuration of the prototype

The model, based on a NN trained to make decisions with random variations, processes images and music to be presented in the corresponding notebook (i.e. the one being observed by the counterpart). This cross feeding is performed in both computers and intends to function as a mood connection between both participants, which will be watching and listening the result of certain combination of a human mood (the one of the counterpart), original artistic piece transformed by the IA model (Tf) following its learned “creative” approach and the current set of active-features. The interaction between computers is by using Style Transfer.

IV. CASE STUDY

The case study aims to determine whether the hardware and software configuration are suitable for the proposed activity. It is an initial test limited to one volunteer with one computer. The hardware is an Intel Core i7, with 4 GB DDR4-2400 SDRAM, running with a video graphics NVIDIA® GeForce® GTX 1050 (4 GB GDDR5 dedicated). The Hard drive is a 1 TB SATA. The Display has a 15.6" diagonal. The webcam is an HD Camera with integrated dual array digital microphone. The software configuration is also restricted to basic image processing, mood detection, color and textures transformation, music combination and a few more features. Everything running under Windows 10 operating system. The low performance resulted in several interruptions during the interaction, mainly due to rendering, so it had to be partially performed in advance and part in parallel in future implementations (probably with a FPGA). The reactions are registered with a log file and recorded with a video camera. Table I is a summary of user interactions during the test.

TABLE I
SEQUENCE OF TRANSFORMATIONS IN THE TEST

Mood	Table Column Head		
	Previous status	Final status	Expression
Neutral	NO		
Sad			
Horror			
Surprise			

Table II presents some of the music sheets corresponding to some of the emotions. Music was specially written by a musician from the National Manuel de Falla Superior Conservatory and played with MuseScore ©. The prototype modifies images and music according to moods.

TABLE II
MUSIC SHEET FOR EACH MOOD

Mood	Music		
	Previous status	Tempo	Instruments
Surprise		80	Flute Violin 3 cellos
Neutral		120	Violin 3 cellos
Horror		80	Pipe organ Fem. voices Male voices Violin 2 cellos

The first reaction was confusion. After realizing the dynamic of the changes, the next status was surprise followed by a mix of horror and mistrust. One explanation of this last could be a well known human brain bias called Animism. It is the belief that objects possess expressions of intelligent life [42]. It is important to note that the session was really short (23 seconds), but several emotions were detected and the prototype had to react quickly: Neutral (0:00’), Happy(0:06’), Sad(0:09’), Horror (0:13’), Surprise (0:17’), and Horror (0:21’). Angry and Disgusting were not detected. The test

showed to be faster than expected and the prototype worked with some latency. This is one of the future improvements.

V. CONCLUSIONS AND FUTURE WORK

Although the prototype at its current stage lacks many of the proposed features, the results show that human reaction to the artificial creations could be surprise, emotion and involvement. This could be interpreted as a result of one of the natural biases of the human brain: animism, which assigns human abilities to objects or animals. Among the pending tasks it remains the extension of the set of features. From the artistic perspective: to add sounds and more alterations to the musical pieces. Hardware will be reconsidered in order to improve response time, probably adding co-processors or other dedicated hardware. Statistically, the first step is to test with two observers, and an extensive test. The authors aim to use them to know more about the interaction between art, creativity and emotions, and to determine the psychological effects (like animism manifestations), and to advance in new approaches on artificial creativity.

ACKNOWLEDGMENT

The authors want to recognize the volunteer collaboration of musician S. I. Marcovecchio, from the National Manuel de Falla Superior Conservatory.

REFERENCES

- [1] J. McCormack, "SensiLab". Cornell University. ARXIV. <https://sensilab.monash.edu/research/art-or-not/>. 2022
- [2] M. du Sautoy. "True AI creativity is coming and will reveal the minds of machines". Monash University. <https://www.newscientist.com/>. 2022
- [3] B. Dickson, "Deep Learning is not a Replacement for Human Creativity Period". In TechTalks. <https://bdtechtalks.com/>. 2022
- [4] N. Bourriaud, "Relational Aesthetics". Les presses du réel. 2020.
- [5] E. Kac. "Aspects of the Aesthetics of Telecommunications". International Conference on Computer Graphics and Interactive Techniques: ACM SIGGRAPH 92 Visual Proceedings. 1992.
- [6] S. Penny. "Agents as artworks and agent design as artistic practice". Human cognition and social agent technology. 2000.
- [7] S. Penny, "Systems Aesthetics and Cyborg Art: The Legacy of Jack Burnham". Sculpture. 1999.
- [8] Data.ai. <https://www.data.ai/es/apps/ios/app/3d-video-cam-turn-your-media-into-3d/>. 2022
- [9] J. Xie, R.B. Girshick, and A. Farhadi, "Deep3D: Fully Automatic 2D-to-3D Video Conversion with Deep Conv. N Networks". In ECCV. 2016.
- [10] S. Yao, L. Wang, D. Li and M. Zhang, "A Real-Time Full HD 2D-to-3D Video Conversion System Based on FPGA". 7th ICIG & Graph. 2013
- [11] Y. Lv et al., "Research on microsystem interposer designer software with through silicon". In 13th Int. Conf. on Electronic Packaging Tech.. 2012
- [12] G. Guo, N. Zhang, L. Huo, and W. Gao, "2D to 3D conversion based on edge defocus and segmentation". IEEE ICASSP 2008
- [13] B. Mildenhall et al., "Representing Scenes as Neural Radiance Fields for View Synthesis". In proceedings ECCV. 2020
- [14] T. Komatsu, and T. Saito, "Color Transformation and Interpolation for Direct Color Imaging with a Color Filter Array". ICASSP. 2006
- [15] M. S. Imtiaz and K. A. Wahid, "Image enhancement and space-variant color reproduction method ". In 36th EMBC, pp. 3905-3908. 2014
- [16] Y. Wen, M. Ng and Y. Huang, "Efficient Total Variation Minimization Methods for Color Image Restoration". Trans. on Image Proc. 2008
- [17] G. Demarcq, L. Mascarilla, and P. Courtellemont, "The Color Monogenic Signal: framework for color processing". In 16th ICIP. 2009
- [18] J. Lin, T. Horiuchi, K. Hirai, and S. Tominaga, "Color image recovery system from printed gray image". SSIAP, pp. 41-44. 2014
- [19] Y. Wang, X. Shi, F. Li, and Y. Ye, "An effective color image fusion algorithm of visual and infrared night images". In 26th CCDC. 2014
- [20] T. Li and H. Zhu, "Research on Color Algorithm of Gray Image Based on a Color Channel". CCDC. 2020
- [21] M. S. Imtiaz, T. H. Khan, and K. Wahid, "New color image enhancement method for endoscopic images". In 2nd ICAEE. 2013
- [22] X. Luo, J. Huang, R. Szeliski, K. Matzen, and J. Kopf, "DeOldify: open photo-retoucher based on machine learning". Tech4data feeds. 2018
- [23] B. Kamolrat, W. A. C. Fernando, and M. Mrak, "Adaptive motion-estimation-mode selection for depth video coding". IEEE ICASSP. 2010
- [24] T. Kuo, C. Hsieh, and Y. Lo, "Depth map estimation from a single video sequence". IEEE Int. Symposium on Consumer Electronics. 2013
- [25] B. Wang, J. Zou, Y. Li, K. Ju, H. Xiong, and Y. F. Zheng, "Sparse-to-Dense Depth Estimation in Videos". Trans. on Circ. & Sys.. 2019
- [26] S. Lee, and Y. Ho, "View-consistent multi-view depth estimation for three-dimensional video generation" In 3DTV-Conf., pp. 1-4. 2010
- [27] J. Zhang et al., "A Unified Scheme for Super-Resolution and Depth Estimation". Trans. on Circ. & Sys. for Video Tech. 2016
- [28] Q. Zhang, P. An, Y. Zhang, and Z. Zhang, "Improved multi-view depth estimation for view synthesis in 3D video coding". In 3DTV Conf., 2011
- [29] Y. Fan, S. Wu, and B. Lin, "3D Depth Map Motion Estimation & Compensation". In IEEE Trans. on Magnetics, v47, n3. 2011
- [30] A. A. Alatan, and L. Onural, "Estimation of depth fields suitable for video compression based on 3-D structure and motion of objects". In IEEE Trans. on Image Processing, v7, n6, pp. 904-908. 1998
- [31] J. Seo, D. Park, H. C. Wey, S. Lee, and K. Sohn, "Motion information sharing mode for depth video coding". In 3DTV-Conference. 2010
- [32] K. Vougioukas, P. Ma, S. Petridis, M. Pantic, "Video-Driven Speech Reconstruction using GAN". 2019
- [33] E. Buchanan et al., "Speechreading Ability in Both Deaf and Hearing Children". J. Speech Lang Hear Res. 13;63(11). 2020
- [34] L.N. Oliveira, A.D. Soares, B.M. Chiari, "Speechreading as a communication mediator". Codas;26(1):53-60. PMID: 24714859. 2014
- [35] P. Arnold, "The structure and optimization of speechreading". Journal of Deaf Stud Deaf Educ. Fall;2(4):199-211. 1997
- [36] A. Ephrat, T. Halperin, S. Peleg, "Improved Speech Reconstruction from Silent Video". arXiv:1708.01204v3 [cs.CV] 2017
- [37] Y. Yorozu et al., "Electron spectroscopy studies on magneto-optical media and plastic substrate interface" IEEE Trans. V 2. 1987
- [38] W. Hsu et al., "Knowledge discovery over community-sharing media: From signal to intelligence". Int. Conf. on Multimedia and Expo. 2009
- [39] Y. Shen et al. "InterFaceGAN: Interpreting the Disentangled Face Representation Learned". Trans. Pattern Anal. & MI. 2022
- [40] M. Young, "DeepFace: A Lightweight Face Recognition and Facial Attribute Analysis Framework". Tech.I Writer's Handbook. 1989.
- [41] Li, Yanghao, et al. "Demystifying neural style transfer." arXiv preprint arXiv:1701.01036. 2017.
- [42] T. Northcote Whitbridge, "Animismo", Encyclopædia Britannica.

Síntesis de espectrogramas de sonidos subacuáticos con Autoencoders y Transfer Learning

María C. Cebedio y Marco Carnaghi

ICYTE, Depto. de Electrónica y Computación, Facultad de Ingeniería - UNMDP

Mar del Plata, 7600, Argentina

{mcarnaghi,celestecebedio}@fi.mdp.edu.ar

Resumen—En este trabajo se presenta la síntesis de espectrogramas de sonidos subacuáticos de baja frecuencia, a partir de aplicar el método de aprendizaje por transferencia (Transfer Learning) en Autoencoder Convolucionales simples.

El estudio abarca el acondicionamiento de los datos, la utilización de métodos de Transfer Learning con modelos pre-entrenados sencillos, el análisis de la mejor opción y la generación de los espectrogramas ficticios. La síntesis final se realiza a partir de vectores de baja dimensionalidad generados con distribución gaussiana.

Los resultados obtenidos demuestran que, a partir de pocos datos de muestra, con arquitectura de baja complejidad y asumiendo una distribución normal de los vectores reales, es posible sintetizar espectrogramas de sonidos de baja frecuencia subacuáticos.

Palabras Claves—Autoencoders convolucionales, espectrogramas, Transfer Learning, sonidos subacuáticos, síntesis.

I. INTRODUCCIÓN

Las características del medio submarino hacen ideal la transmisión acústica y, por este motivo, el estudio del entorno y todo lo concerniente a él se basa en señales de audio. Una gran cantidad de trabajos en el área requieren acceso a registros reales de sonidos subacuáticos para sus análisis, pero las pruebas en campo son costosas, requieren de infraestructura, suelen llevar un tiempo considerable y, en muchos casos, dependen de factores externos [1]. Por estos motivos, la ausencia del volumen requerido de datos de audio submarino es un problema habitual.

En consecuencia, la posibilidad de generar datos artificialmente podría ser de utilidad para los primeros pasos de una investigación, entrenamiento de redes neuronales, simulación de entornos marinos, etc. Asimismo, si estos datos ficticios pueden ser generados en tiempo real por dispositivos de baja potencia computacional, podrían generarse bancos de prueba artificiales, lo cual sería muy beneficioso para el estudio del medio.

En la actualidad se han diseñado modelos de aprendizaje profundo (ML) generativos basados en técnicas de aprendizaje profundo no supervisado, como el Autoencoder Convolutivo (AE-CNN) y el Autoencoder Variacional (VAE) para reconstruir señales [2]. Sin embargo, estos modelos requieren disponer de grandes volúmenes de datos, amplia capacidad de cómputo y largos períodos de tiempo de entrenamiento. Debido a esto, el Transfer Learning (TL) ha surgido como una técnica atractiva y se emplea con el objetivo de reutilizar un modelo previamente entrenado, como punto de partida en una nueva tarea o aplicación [3].

La aplicación de metodologías de TL incrementa el grado de generalización de los modelos de ML, permitiendo la utilización de conjuntos de datos de menor tamaño y mayor simplicidad (baja dimensionalidad). Además, posibilita una aceleración en el entrenamiento, permite representaciones más robustas y de amplia aplicación [4].

Existen modelos pre-entrenados, que han demostrado ser útiles para realizar TL en la etapa de extracción de características. El problema de estos modelos radica en la complejidad de los mismos. Los modelos típicamente utilizados para realizar TL poseen una cantidad de capas profundas alta y, por ende, la cantidad de parámetros asociados también. A modo de ejemplo, en el caso de AlexNet presenta 62,3 millones de parámetros [5], lo que implica igual orden de operaciones aritméticas y necesidad de trabajar en paralelo con GPU's. Por ende, la complejidad asociada a cualquier implementación de este modelo en un sistema embebido es muy grande. De la misma manera Densenet201 [6] presenta 201 capas convolucionales profundas.

En este contexto, el objetivo del presente trabajo radica en la generación de espectrogramas ficticios partiendo de un escaso volumen de datos de entrenamiento y aplicando técnicas de TL [7] con arquitecturas de Autoencoders de baja complejidad a nivel de operaciones y de pocas capas convolucionales. La elección de la arquitectura y la evaluación de un modelo sencillo, es el punto de partida para la posible implementación en un sistemas embebido que posea bajo potencia computacional.

II. METODOLOGÍA

La metodología de síntesis consiste en: Adecuación de los datos de entrada, Selección de modelos pre-entrenados con set de datos correspondiente a ballenas barbadas, Utilizar técnicas de TL sobre arquitecturas pre-entrenadas, evaluación de resultados y obtención del modelo, Utilizar técnicas de TL a conjuntos de datos minoritarios correspondientes sonidos subacuáticos y generación de espectrogramas ficticios.

II-A. Adecuación de los datos de entrada

El objetivo de esta etapa es obtener una matriz bidimensional que represente el espectrograma de magnitud logarítmica para cada registro de audio. El conjunto de todos los espectrogramas normalizados entre 0 y 1 para ser interpretados como una imagen en escala de grises, se

agrupa en un tensor. Las características de los registros de audio se presentan en [8].

Los datos recolectados corresponden a audios de diferentes sonidos de baja frecuencia presentes en el entorno submarino. De esta manera, se tienen registros agrupados en diferentes clases: sub-especies de Ballenas Barbadas, Peces y Ruidos de mar. En la Tabla I se resumen la cantidad de datos disponibles y la forma del tensor de entrada para cada categoría.

DATOS	N	Dimensión del tensor de entrada	
		Datos Train	Datos Test
B. Barbadas (todas)	6715	[6043,128,196,1]	[672,128,196,1]
B. Azul	163	[146,128,196,1]	[17,128,196,1]
B. Gris	55	[49,128,196,1]	[6,128,196,1]
B. Rorcual	2884	[2595,128,196,1]	[289,128,196,1]
B Groenlandia	434	[393,128,196,1]	[44,128,196,1]
B. Minke	122	[109,128,196,1]	[13,128,196,1]
B. Jorobada	2879	[2591,128,196,1]	[288,128,196,1]
B. Franca	147	[132,128,196,1]	[15,128,196,1]
Peces	313	[281,128,196,1]	[32,128,196,1]
Ruido de Mar	1625	[1462,128,196,1]	[163,128,196,1]

Tabla I: Datos disponibles para el entrenamiento de los modelos: cantidad y forma del tensor de entrada

II-B. Modelos pre-entrenados.

Los modelos de AE-CNN y VAE utilizados durante el pre-entrenamiento cuentan con 4 capas en la etapa de extracción de características. En todas ellas, la función de activación es tipo RELU, el Padding se re-acomoda de forma automática y se utiliza la operación de Batch Normalization [9].

El modelo AE-CNN, tiene una sola capa densa en la sección de capa profunda y para VAE dicha capa está representada por capas densamente conectadas de varianza y media. Estos modelos entrenados y las características del entrenamiento con los datos mayoritarios correspondientes a la categoría “Ballenas Barbadas” se presentan en [8].

La dimensión del espacio latente que presenta mejor performance de cara a su utilización como modelos pre-entrenados surge de una análisis previo realizado en [10]. En dicho análisis se comparan arquitecturas de 1, 2, 3 y 4 dimensiones para las dos arquitecturas, pero entrenadas con el conjunto de todas las ballenas barbadas, obteniéndose un error de entrenamiento menor con D=4 para AE-CNN y D=3 para VAE.

II-C. Transfer Learning

Esta técnica permite sacar provecho de la experticia ganada en las capas convoluciones de modelos entrenados con una gran cantidad de datos y utilizarlas para extraer características de otras conjunto minoritario de datos.

El concepto principal radica en congelar, es decir, fijar los pesos de ciertas capas durante el entrenamiento y hacer un ajuste fino de los pesos restantes para responder al nuevo problema. Esta estrategia permite reutilizar los conocimientos en términos de la arquitectura global de la red y explotar sus estados como punto de partida para el entrenamiento [11]. En este caso en particular, se congelan los pesos de las capas convolucionales y se re-entrenan las capas densamente conectadas. En la Fig. 1 se muestra el modelo AE-CNN y las capas que se “congelan”, ajustando los pesos únicamente de las capas densamente conectadas.

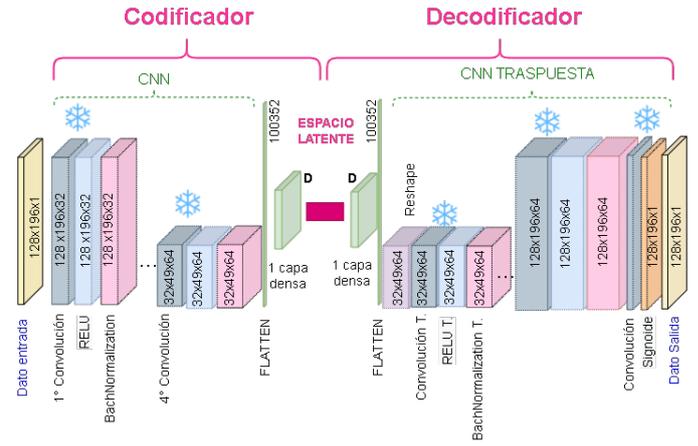


Fig. 1: Arquitectura Autoencoder CNN para TL. Las capas convolucionales marcadas con el símbolos son aquellas que se “congelan”.

III. ANÁLISIS DE LOS RESULTADOS

El resultado de implementar cada arquitectura, muestra una cantidad de parámetros entrenables y otros fijos, que dan un noción de la dimensión del modelo, la complejidad de las operaciones involucradas y los tiempos de entrenamiento requeridos. Los parámetros resultan de aplicar TL a CNN son: 1.137.093, de los cuales 932.805 son entrenables y los parámetros resultan de aplicar TL a VAE son: 1.209.223, de los cuales 1.004.358 son entrenables.

III-A. Re-entrenamiento para subespecies de ballenas

En este punto, se re-entrenan los modelos seleccionados con distintos sub-clases de ballenas, se calcula el error cuadrático medio (MSE) y el error de similitud estructural (SSIM). Para realizar comparativa de la mejora obtenida al utilizar TL, se entrenan los modelos AE y VAE, sin haber realizado un pre-entrenamiento. A modo de resumen, en la tabla II, se presentan los resultados obtenidos. Del análisis de estos datos presentados, resulta que aplicar TL a la arquitectura AE-CNN pre-entrenada con el conjunto de todas las ballenas Barbadas, posee el mejor desempeño.

Datos de TEST	N	CNN		TL CNN		VAE		TL VAE	
		MSE	SSIM	MSE	SSIM	MSE	SSIM	MSE	SSIM
B. Azul	146	0.026	0.590	0.008	0.64	0.024	0.598	0.053	0.499
B. Minke	109	0.068	0.345	0.009	0.553	0.011	0.519	0.013	0.482
B. Jorobada	2591	0.026	0.392	0.007	0.452	0.052	0.365	0.017	0.397
B. Franca	132	0.066	0.413	0.016	0.478	0.050	0.433	0.017	0.322
B. Rorcual	2595	0.006	0.460	0.006	0.463	0.029	0.324	0.061	0.139
B. Gris	49	0.038	0.412	0.005	0.579	0.059	0.384	0.011	0.49
B. Groenlandia	393	0.024	0.546	0.003	0.629	0.03	0.535	0.018	0.507

Tabla II: Resumen de métricas obtenidas para las diferentes arquitecturas, con y sin Transfer Learning (TL)

En la Fig. 2, se muestra un espectrograma correspondiente a la categoría “Ballena Groenlandia” y las distintas reconstrucciones realizadas. Se observa que la reconstrucción obtenida con la arquitecturas CNN y VAE, sin aplicar TL, son muy desalentadoras. Esto es de esperarse ya que la cantidad de datos de esta subclase es considerablemente reducida. Sin embargo, al realizar TL se aprecia una mejora del desempeño.

III-B. Extensión a otras fuentes de sonidos

El modelo AE-CNN pre-entrenado [8] se utiliza para realizar TL, con el fin de adecuarlo a otros sonidos subacuáticos. En este caso en concreto, la técnica es evaluada en base a sonidos provenientes de peces y ruido de mar.

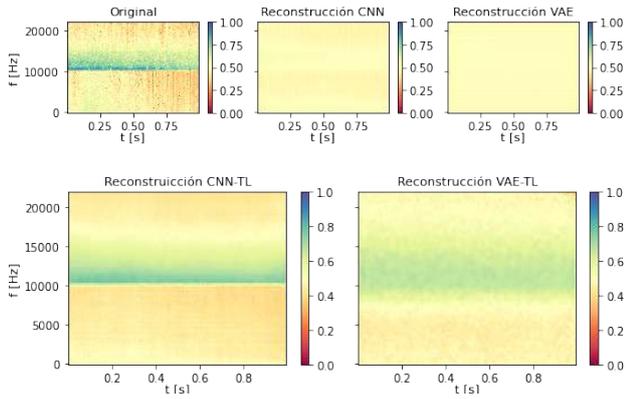


Fig. 2: Espectrograma Ballena Groenlandia y sus diferentes reconstrucciones. La cantidad de datos de entrenamiento es $N_{train} = 393$.

La Fig. 3 muestra las reconstrucciones obtenidas para estos registros. Se observa que el modelo pre-entrenado aporta una etapa de extracción de características de utilidad en ambos casos y las reconstrucciones obtenidas recuperan las propiedades esenciales de ambas fuentes sonoras.

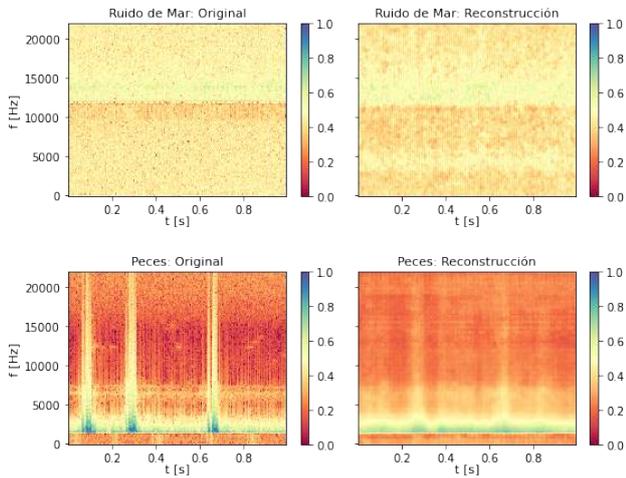


Fig. 3: Espectrogramas de sonidos subacuáticos: Espectrograma Original vs Reconstrucción realizada con TL aplicado a AE-CNN.

IV. GENERACIÓN DE ESPECTROGRAMAS FICTICIOS

Para la generación de datos ficticios se utiliza sólo el bloque decodificador del Autoencoder CNN y se ingresa a éste (según sea el espectrograma a generar), con un vector de dimensión 4.

Se asume que la distribución de cada valor del espacio latente es Gaussiana y, a partir de todos los vectores de espacio latente generados con los datos de Train, se obtiene un valor medio y una desviación. Estos valores actúan como punto de partida para generar vectores de códigos aleatorios de dimensión 4 a ingresar en el bloque decodificador entrenado.

A modo de ejemplo, la Fig. 4 muestra imágenes sintetizadas y la Fig. 5 imágenes reales, para ballenas “Groenlandia”.

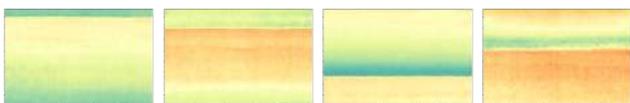


Fig. 4: Espectrogramas sintetizados, obtenidos con un decodificador entrenado, a partir de registros de ballenas Groenlandia.

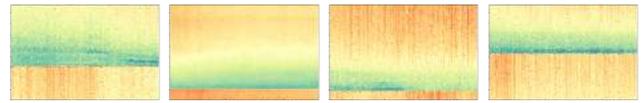


Fig. 5: Espectrogramas Reales, obtenidos a partir de registros aleatorios de ballenas Groenlandia.

V. CONCLUSIÓN Y TRABAJO A FUTURO

En este trabajo se logra la generación de espectrogramas ficticios, correspondientes a fuentes sonoras de baja frecuencia del entorno submarino. La propuesta de utilizar técnicas de TL, dado el marcado desbalance en los datos disponibles, muestra buenos resultados.

El modelo presentado posee un reducido número de capas, las operaciones involucradas son simples y la cantidad de parámetros a implementar son bajos, en comparación con las arquitecturas de código abierto usualmente utilizadas. Esta particularidad le permite un entrenamiento en una PC personal y una posible implementación en un sistema embebido de baja potencia computacional, donde las operaciones aritméticas y errores asociados a estas, sumado a la cantidad de parámetros son un punto importante a tener en cuenta. A medida que las redes se vuelven más profundas y complejas, el costo de una implementación aumenta.

Las innovadoras herramientas de código abierto, tales como Tensor Flow Lite [12] o aquellas que permiten la utilización de código python sobre FPGA, facilitan la implementación de este tipo de redes de baja complejidad.

VI. AGRADECIMIENTOS

Al Dr. Diego Comas y al Dr. Gustavo Meshino por los conocimientos impartidos sobre la temática.

REFERENCIAS

- [1] E. Tejero, “Aplicaciones de Machine Learning a la Bioacústica Marina,” Ph.D. dissertation, 07 2020.
- [2] Q. Xu, Z. Wu, Y. Yang, and L. Zhang, “The difference learning of hidden layer between autoencoder and variational autoencoder,” in *29th Chinese Control And Decision Conference*, 2017, pp. 4801–4804.
- [3] D. Sarkar, R. Bali, and T. Ghosh, *Hands-On Transfer Learning with Python*, 1st ed. Packt, 1995, ISBN 13: 9781788831307.
- [4] L. Torrey and J. Shavlik, *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, 1st ed. IGI Global, Hershey, 2010, DOI:10.4018/978-1-60566-766-9.ch011.
- [5] J. Wei, “AlexNet: The Architecture that Challenged CNNs,” *Towards Data Science*, 2019. [Online]. Available: <https://acortar.link/IrMULc>(acceso:25dejuniode2022).
- [6] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269.
- [7] E. Tsalera, A. Papadakis, and M. Samarakou, “Comparison of Pre-Trained CNNs for Audio Classification Using Transfer Learning,” *Journal of Sensor and Actuator Networks*, vol. 10, p. 72, 12 2021.
- [8] M. C. Cebedio and M. Carnaghi, “Repositorio-CASE2022,” GitHub, 2022. [Online]. Available: <https://github.com/Reposinnombre/CASE2022>
- [9] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” 2015. [Online]. Available: <https://arxiv.org/abs/1502.03167>
- [10] M. C. Cebedio and M. Carnaghi, “Espectrogramas de registros de Ballenas Barbadas, sintetizados a partir de Autoencoders,” 2022, en evaluación.
- [11] A. Yoss, “Transfer Learning using Pre-Trained AlexNet Model and Fashion-MNIST,” *Towards Data Science*, 2020. [Online]. Available: <https://acortar.link/IBrXLM>(acceso:25dejuniode2022).
- [12] “Tensorflow lite,” TensorFlow. [Online]. Available: <https://www.tensorflow.org/lite/guide?hl=es-419>(acceso:23dejuniode2022)

Exploring design choices in Image Classification Convolutional Neural Networks

Nicolás E. Cecchi
Facultad de Ingeniería
Universidad de Buenos Aires
 nicolas.cecchi@ing.unlp.edu.ar

Javier Kreiner
Almafintech
Universidad de Buenos Aires
 javkrei@gmail.com

Abstract—Hyper-parameter setting is crucial to the parameter convergence and performance of artificial neural networks. Doing this task properly usually requires highly-skilled professionals with experience in the field of interest or in AI, or a great amount of computational power. Not having access to these, small organizations and individuals tend to draw upon configurations published in research papers and/or default settings in AI-frameworks. In this paper we discuss an heuristic methodology that relies on early-stage evaluation metrics, which allows developers to explore other possibilities. We explore a variety of architectures coupled with a cyclical learning rate schedule and two optimizers. The tests are performed on CIFAR10, CIFAR100 and EuroSAT datasets. The results are promising, allowing us to select configurations with high convergence ratios and expected performance. From the 126 first-stage experiments performed, 37 showed early-stage signs of convergence and the 10 selected for extended training (50 epochs) averaged 74.49% top 1 categorical accuracy on the 10 classes datasets.

Index Terms—hyper-parameters, convolutional neural networks, classification, optimization

I. INTRODUCTION

Design choices and hyper-parameter settings are crucial to the parameter convergence and performance of neural networks, yet there is no clear theoretical framework that allow professionals to make educated decisions in this domain [1]. This turns out to be particularly troublesome for inexperienced individuals or small organizations that do not have the knowledge nor the computational resources required to use the available techniques. This project aims to simplify the evaluation of some design choices in image classification problems with convolutional neural networks (CNN) through the validation of a simple methodology for hyper-parameter selection. In particular, experiments are run with varying architectures, learning rate schedules and optimizers on 3 different datasets.

Regarding the architecture choice, Frankle and Carbin [2] present the "lottery ticket hypothesis" which, through an iterative pruning procedure, succeeds in finding a sub-architecture that performs similar to the parent one with a significantly smaller computational cost. Nonetheless, this method still requires human input for the architecture before pruning and significant computational power for that first iteration. Elksen, Metzen and Hutter [3] explore the field of neural architecture search, a branch of AutoML that focuses on the architecture of networks. This technique relies on human-set searching

space and strategy. Another approach is taken by Baldominos, Saez and Isasi [4], whose work emphasizes neuroevolution design on CNN and genetic algorithms are applied varying the number of layers, type of layers, activation functions, etc. Finally, Smith and Topin [5] review the literature on the topic and summarize 14 design patterns commonly found in well-performing projects.

Regarding the learning rate (LR), some researchers advocate for a dynamic learning rate over an static one [6]. Bengio [7] also emphasizes the need for a learning rate schedule for gradient based optimization. Smith [8] focuses on cyclical learning rate (CLR) schedules and concludes that with a triangular schedule (linear increasing and decreasing) the best complexity-performance trade-off is achieved.

As seen, most of the existing techniques require expert knowledge or plenty of resources. We try to tackle this by distilling a simple and efficient approach for this task.

In Section II we introduce the methodology and scope of the work, describing in depth the variations of our experiments. In Section III we present and discuss the results obtained. Finally, in Section IV, we conclude and suggest possible lines of future work.

II. METHODOLOGY AND SCOPE

In this paper we analyze an heuristic that allows practitioners to test CNNs expected parameter convergence and performance. We run experiments with variations in the architecture, the learning rate policy and the optimizer in CNN for image-classification over 3 datasets.

A. Datasets

For testing the methodology, we worked with CIFAR10, CIFAR100 [9] and EuroSAT [10] datasets. CIFAR10 and CIFAR100 are composed of 32x32 RGB images with 10 and 100 classes, respectively. They both have 50 000 training samples and 10 000 validation samples with balanced classes. With respect to EuroSAT, it contains a unique set of 27 000 samples of 64x64 RGB images and 10 balanced classes. This dataset is split into train and validation sets, with 80% and 20% of the images, respectively.

A standard preprocessing is used, consisting in a one hot

encoding of the labels together with a re-scaling and centering of the pixel intensities of the 3 channels, as following:

$$f : \{0, 1, \dots, 255\} \rightarrow [-0.5, 0.5]$$

$$f(p) = \frac{p}{255} - 0.5 \quad (1)$$

B. Architecture

For architecture design we work with the concept of a "block", defined as a group of two consecutive convolutional layers followed by a pooling layer. Convolutional layers of the same block have the exact same configuration, while the only inter-block difference is the number of filters.

- Activation function: ReLU
- Kernel shape: (3,3)
- Stride: (1,1)
- Padding: 'same'
- Weight initializer: GlorotUniform
- Regularizer: L2
- Filters: $F_0 \cdot 2^n$

Where $n \in \{0, 1, 2, 3\}$ denotes the block-number, according to the number of blocks of the model being tested. F_0 is the number of filters in the convolutional layers of the first block, and can take values in $\{4, 8, 16, 32\}$ for CIFAR datasets and $\{2, 4, 8, 16\}$ for EuroSAT.

Pooling layers will always have the same configuration, regardless of any other hyper-parameter or feature being tested. This configuration is the following:

- Pool size: (2,2)
- Stride: (2,2)
- Operation: Max

At the end of every architecture 3 fully-connected layers are added: the first 2 of them having as many nodes as the size of flattening the output of the final block and the last one, being the output layer, 10 or 100 nodes (depending on the dataset) with soft-max activation for classification purposes.

These decisions were made taking into account some of the design patterns described by Smith and Topin [5], in particular: "architectural structure follows the application", by which CNN are used in the context of their canonical application: image classification; "strive for simplicity", that suggests keeping the network as simple as possible and serves our purpose of helping non-expert practitioners; "pyramid shape", which says there should be an overall smooth down-sampling combined with an increase in the number of channels throughout the architecture; "input transition", that remarks the common practice of increasing the number of channels from the input, which increases the number of parameters in the network; and "available resources guide layer widths", for the trade-off between model complexity and computational cost.

C. Optimizer

For being among the most popular options in the literature, two gradient descent [12] variations were used in the experiments as optimizers: mini-batch gradient descent and Adam [13]. Default configurations from the Keras API and batch sizes set to $b = 100$ were used in every case.

D. Learning rate

"Triangular2" [8] [11] learning rate policy is used in every experiment. This policy consists of cycles of linear increments and reductions of the learning rate between a minimum and a maximum boundary, the latter being halved on every cycle, as in Figure 1, where the learning rate is shown from its starting point ("0th epoch") until the end of the 8th epoch.

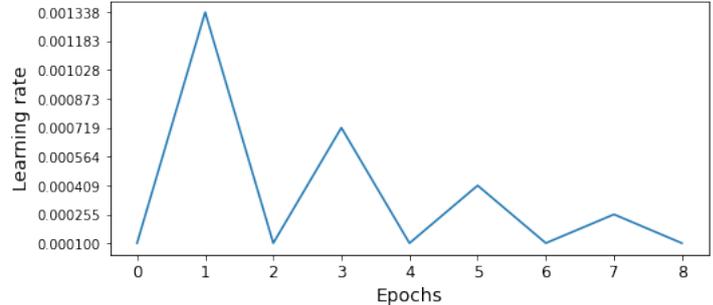


Fig. 1. "Triangular2" learning rate policy.

To determine these boundaries we make use of the "learning rate range test" (LRRT) [8]. In this test, a CNN is trained for a small number of epochs (8, in our case) with the learning rate growing linearly. Afterwards, analyzing the categorical accuracy (CA) evolution, we look for two regions: a first one with high slope and another one where the growth stalls. The lower limit for the learning rate is to be chosen from the former and the upper limit from the latter, as demonstrated in subsection II-E.

Experiments were performed with the learning rate in the intervals $R_1 : [0.0001; 0.01]$ and $R_2 : [0.01; 0.1]$. These ranges were suggested by Smith [11]. Regarding the length of a cycle, since Smith recommends to run the test for 4 cycles, we calculate a length of 6250 batches-per-cycle in CIFAR datasets and 1350 batches-per-cycle in EuroSAT.

E. Experiments

Each experiment consisted in selecting an element from the set of possible combinations $C = \{\mathcal{D}_i, \mathcal{R}_j, \mathcal{O}_k, \mathcal{A}_l\}$, that describe the previously explained variations, where \mathcal{D}_i is a dataset, \mathcal{R}_j is a range for the LRRT, \mathcal{O}_k is an optimizer and \mathcal{A}_l an architecture, defined by a pair (nblocks, F_0).

The LRRT is executed 5 times for each combination C_{ijkl} , varying the random seed fed to the initializer. To analyze the results, the median of the 5 runs is considered. If the experiment does not show signs of convergence, the configuration is discarded. Otherwise, configurations are grouped by $\{\mathcal{D}_i, \mathcal{R}_j\}$ and compared. For each group, the top performing pair $\{\mathcal{O}_k, \mathcal{A}_l\}$ is selected to make an extended training of 50 epochs with the triangular2 learning rate policy with the limits identified by the LRRT.

As an example, Figure 2 shows the CA for 3 different configurations in the group $\{\text{EuroSAT}, [0.0001, 0.01]\}$. The orange configuration shows no signs of learning, so it is discarded. Both the blue and green curves show converging

behaviour, since the green CA is higher, this configuration should be chosen. For the LR limits, the first region would be between the beginning of the training and the 3rd epoch ($LR \in [0.0001; 0.0038125]$), with the stalled region corresponding to $LR > 0.0038125$.

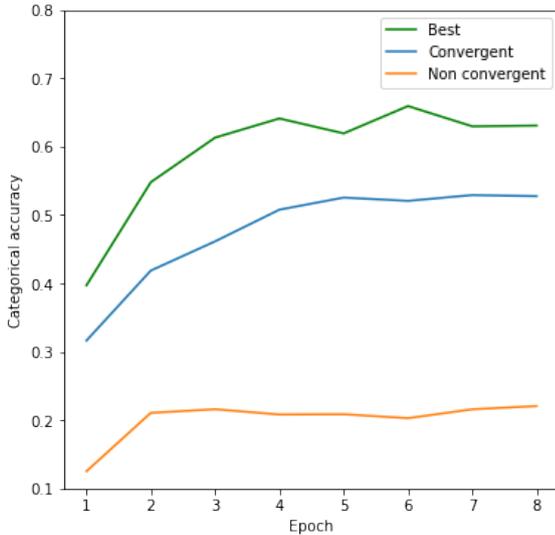


Fig. 2. CA for LRRT with EuroSAT in the [0.0001 - 0.01] range.

III. RESULTS

Table I shows the results of the first stage of the experiments. Accounting for all the different variations, a total of 48 experiments should have been run with each dataset. Nonetheless, given that experiments with CIFAR100 showed no sign of convergence, not all the scheduled experiments were executed, as can be seen in "Executed experiments". The "convergence rate" is defined as the proportion of convergent configurations to the number of performed experiments.

TABLE I
FIRST STAGE RESULTS

Dataset	Executed experiments	Convergence rate
CIFAR10	48	31.25 %
CIFAR100	30	0 %
EuroSAT	48	54.17 %

Table II summarizes the results of extended training with EuroSAT selected models. It is worth noting that it was not necessarily the most complex model the one which performed the best, even between varying configurations.

TABLE II
EXTENDED TRAINING ON EUROSAT

Blocks	F_0	Range	Optimizer	CA (%)
2	8	[0.0001; 0.0038125]	Adam	79.74 %
2	8	[0.01; 0.04]	SGD	78.07 %
3	8	[0.0001; 0.0045]	Adam	83.59 %
3	8	[0.01; 0.1]	SGD	71.66 %
4	16	[0.0001; 0.0015]	Adam	77.68 %
4	8	[0.01; 0.07]	SGD	79.24 %

CIFAR10's results are shown in table III. Since none of the 4-block experiments showed signs of convergence, no extended training was executed.

TABLE III
EXTENDED TRAINING ON CIFAR10

Blocks	F_0	Range	Optimizer	CA (%)
2	32	[0.0001; 0.0013375]	Adam	73.82 %
2	32	[0.01; 0.047125]	SGD	67.32 %
3	32	[0.0001; 0.0013375]	Adam	74.57 %
3	32	[0.01; 0.047125]	SGD	66.06 %
4	N/A	N/A	N/A	N/A
4	N/A	N/A	N/A	N/A

IV. CONCLUSIONS

The studied heuristic methodology allowed us to find configurations for CNN with both high convergence rate and expected performance, making use of an inspection of the metric of interest on early-stage training with the LRRT to configure an adequate learning rate schedule. At the same time, it allows one to discard configurations that show no signs of convergence. These results validate and extend the CLR schedule and LRRT, all this is achieved without the need for powerful hardware or the use of more sophisticated techniques, yet some expert knowledge was needed for setting LRRT ranges. Performance-wise these results call for further investigations on hyper-parameter fine tuning and extending the number of training epochs for the converging configurations. It would also be valuable to extend this work to datasets with bigger images, as well as other types of problems. Finally, the tested methodology required us to select the definitive boundaries for the CLR as well as the CNN configuration, tasks that may be automated.

REFERENCES

- [1] L.N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay", 2018, arXiv: 1803.09820.
- [2] J. Frankle and M. Carbin, "The lottery ticket hypothesis: training pruned neural networks", 2018, arXiv: 1803.03635.
- [3] T. Elsken, J. H. Metzen and F. Hutter, "Neural architecture search: A survey", 2018, arXiv:1808.05377v3.
- [4] A. Baldominos, Y. Saez and P. Isasi, "On the automated, evolutionary design of neural networks: Past, Present and Future", Neural Computing and Applications, 32.2, pp 519-545, 2020.
- [5] L. N. Smith and N. Topin, "Deep Convolutional Neural Network Design Patterns", 2016, arXiv:1611.00847v3.
- [6] "Neural Networks: Tricks of the Trade", Springer, 2012, Volumen 7700, ISBN : 978-3-642-35288-1.
- [7] Y. Bengio, "Practical Recommendations for Gradient-based Training of Deep Architectures", 2012, arXiv:1206.5533v2.
- [8] L.N. Smith, "Cyclical learning rates for training neural networks", IEEE WACV, pp. 464-472, 2017.
- [9] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images", 2009.
- [10] P. Helber, B. Bischke, A. Dengel and D. Borth, "EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification", 2017.
- [11] L.N. Smith, "No more pesky learning rate guessing games", IEEE WACV, 2015.
- [12] L. A. Cauchy, "General Method for the Resolution of Systems of Simultaneous Equations", 1847.
- [13] D. P. Kingma y J. Ba, "Adam: A method for stochastic optimization", ICLR, 2015, arXiv:1412.6980v9.

Categoría

FORO TECNOLÓGICO

Internet de las Cosas (Internet of Things, IoT)

Área Temática

IoT Device Network for Remote Monitoring

Hernán Rodríguez
 Departamento de Glaciología,
 Instituto Antártico Argentino.
 Facultad Regional Buenos Aires,
 Universidad Tecnológica Nacional.
 Buenos Aires, Argentina
 hrodriguez@frba.utn.edu.ar

Sebastián Marinsek
 Departamento de Glaciología,
 Instituto Antártico Argentino.
 Facultad Regional Buenos Aires,
 Universidad Tecnológica Nacional.
 Buenos Aires, Argentina
 smarinsek@dna.gov.ar

Abstract— Internet of Things paradigm is considered one of the greatest technological revolutions of recent times. This new concept allowed the development of new and varied control and monitoring systems. In this work, a remote monitoring network that can be applied in various fields of science and industry, such as climatology, agriculture or energy, is introduced. The network has been installed and started up in the Antarctic Peninsula during the last Antarctic Field Campaign to monitor glaciers and other sites of interest.

Keywords—Internet of Things, remote monitoring, wireless sensor network, embedded software

I. INTRODUCTION

Internet of Things (IoT) concept is based on the development of a device network, equipped with embedded software, sensors and wireless connection, which share information from one device to another or with a central server for data storage. Data visualization through any smart device with internet connection is also part of the IoT basis [1]. These features make IoT an ideal structure for autonomous remote monitoring applications in any science field [2]. The number of IoT devices has significantly increased over the last years and it is expected to continue growing. According to Gartner [3] research and advisory firm, between 2020 and 2030 the installed IoT meters will grow from 1.6 billion to 3.2 billion worldwide.

Glaciology Department from Instituto Antártico Argentino (IAA) has got several research programs to monitor glaciers along Antarctic Peninsula (AP). Mass-balance variations in Antarctic glaciers is directly related to global warming [4], so its study is highly relevant for worldwide researchers. Parameters such as air temperature, relative humidity, glacier surface displacement and glacier surface images are of great relevance for an accurate study of the glaciers dynamics throughout the year.

Currently, data collection takes place only during the Antarctic Field Campaigns which require a very important logistical effort involving ships and aircrafts. One of the most studied glaciers by IAA is Bahía del Diablo glacier (GBD), located on Vega Island, northeastern AP, 60km from Marambio Station [5]. A camp is set at Vega Island every summer to carry out all field surveys and monitoring tasks in GBD. In addition to the complexity for activities deployment, it is important to highlight the extreme climate conditions which make scientists activities more difficult.

Several networks for remote monitoring have been implemented in Antarctica. Some of them rely exclusively on satellite communication, whose link is very expensive [6]. There are also networks that implement a direct link via RF [7], involving new problems such as scope, dispersion, scalability and energy consumption.

The current work introduces a remote monitoring network capable of acquiring and transmitting relevant environmental parameters. This is of great interest in order to track all variables in real time from anywhere in the world. The presented IoT device network is applicable to any science field thanks to its versatility and scalability, which allows accessing to data that was previously only gathered in-situ. The network has been deployed and is currently working successfully in AP.

II. MATERIALS & METHODS

Nowadays, there are various communication protocols and network topologies, each of them with its advantages and disadvantages [8]. The designed network especially focuses on low device consumption, deployment in remote and inaccessible locations and the independence of the devices from each other, so failure of one of the devices does not affect the other devices on the network. In order to satisfy these requirements, the developed equipment allows communication via 3g/4g or satellite networks.

These communication networks also have their advantages and disadvantages. The cellphone network allows establishing communications with a large data flow at a very low cost, but it must be located near a 3g/4g cell to have internet access. However, in some remote sites there is no cellular coverage. For those cases, satellite network is ideal due to its worldwide coverage, allowing device installation in places where there is no reach of cellphone network. Nevertheless, satellite links are very expensive, so it is not convenient to send large amounts of data in order to reduce costs [9].

Regardless of the method of communication, all data is finally stored on a FTP server located in Buenos Aires, Argentina. Through this server, the data can be visualized in real time from anywhere in the world through a smart device and internet connection. A website was designed for this purpose.

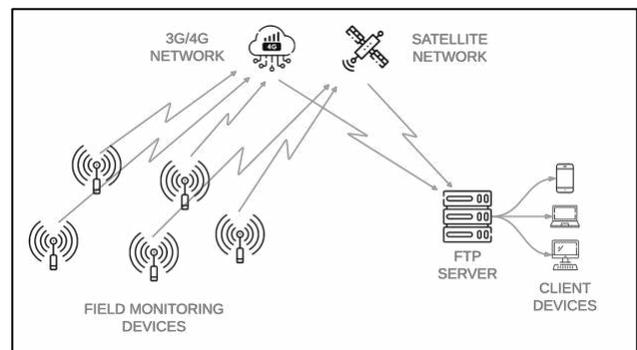


Fig. 1. Scheme corresponding to the connections of the monitoring devices installed in the field and the client devices, passing through the FTP Server.

A. Cellular Communication

EC25 MiniPCIE (Quectel, Shanghai, China) module was used for cellular communication. This is a series of LTE category 4 module adopting standard PCI Express Mini Card form factor (Mini PCIe). It is optimal especially for M2M and IoT applications. These modules are backward-compatible with existing EDGE and GSM/GPRS networks, ensuring that it can be connected even in remote areas devoid of 3g or 4g coverage [10].

B. Satellite Communication

Satellite link was carried out with SBDN9602 (Iridium, Virginia, USA) transceiver. This module implements a Short Burst Data (SBD) interface which offers low service latency and worldwide coverage. Just as EC25, the communication interface is achieved through serial port, which makes both modules compatible and interchangeable.

C. Processing

The core of the developed system is the SoC (System On a Chip) ESP32 (Espressif Systems, Shanghai, China). ESP32 has a great processing capacity and connectivity. It integrates a powerful microcontroller with 32-bit architecture. ESP32 implements an ESP-IDF FreeRTOS based on Vanilla FreeRTOS v10.4.3. This real-time operating system for microcontrollers is open source and freely distributed. It is a robust operating system that supports low power applications. For this project, the ESP32-CAM module is used because it has integrated the OV2640 video camera and a micro-SD slot for back-up data storage.

The software was designed to periodically interact with each sensor and the communication module through independent tasks. This design allows new modules and transducers to easily be attached through new tasks without affecting the modules that are already operational.

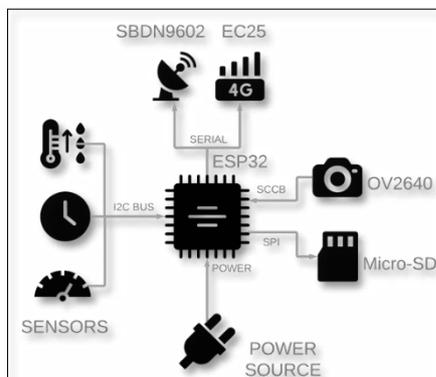


Fig. 2. Scheme corresponding to the connections of the microcontroller and the different peripherals.

III. RESULTS

The proposed device is visualized in Fig. 3. The printed circuit board (PCB) modular design allows using the same board whether cellular or satellite communication is available or not. i2c port is available for various sensors connection. In the particular case installed in AP, air temperature and relative humidity sensors SHT31 or SHT71 (Sensirion AG, Stäfa, Switzerland) and DS3231 (Maxim Integrated, California, USA) real-time clocks were used. An additional serial port can also be used. In the mentioned use case, a GNSS receiver L76 (Quectel, Shanghai, China) was attached to measure glacier surface displacement.

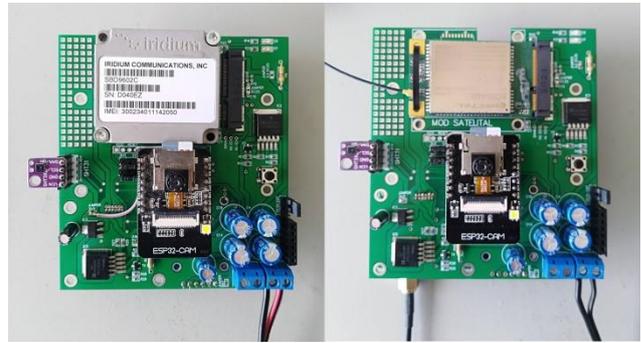


Fig. 3. PCB equipped with satellite communication (left) and with cellular communication (right).

As it can be seen, the board has 2 power inputs, 12 V and 3.3 V. The one typically used is 12 V, since it provides power for cellular and satellite transmission modules. However, it is also possible to supply the circuit with 3.3 V. In this configuration, the measurement device stores the data locally in the micro-SD card. Although this project aim is to achieve remote monitoring, there may be cases in which it is only necessary to store the information for later survey. For these particular cases, this alternative is offered to reduce energizing costs.

To energize the system with 12 V, the necessary level of energy to achieve effective transmissions must be taken into account. Therefore, the energy availability should be considered when sizing the installation. If access to the electrical grid is available, a 12 V – 1 A source will be enough to keep the device operating properly. However, if the installation takes place in an isolated location, data transmission frequency and maintenance tasks periodicity must be taken into account. In AP, maintenance tasks are carried out annually during Antarctic Summer Campaigns. In addition, devices with cellular communication transmit data every half hour, while those with satellite communication do so every hour to reduce cost. In every case, deep-cycle gel batteries (12 V - 26 Ah) are used to power the device. This kind of accumulators can be discharged to 70% or more of the battery capacity and tolerate a greater number of charge and discharge cycles than conventional batteries. Additionally, batteries have solar panels to be recharged.

It should be noted that measurements are taken periodically. Time between measurements will depend on how expensive the communication is in both economic and energetic terms. For the current project, measurements are taken every 30 minutes when 3g/4g network is available and every hour where the satellite link is used. Once the transmission is over, all modules are turned off by ESP32 or start low consumption mode. During rest period, consumption is reduced to approximately 2 mA. If any module fails, the device tries to avoid the error by retrying to establish the link. However, if the fault persists, the equipment goes into low consumption and the cycle starts again in the next measurement.

During 2022 Summer Antarctic Campaign, seven devices were installed in AP. Three of them have access to 3g/4g network due to their proximity to Marambio Island, so they have cellular communication. These devices were installed on Cockburn Island, 10 km from Marambio Station, on Vega Island, 60 km from Marambio Station, and at Marambio Meteorological Station. Devices installed on Cockburn and Vega transmit images. The one at Marambio Station sends temperature and humidity data.

Two devices have satellite communication. They are also installed on Vega Island, but at locations with no 3g/4g network availability, for this reason it was decided to use the satellite links. One of the devices was installed at Bahía del Diablo Meteorological Station (EMBD) to obtain temperature and humidity data from GBD in real time for the first time since the Meteorological Station installation. The remaining device includes a GPS and was installed on a glacier reference rock to measure its surface speed.

The remaining devices were installed without connectivity, only with an SD card to save temperature and humidity data. One of them was installed on EMBD as a backup for the devices already available at this location. The last one is placed on the Meteorological Station located in Matienzo Station.



Fig. 4. Installed devices in EMBD, Vega Island to measure temperature and humidity (right panel) next to installed device on a rock in GBD to monitor glacier displacement (left panel).

IV. DISCUSSION

In this work, an IoT device network for remote monitoring of various variables was developed and implemented in AP. The deployed devices continue successfully working, allowing real-time access to all measured data via web interface.

The currently deployed network is responsible for remote monitoring glaciers and environmental parameters. The devices capture and send images, measure air temperature, relative humidity and position through GNSS, and store data locally as back-up. However, this is only one of the possible applications. The same measurements can be applied to other fields of science, and other sensors compatible with i2c protocol, which allows multiple connections on a single bus, can also be used. This protocol is widely used and a large number of i2c compatible sensors can be found on the market.

It should be noted that the implemented system is in line with several monitoring networks developed in remote locations [11], [12]. These include the use of IoT devices for monitoring several parameters. In this sense, the possibility of adding new measurement modules or sensors without considerable effort and keeping the hardware design is relevant.

Having deployed this first version of the network and verifying its successful operation, one of the next objectives is to implement communication between the different nodes. This way, devices will be able to exchange information between them. This will allow taking greater advantage of the locations where communication is more stable and cheap.

Furthermore, it must be noted that the devices are operating in one of the most unfavorable environments on

the planet. The extreme weather conditions that take place in Antarctica constantly test the reliability and robustness of the designed system. This shows that the network can be deployed in remote locations without major inconvenience. The current installation also shows the low consumption and the high autonomy of the devices. The network, once deployed, does not require operators or maintenance other than battery replacement in case of eventual damage. Also noteworthy is the network versatility in terms of possible locations. Using 3g/4g network means that devices can be deployed in urban or suburban areas. Additionally, satellite communication means that the network can be installed in rural or isolated regions, which allows monitoring despite the site remoteness.

ACKNOWLEDGMENT

This project was supported by Ministerio de Defensa de la Nación through Programa de Investigación y Desarrollo para la Defensa (PIDDEF). Field campaigns and previous programs were funded by Instituto Antártico Argentino-Dirección Nacional del Antártico. We do thank the people from Comando Conjunto Antártico who participated in the field campaign and helped to achieve the presented results.

REFERENCES

- [1] X. Chen, L. Sun, H. Zhu, Y. Zhen and H. Chen, "Application of Internet of Things in Power-Line Monitoring". 2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, pp. 423-426, 2012.
- [2] S. R. J. Ramson, S. Vishnu and M. Shanmugam, "Applications of Internet of Things (IoT) – An Overview". 2020 5th International Conference on Devices, Circuits and Systems (ICDCS), pp. 92-95, 2020.
- [3] Gartner, "Forecast: IoT Meters by Use Case, Worldwide, 2020-2030", Gartner Research, accessed 9 Jun 2022, <<https://www.gartner.com/en/documents/3996804>>.
- [4] P. Skvarca, H. De Angelis and E. Ermolin, "Mass balance of 'Glaciar Bahía del Diablo', Vega Island, Antarctic Peninsula". *Annals of Glaciology*, vol. 39, pp. 209–213, 2004.
- [5] S. Marinsek and E. Ermolin, "10 Year mass balance by glaciological and geodetic methods of Glaciar Bahía del Diablo, Vega Island, Antarctic Peninsula". *Annals of Glaciology*, vol. 56, no 70, pp. 141–146, 2015.
- [6] Z. Y. Wang, J. Wang, Y. Q. Chen, M. H. Jia, G. Y. Zhang, Q. J. Tang, J. T. Chen, Y. H. Zhang, J. Zhang, H. F. Zhang and J. Chen "Design of EPICS and web-based remote control software of near-infrared sky brightness monitor in Antarctica." *IEEE Transactions on Nuclear Science*, vol. 66, no 8, pp. 1998-2004, 2019.
- [7] J. L. Pijoan, "Remote geophysical observatory in Antarctica with HF data transmission: A review." *Remote Sensing*, vol. 8, no 70, pp. 7233-7259, 2014.
- [8] J. Espina, T. Falck, A. Panousopoulou, L. Schmitt, O. Mühlens, and G. Z. Yang, "Network topologies, communication protocols, and standards". *Body sensor networks*, pp. 189-236, Springer, London, 2014.
- [9] Z. Y. Wang, J. Wang, Y. Q. Chen, M. H. Jia, G. Y. Zhang, Q. J. Tang, J. T. Chen, Y. H. Zhang, J. Zhang, H. F. Zhang and J. Chen "Design of EPICS and web-based remote control software of near-infrared sky brightness monitor in Antarctica". *IEEE Transactions on Nuclear Science*, vol. 66, no 8, pp. 1998-2004, 2019.
- [10] Quectel, "Quectel EC25 Series", Quectel EC25 brochure, 2020.
- [11] J. C. Zhao, J. F. Zhang, Y. Feng, and J. X. Guo, "The study and application of the IOT technology in agriculture". 2010 3rd international conference on computer science and information technology, vol. 2, pp. 462-465, 2010.
- [12] S. Adhya, D. Saha, A. Das, J. Jana and H. Saha, "An IoT based smart solar photovoltaic remote monitoring and control unit". 2016 2nd International Conference on Control, Instrumentation, Energy & Communication (CIEC), pp. 432-436, 2016.

Diseño de una plataforma para servicios IoT con aplicaciones en el sector productivo

Chazarreta Facundo A.
Programa TICAPPS, IlyA, UNAJ
facundochaza@hotmail.com

Salvatore Juan E.
Programa TICAPPS, IlyA, UNAJ
jsalvatore@unaj.edu.ar

Morales D. Martín
Programa TICAPPS, IlyA, UNAJ
martin.morales@unaj.edu.ar

Busum Fradera Matías
Programa TICAPPS, IlyA, UNAJ
matibf99@gmail.com

Osio Jorge R.
Línea CeTAD, GCA, Instituto LEICI
(UNLP- CONICET)
Programa TICAPPS, IlyA, UNAJ
josio@unaj.edu.ar

Resumen— El presente proyecto se basa en la utilización de internet de las cosas (IoT) como herramienta fundamental para proveer soluciones a problemáticas de interés social y en el sector productivo, contribuyendo en la obtención de datos que quedarán disponibles para el posterior procesamiento mediante técnicas de aprendizaje automático.

Entre los temas de investigación que se desarrollarán, se incluye el diseño e implementación de una plataforma para proveer servicios de IoT, que permita gestionar peticiones de clientes, y almacenar datos recibidos de dispositivos electrónicos de manera eficiente.

Por otro lado, se incluyeron herramientas para ayudar a la gestión eficiente durante la interacción con el sistema, tales como un servidor web open source y Node Red (necesaria para vincular los dispositivos de hardware con el servidor de IoT o Broker). Finalmente, se incluyeron dispositivos reales que permitieron evaluar las prestaciones de la plataforma, proporcionando muy buenos resultados en cuanto a los tiempos de respuesta a los cambios realizados en las mediciones en comparación con plataformas comerciales como thingspeak.

Palabras clave—plataforma IoT, protocolo MQTT, Servidor Web, Proxy inverso, Node Red, base de datos.

I. INTRODUCCIÓN

En la última década se han incrementado notablemente las tecnologías que permiten la gestión eficiente de sensores y dispositivos de HW (hardware) en aplicaciones IoT, especialmente en aplicaciones industriales (tecnologías de la industria 4.0)[1].

El diseño de la plataforma de aplicaciones con servicios de internet de las cosas se basa en tres aspectos principales:

- En primer lugar, se seleccionaron tecnologías y protocolos estandarizados para asegurar compatibilidad con la mayoría de las tecnologías y sistemas de IoT actuales [2].
- Además, se evaluaron las prestaciones de cada tecnología en base a la taxonomía de las diferentes arquitecturas dada en [2].
- Otro de los aspectos evaluados es el uso de herramientas de software libre que provean la posibilidad de agregar nuevas características.

Teniendo en cuenta las recomendaciones descritas en [2], para el diseño de la plataforma se seleccionaron herramientas para cubrir los siguientes aspectos de un sistema IoT:

- Implementación de control.
- Implementación de administración de dispositivos.
- Implementación de almacenamiento.
- Protocolo de aplicación.

La arquitectura implementada es una Arquitectura Orientada a Servicios (SOA), de tipo publicador suscriptor. Esta selección se fundamenta en que los dispositivos finales típicamente son de clases C1-C3 (con frecuencia de procesamiento entre 16 y 980Mhz y tamaño de datos de 8 a 32 bits), lo que significa que no tienen gran poder de cómputo para llevar adelante aplicaciones complejas de forma eficiente. Esta característica lleva a una separación de la funcionalidad (servicios) y la coordinación de estos (orquestración).

Respecto al protocolo de aplicación adoptado para la capa de aplicación, se utilizó el Message Queuing Telemetry Transport (MQTT) que es un protocolo ligero y simple para el estilo de arquitectura de publicación/suscripción con comunicación indirecta [3].

Actualmente, nuestro grupo está investigando y desarrollando tecnologías de la industria 4.0 orientadas al cuidado del medioambiente y problemática de interés social directamente ligadas al medio socio productivo (especialmente pequeños productores agrícolas e industriales de la zona de influencia). Todas estas líneas de investigación demandan la incorporación de las nuevas tecnologías para recabar grandes cantidades de datos en tiempo real. Además, se debe destacar que la plataforma desarrollada se complementa con aplicaciones de HW previamente desarrolladas para cubrir las necesidades mencionadas [4-6].

Con el diseño de la plataforma se pretende poder gestionar los datos provenientes de diferentes dispositivos, de manera que sean almacenados en una base de datos, pero también puedan ser visualizados por el usuario final. Además, dicha plataforma permitirá realizar el control de dispositivos y de los datos desde

que son enviados por el nodo, hasta que son procesados y almacenados en el servidor.

En la plataforma se implementó un bróker MQTT configurado específicamente con tópicos para la medición de parámetros de interés, (como por ejemplo temperatura y humedad), que se envían a través de dispositivos de hardware y son almacenados en una base de datos creada por el grupo de IoT e Inteligencia Artificial del Instituto, cumpliendo de esta manera con los requerimientos necesarios para la posterior gestión de la información. Finalmente, estos datos pueden ser accedidos por los clientes (suscriptores) y a su vez ser visualizados en una dashboard (pizarra) personalizada.

Para las pruebas de funcionamiento y la determinación de la respuesta del sistema en tiempo real se realizó una aplicación de automatización en invernaderos, apuntando a mejorar la eficiencia a la hora de utilizar recursos como agua y energía al independizarlos del factor humano. Para lograr que este sistema funcione de manera eficaz, primero se debe obtener una lectura de valores ambientales (humedad de suelo, humedad y temperatura), a través de dispositivos electrónicos, los cuales enviarán los mismos mediante “WiFi” hacia el servidor.

Se plantea como objetivo principal la implementación de la plataforma de aplicaciones IoT para brindar servicios que permitirán a las PyMEs gestionar y optimizar los parámetros de interés y la utilización de los recursos en todo tipo de ambientes, desde cualquier ubicación mediante un dispositivo móvil, Tablet, notebook, etc.

Entre los objetivos específicos se propone desarrollar un sistema de “Big Data” mediante una base de datos que se encargue de almacenar información relevante a lo largo del tiempo para un posterior análisis y procesamiento, permitiendo encontrar patrones para brindar recomendaciones que tiendan a optimizar las técnicas aplicadas y posibiliten el uso eficiente de los recursos.

El artículo está organizado en una introducción, en donde se describe la necesidad y los objetivos del trabajo; la implementación que contiene los detalles del sistema tanto de hardware (todo tipo de sensores), como de software (software embebido para lectura de sensores y control de actuadores, implementación de base de datos, configuración de un bróker Emqx, de la herramienta Node-Red y de Grafana). La descripción de los resultados obtenidos mediante las pruebas de campo y las conclusiones.

II. IMPLEMENTACIÓN

En la Fig. 1 se muestra un diagrama en bloques de uno de los nodos cliente, implementado con el fin de evaluar la plataforma que se propuso en los objetivos de este proyecto. Utilizando el módulo ESP32 se obtuvieron los valores provistos por un sensor DHT11 (que mide temperatura y humedad) y un YL-69 (que mide humedad del suelo) del entorno donde se encuentran ubicados. Por otra parte, se realizó el control de una electroválvula para el riego y mediante un led RGB se indica de forma visual dentro de que rango se encuentran los parámetros de

interés y si se superaron los umbrales de temperatura y humedad prefijados. El proceso de control se realiza desde el servidor y en caso de pérdida de conexión las decisiones las toma el ESP localmente. Finalmente, con un módulo GPS se obtiene la posición donde se encuentra ubicado el nodo cliente y se puede realizar un seguimiento de los dispositivos en caso de hurto.

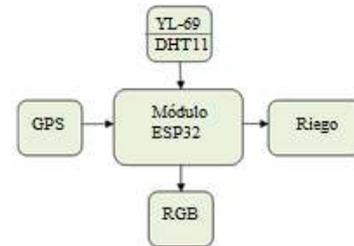


Fig. 1 Diagrama en bloques del nodo cliente

Los valores obtenidos mediante los sensores son enviados vía WiFi por medio del ESP32 al broker MQTT, el cual se encarga de recibir los mensajes enviados por los nodos cliente y distribuirlos entre sí mediante un sistema de publicación-suscripción a distintos tópicos.

A continuación, en la Fig. 2 se muestra un esquema de la plataforma IoT implementada en el servidor con el fin de poder gestionar la información de forma más eficiente.

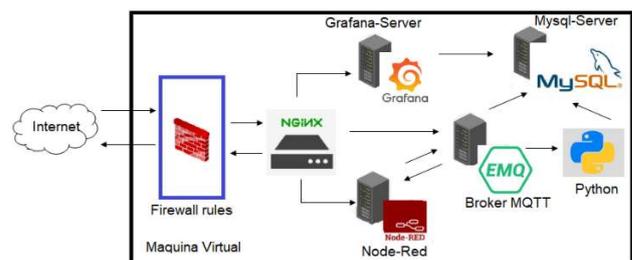


Fig. 2 Plataforma de aplicaciones IoT

En la gráfica anterior se observa que se utilizó NGINX, que es un servidor web open source de alta performance. Una de las prestaciones que ofrece es la de poder ser utilizado como proxy inverso para así poder reenviar las solicitudes entrantes a Grafana, o Node Red o bien al Broker MQTT EMQX.

Los usuarios solo pueden acceder al sistema mediante autenticación, la cual se realiza mediante el plugin emqx_auth_mysql de EMQX. Este plugin permite gestionar la autenticación de los usuarios que requieren conectarse al broker por medio de una base de datos SQL.

Cada nodo cliente, autorizado para acceder al servidor, podrá publicar en un tópico o bien suscribirse a uno por medio del broker MQTT. Éste último por medio de un algoritmo escrito en lenguaje Python permite almacenar los datos que llegan al broker en una base de datos implementada con Mysql. Por otra parte, Grafana es un software que permite consultar, visualizar, alertar y comprender distintos datos independientemente de donde se encuentren almacenados. De esta forma cada usuario podrá acceder a las dashboard de su sistema para ver en tiempo real el relevamiento de los datos de forma gráfica.

Finalmente, Node Red, que también es software libre, es un editor de flujo basado en el navegador donde se

pueden añadir o eliminar nodos y conectarlos entre sí con el fin de hacer que se comuniquen entre ellos. Conecta varios dispositivos a la vez, tanto de hardware como de servicios de Internet. Es perfecto para los equipos dedicados al trabajo industrial o para prueba de soluciones en equipos de planta que necesiten comunicarse entre sí. Se trata de un motor de flujo que cuenta con un enfoque IoT, ya que esta condición permite definir flujos de servicios a través de protocolos como el MQTT.

Cabe destacar que la implementación de la plataforma se realizó en una VPS con las siguientes prestaciones:

- 1 vCPU Intel Xeon
- 1.5 GB de RAM
- 10 GB SSD
- 100 GB transferencia
- 1 IP propia
- Sin límite de E-Mails y Sitios Web

III. RESULTADOS

En la Fig. 3 se muestra la dashboard a la que el usuario puede acceder para visualizar los datos sensados, la ubicación del sistema y realizar el control de riego.

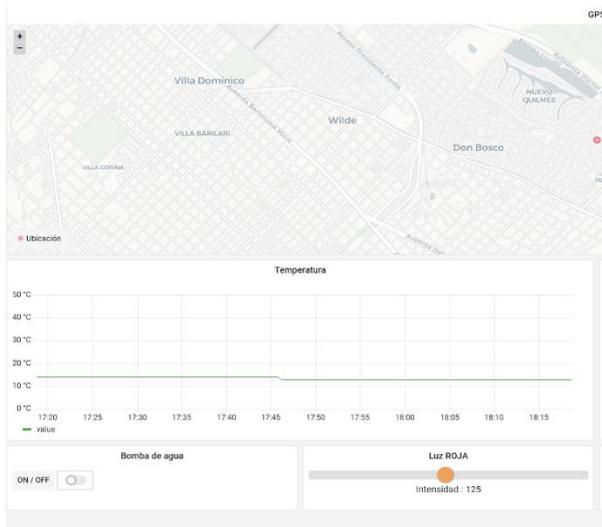


Fig. 3 Dashboard para visualización de sensores, actuadores y ubicación

Se debe destacar que las prestaciones logradas son excelentes, teniendo en cuenta que por lo general se miden fenómenos físicos con variaciones lentas. En ese sentido se realizaron pruebas de funcionamiento a intervalos de tiempo de entre 500 ms y 1 s. Obteniendo muy buenos resultados sin pérdida de datos. Para medir la latencia del sistema se utilizó un módulo RTC, en donde se fijó una timestamp, comparando esto con lo indicado por Node-Red, en donde los valores obtenidos siempre fueron inferiores a 1 s.

Respecto a la latencia, tomando la marca de tiempo del dispositivo esp32 al momento de enviar el dato y la marca de tiempo del servidor al momento de recibir el dato, en el peor caso la latencia medida fue del orden de un segundo lo que es muy bueno en función de la característica lenta de los parámetros que se relevan

habitualmente.

En comparación con algunas plataformas privadas como por ejemplo thingspeak, se pudo constatar que, para similares exigencias, se verificaron situaciones como pérdida de datos, o recepción de información con latencias de 15 segundos o más, por lo que hay muchas plataformas disponibles, pero muy pocas proveen buena calidad de servicio a un precio razonable.

IV. CONCLUSIONES

En primer lugar, se cumplió el objetivo principal de implementar una plataforma IoT completa cubriendo aspectos de seguridad como el acceso por autenticación. Además, se logró un funcionamiento eficiente y confiable, debido a que tiene una mucho mayor velocidad de respuesta que otras plataformas y muy poca variación de la latencia entre muestras.

Las plataformas de IoT resultan muy útiles cuando se cuenta con una base de datos que permita el almacenamiento de la información a lo largo del tiempo, ya que de esta manera se pueden obtener importantes cantidades de información para el entrenamiento de algún modelo de aprendizaje automático que permita optimizar el uso de los recursos y posibilite la toma de decisiones en pos de lograr un mejor aprovechamiento de los recursos.

Respecto a los beneficios proporcionados durante la utilización de la plataforma en el control de invernaderos. Se independizó el riego y el consumo del factor humano, se pudieron detectar picos de temperatura de forma automática generando alertas que permiten al usuario tomar acciones para mejorar la circulación de aire y evitar temperaturas extremas en ambientes cerrados. En definitiva, se permitió el acceso a la información de forma remota y la automatización del sistema, proporcionando las ventajas de las tecnologías de la Industria 4.0.

Como tareas a futuro, se pretende incorporar una amplia variedad de aplicaciones y medir el desempeño del sistema con la utilización de un mayor porcentaje de sus recursos.

Respecto a la capa de aplicación, es deseable a futuro instalar oneM2M (ideal para comunicación machine to machine) para disponer de los distintos protocolos de la capa de aplicación y seleccionar el más adecuado para cada aplicación.

REFERENCIAS

- [1] Luis joyanes Aguilar, "Industria 4.0, la cuarta revolución industrial, Alfaomega, 2016. ISBN: 978-607-622-942-2
- [2] Leila Fatmasari Rahman, "Choosing your IoT Programming Framework: Architectural Aspects", 2016 IEEE 4th International Conference on Future Internet of Things and Cloud
- [3] "MQTT: The Standard for IoT Messaging" mqtt.org (accedida en junio de 2022)
- [4] Osio J., C. Acquarone, E. Hromek, J. Salvatore, "Plataforma de desarrollo para IoT", IV conaiisi, 2017
- [5] J. Osio, J. Salvatore, M. Cappelletti, "Tecnologías de la información y las comunicaciones mediante IoT para la solución de problemas en el medio socio productivo", UNNE, WICC 2018.
- [6] J. Osio, M. Cappelletti, G. Suárez, J. Salvatore "Diseño de aplicaciones de IoT para la solución de problemas en el medio socio productivo", UNSJ, San Juan, WICC 2019.

Categoría

FORO TECNOLÓGICO

Procesamiento Digital de Señales (Digital Signal
Processing, DSP)

Área Temática

Extended bank of observers for fault detection in hexarotors

1st Claudio Pose

Facultad de Ingeniería -
Universidad de Buenos Aires and
Universidad de San Andrés - CONICET
Buenos Aires, Argentina
cldpose@fi.uba.ar

2nd Leonardo Garberoglio

Universidad Tecnológica Nacional
Facultad Regional San Nicolás
and CONICET
Ramallo, Buenos Aires, Argentina
lgarberoglio@frsn.utn.edu.ar

3rd Ezequiel Pecker-Marcosig

Instituto de Ciencias de la Computación
(ICC-CONICET) and Facultad de Ingeniería
Universidad de Buenos Aires
Buenos Aires, Argentina
emarcosig@dc.uba.ar

4th Ignacio Mas

Universidad de San Andrés
and CONICET
Buenos Aires, Argentina
imas@udesa.edu.ar

5th Juan I. Giribet

Universidad de San Andrés
and CONICET
Buenos Aires, Argentina
jgiribet@conicet.gov.ar

Abstract—A fault detection system for actuator faults based on a bank of observers is proposed for a multi-rotor aerial vehicle. Contrary to classical bank of observers, the presented solution considers a failure model that takes into account the speed decay after failure instead of assuming that the motor stops instantly. Experimental validation shows that the inclusion of this model allows for a faster failure detection.

Index Terms—Flight computer, Unmanned Aerial Vehicles, Fault Tolerance, Fault Detection and Identification

I. INTRODUCTION

During the last few years, small-scaled unmanned aerial vehicles (UAV) have become very popular. As commercial and civil uses increase, fault tolerance becomes an important aspect to be accounted for in unmanned vehicle design. The ability to continue normal flight in the event of a component failure is not only critical to ensure the integrity of the vehicle and prevent possible damages to third parties, but also to provide a high degree of reliability in the completion of sensitive missions.

In many occasions, multirotors with a high number of motors are used to increase payload capacity and improve stability and robustness against perturbations. In these cases, the design can be exploited to achieve fault tolerance against motor failures based on hardware redundancy, when certain conditions are satisfied [1], [2]. If hardware redundancy is not possible, more complex solutions based on partial loss of control, mechanical design and/or vehicle reconfigurability have been proposed to achieve fault tolerance.

In case of failure in one of the motors in a multirotor vehicle, the control strategy will heavily depend on which of the motors is the one presenting a failure, so that maximum performance can be achieved in any situation. In consequence, the failure has to be properly detected and identified in order to choose the optimal control solution, for which a Fault Detection and Identification (FDI) algorithm is required.

This kind of algorithms are usually implemented through an observer-based solution [3], and may be quite demanding in terms of required processing power.

This work proposes an advanced bank of observers that includes a model of how a faulty motor behaves. In particular, this bank incorporates the motor's transient response after a failure occurs and until it stops working completely. This algorithm allows for faster detection and identification, at the cost of an increased processing time. The latter is a limiting aspect when implementing this kind of solutions in low-level flight computers (FC) of the Cortex-M family. A flight computer was developed in our Lab, using a Cortex-M7 microcontroller and it is shown it is possible to run this FDI algorithm along with all the common guidance, navigation and control (GN&C) algorithms required in multirotor systems.

II. FLIGHT CONTROLLER IN UAV VEHICLES

The flight computer used in this paper was developed in our lab based on a STM32F722, an IC from the ARM Cortex-M7 family, that operates at 216 MHz and has an integrated single precision FPU, 512KB of flash memory and 256kB of RAM; 64kB of which are used to store data for critical real-time tasks and another 16KB to execute these tasks. It also includes a ICM20602 triaxial Inertial Measurement Unit (IMU), HMC5883 magnetometer, MS5611 barometer, and several communication ports for other sensors such as GPS, rangefinders, and optical flow, among others.

III. FAULT TOLERANT CONTROL FOR AN HEXAROTOR

The analysis on fault tolerance will be carried out for a hexarotor-type vehicle, which is depicted in Fig. 1. Its six rotors (M1 to M6) are placed at the vertices of a regular hexagon, at the same distance d_i from its geometric center.

For this kind of vehicles, there are several works that implement fault tolerant control systems, here the design proposed

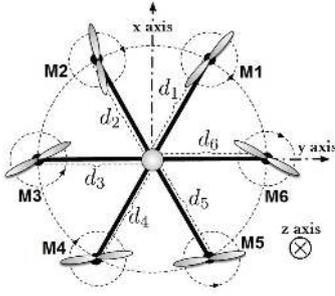


Fig. 1: Top view of a standard hexarotor vehicle distribution.

in [4] is used. To implement the fault detection and isolation system (FDI), there are two main approaches commonly found in the literature: direct measurement of the state of the motors, which requires additional sensors increasing the cost, and indirect measurement or estimation, which requires additional computational power.

The indirect measurement approach is generally implemented by means of an observer-based detection algorithm. In a typical attitude control loop of a multirotor, the navigation algorithm estimates the attitude using the on-board sensors, and then generates a desired torque \mathbf{q} to follow the reference, which is accomplished by commanding an adequate set of forces \mathbf{f} to the motors. The FDI subsystem uses the information of the commanded motor forces to predict the dynamics of the multirotor; if the difference between the predicted dynamics with respect to the real behavior (called the *residues*) is close to zero, then all the motors are working properly, if a deviation exists, it may be caused by the appearance of a failure.

The simplified vehicle dynamic for the UAV is shown below [5]:

$$\begin{aligned} \mathbf{x} &= [\phi \ \theta \ \psi \ \dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T, \quad \mathbf{y} = [\phi \ \theta \ \psi]^T \\ \dot{\mathbf{x}} &= \underbrace{\begin{bmatrix} \mathbf{0}^{3 \times 3} & \mathbf{1}^{3 \times 3} \\ \mathbf{0}^{3 \times 3} & \mathbf{0}^{3 \times 3} \end{bmatrix}}_{A_c} \mathbf{x} + \underbrace{\begin{bmatrix} \mathbf{0}^{3 \times 6} \\ I_m^{-1} A \end{bmatrix}}_{B_c} \mathbf{f}, \quad \mathbf{y} = \underbrace{\begin{bmatrix} \mathbf{1}^{3 \times 3} & \mathbf{0}^{3 \times 3} \end{bmatrix}}_{C_c} \mathbf{x} \end{aligned} \quad (1)$$

where $[\phi, \theta, \psi]$ correspond to the rotation around the x , y and z axes, and $I_m = \text{diag}(J_x, J_y, J_z)$ is the simplified inertia tensor. The exerted torque is $\mathbf{q} = [q_x, q_y, q_z]^T = A \cdot \mathbf{f}$, where \mathbf{f} is the vector of forces exerted by the motors and A is the force-torque matrix dependent on the motor disposition.

To implement a classical FDI algorithm, usually a bank of observers is used over the discrete counterpart of (1). Since the vehicle's state vector is observable, it is possible to design an observer to estimate its attitude around nominal state (i.e. hovering) and with all the motors working properly:

$$\begin{aligned} \hat{\mathbf{x}}_{k+1} &= A_d \hat{\mathbf{x}}_k + B_d \mathbf{f}_k + L(\mathbf{y}_k - \hat{\mathbf{y}}_k), \quad \hat{\mathbf{y}}_k = C_d \hat{\mathbf{x}}_k \\ \mathbf{r}_k &= \mathbf{y}_k - \hat{\mathbf{y}}_k \end{aligned} \quad (2)$$

where matrices A_d and B_d correspond to the discretization model of system (1) with sampling interval T_d , the \hat{i} notation corresponds to the estimation of the variable i , and \mathbf{r}_k is the residue at time k , the difference between the real measurement \mathbf{y}_k and the estimated (observed) output $\hat{\mathbf{y}}_k$.

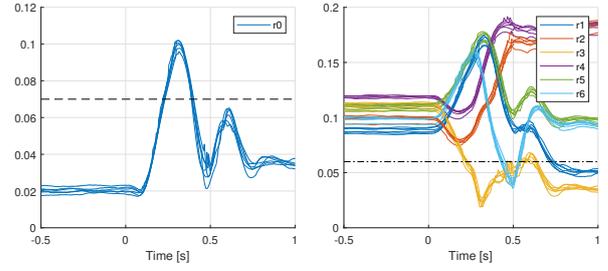


Fig. 2: Residues of a standard bank of observers.

So, if L is designed such as $A_d - LC_d$ is Hurwitz, and the motors are working properly (no failure), the observer is consistent with the behavior of the system, and the residue tends to zero. If a failure appears, the estimated output deviates from the measured output, which allows to use the norm of the residue as a failure detection subsystem.

In a similar way, considering a total failure in motor i with $i \in \{1, \dots, 6\}$, a similar observer to (2) may be proposed, but replacing B_d with $B_{d,i}$, which corresponds to the matrix B_d with its i -th column replaced by zeros. While the vehicle is in nominal state, the estimated output differs from the measured output, but, when a failure occurs in motor i , this observer becomes consistent and the norm of $\mathbf{r}_{k,i}$ tends to zero, identifying that this motor is the one presenting a failure. Adding one observer for every motor, a total failure can be identified. This set of observers is called a bank of observers.

Such a bank of observers was implemented in the FC, mounted in a fault tolerant hexarotor. The control algorithm is executed at a frequency of 200 Hz, which results in $T_d = 5$ ms. This means that every process that involves attitude estimation and control, execution of the FDI, command of the motors, and others, must be executed as a whole in less than 5 ms.

Several experiments were carried out to test the performance and robustness of the bank of observers, performing flights where a fault was injected mid-flight in one of the motors. The residues were recorded in real time, and their values for some of the flights are shown in Fig. 2. For four different flights, the vehicle was driven to a hovering state, remaining in a fixed position, and a fault was injected in $t = 0$ s by cutting the power to motor number 3 (M3 in Fig. 1). It is shown that, for a short time of around 100 ms, the residues show no change, due to the fact that the motor-propeller set continues spinning due to inertia, and is still producing thrust. After that, the detection residue r_0 that corresponds to the observer of the nominal plant begins to increase as the predicted output deviates from the measured output, while the residue r_3 that corresponds to motor 3 tends to zero. When both r_0 is greater than a detection threshold $\tau_d = 0.07$ and r_3 is lower than an isolation threshold $\tau_i = 0.06$, the fault is considered detected and isolated.

IV. EXTENDED BANK OF OBSERVERS

In the previous Section it was shown that the bank of observers works properly and allows to detect and isolate a failure in any motor. However, as can be noticed in Fig 3, there

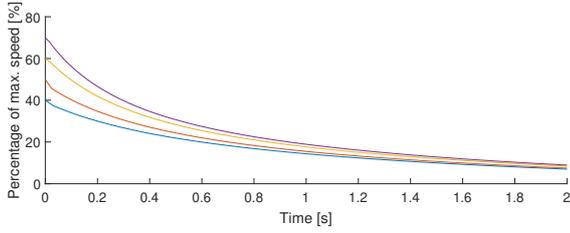


Fig. 3: Speed over time of an hexarotor motor that is initially working at 40%, 50%, 60% and 70% of its maximum speed.

are some types of failures where the motor remains spinning for a short period of time, producing force and torque due to the inertia. This effect causes a delay in the detection, as there is not an immediate change in the value of the residues.

Then, it is of interest to design a bank of observers that takes into consideration the possibility of this type of failure, i.e. one where the motor continues spinning after the failure, which is consistent with a sudden loss of power. That is, the aim is to design observers that are consistent with the exponential decay nature of a motor stopping.

However, designing a consistent observer for this behavior would require to meet two conditions. First, a model for the behavior of the motor is needed, which could be obtained by means of identification tests, such as the one represented in Fig. 3 for a 2213 920KV BLDC motor with 9545 plastic propeller. Second, it is required to estimate the exact moment when the failure occurs, which is very difficult to achieve.

One way to approximate the time when a failure is produced is to incorporate a finite set of new observers. A new observer is initialized every δ_t ; at that moment, it takes the current value of the force applied to its corresponding motor and considers that it goes to zero following the identified exponential decay. The proposition here is to place these observers in a circular queue, re-initializing them once every T_c , when it can be considered that the observers are not being consistent with the behavior of the vehicle. This parameter T_c can be defined from Fig. 3, considering the time for the motor to get to zero speed, or using practical considerations, such as a time limit to detect a failure. Moreover, in order for the circular queue to work properly, the sampling time T_c must be an integer multiple of δ_t .

The selection of T_c and δ_t imposes a trade-off between the precision in the fault detection, and the load imposed on the microcontroller. For a fixed δ_t , a longer T_c allows to evaluate the evolution of the residues over a larger interval of time, at the expense of increasing the number of observers that simultaneously run. On the other hand, for a fixed T_c , a smaller δ_t also means an increased number of observers, but enables for a more precise detection of the moment of occurrence of the failure.

In what follows, we consider $T_c = 1$ s and $\delta_t = 100$ ms, leading to ten new observers per motor. In Fig. 4 it is shown in blue the input force for motor 3 (M3 in Fig 1), where for the sake of clarity we only show the first five observers. Every time an observer is triggered, the force being commanded to motor

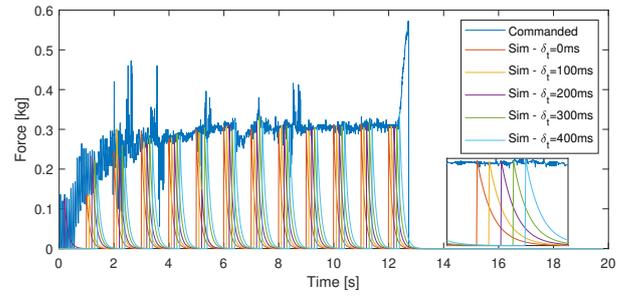


Fig. 4: Every $T_c = 1$ s, and with a separation of $\delta_t = 100$ ms, each observer takes the current force commanded to motor 3 and assumes a failure occurs at that instant.

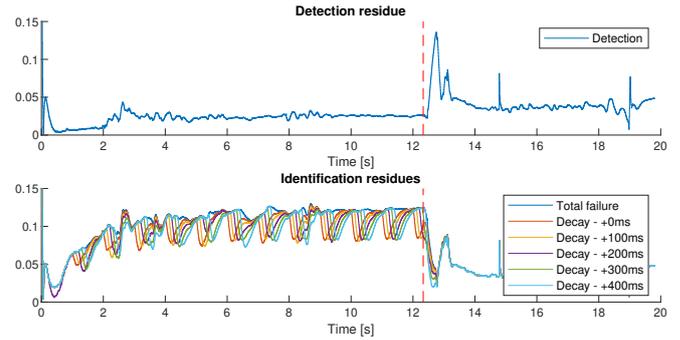


Fig. 5: Detection residue (top) and identification residues of motor 3 in the different observers (bottom), for a vehicle in which a failure occurs at $t_f = 12.33$ s.

3 is registered, and an exponential decay is simulated from that point on. Fig 5 shows the detection (top) and the new failure residues along with the original total failure residue (bottom). Moreover, Fig. 6 shows in detail what happens around $t_f = 12.33$ s when the failure occurs. The green and cyan residues, that correspond to the observers triggered closer to t_f , decay more quickly than the others. It can be noticed that for any threshold in the range $[0.04, 0.07]$ a detection time between 100 ms and 200 ms can be achieved. Therefore, the use of this solution would allow us to detect a failure in less than half the time required by the total failure observer (blue residue) based on the classic bank of observers.

The total execution time for the bank of observers in the Cortex M-7 was of $54 \mu\text{s}$ and was executed at 200 Hz, the same as the control loop, representing around 1% of the available processing power.

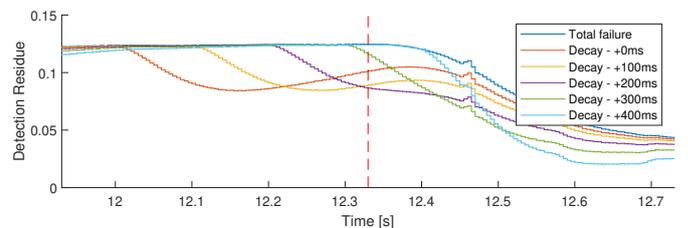


Fig. 6: Detail of the identification residues.

V. CONCLUDING REMARKS

The proposed algorithm incorporates the knowledge of the transient response of a faulty motor from the occurrence of failure until it stops working completely. This leads to up to a 50% reduction in the detection time when compared to a classical bank of observers.

The small impact in the microcontroller load despite the high number of observers is encouraging to continue exploring on other variations, which may contain more information, such as several types of decay for the same motor (due to winds or other factors).

REFERENCES

- [1] M. Saied, B. Lussier, I. Fantoni, C. Francis, and H. Shraim, "Fault tolerant control for multiple successive failures in an octorotor: Architecture and experiments," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 40–45, 2015.
- [2] J. I. Giribet, R. S. Sanchez-Pena, and A. S. Gherin, "Analysis and design of a tilted rotor hexacopter for fault tolerance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1555–1567, 2016.
- [3] D. Vey and J. Lunze, "Experimental evaluation of an active fault-tolerant control scheme for multirotor uavs," in *2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol)*, pp. 125–132, 2016.
- [4] C. D. Pose, J. I. Giribet, and I. Mas, "Fault tolerance analysis for a class of reconfigurable aerial hexarotor vehicles," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 4, pp. 1851–1858, 2020.
- [5] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.

Algoritmo adaptativo NLMS/F basado en conjunto de membresías

Laura J. Hidalgo, Ángel A. Vázquez, Xochitl Maya, Juan G. Avalos, Giovanni Sánchez
SEPI ESIME Culhuacan
Instituto Politécnico Nacional
Ciudad de México, México

laurahidalgo97@gmail.com, vapa1995@hotmail.com, xomaro94@hotmail.com, javaloso@ipn.mx, gsanchezriv@ipn.mx

Resumen— Actualmente, los filtros adaptativos se utilizan en diversas aplicaciones, tales como: identificación de sistemas, canceladores de eco acústico, control activo de ruido, ecualización de canales, entre otras. El componente más importante de los filtros adaptativos es el algoritmo adaptativo. Por lo tanto, varios autores realizan esfuerzos extraordinarios para desarrollar algoritmos adaptativos con buenas propiedades de convergencia y bajo costo computacional. En este trabajo, se presenta una nueva variante del algoritmo de mínimos promediados de cuarto orden normalizado NLMF (del inglés: Normalized Least-Mean-Fourth) basado en conjunto de membresías, además, se presenta un método para calcular automáticamente el tamaño del paso. Para evaluar su desempeño, el algoritmo fue probado en una aplicación de identificación de sistemas. Los resultados de la simulación demuestran que el algoritmo propuesto exhibe una baja complejidad computacional y buenas propiedades de convergencia.

Palabras clave— Algoritmo NLMF, algoritmo NLMS/F, conjunto de membresías, filtrado adaptativo.

I. INTRODUCCIÓN

Los algoritmos adaptativos son utilizados para ajustar los coeficientes de un filtro variante en el tiempo con el fin de minimizar un criterio previamente establecido. Uno de los algoritmos más utilizados es el algoritmo de mínimos cuadrados promediados normalizado NLMS (del inglés: Normalized Least-Mean-Square) debido a que su bajo costo computacional permite su implementación en una gran variedad de aplicaciones [1]. Asimismo, existen variantes que mejoran las propiedades de convergencia de dicho algoritmo, como el algoritmo de mínimos promediados de cuarto orden normalizado NLMF (del inglés: Normalized Least-Mean-Fourth). No obstante, su costo computacional e inestabilidad son mayores con respecto al algoritmo NLMS [2-4]. Esto ha motivado el desarrollo de variantes como el algoritmo NLMS/F [5,6], el cual es una combinación de los algoritmos NLMS y NLMF (del inglés: Least-Mean-Fourth) y presenta las mejores características de ambos.

Por otra parte, en años recientes diversos autores han propuesto métodos para reducir el costo computacional de los algoritmos adaptativos. Uno de los más usados es la técnica de conjunto de membresías SM (del inglés: Set-Membership) [7], en la cual los algoritmos solamente actualizan los coeficientes del filtro si la señal del error es mayor a un umbral previamente establecido.

En este trabajo se propone aplicar la técnica de conjunto de membresías al algoritmo NLMS/F. Además, se presenta un método que evita que algunas de las variables del

algoritmo dependan de la selección del usuario, facilitando de esta forma su implementación en diversas aplicaciones. Los resultados demuestran una considerable reducción en la carga computacional, además, la velocidad de convergencia del algoritmo propuesto es mayor con respecto a la versión convencional.

II. COMBINACIÓN DE LOS ALGORITMOS NLMS Y NLMF

En esta sección se presenta la ecuación del algoritmo NLMS para la actualización de los coeficientes del filtro está dada por:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu}{\mathbf{x}^T(n)\mathbf{x}(n)+\rho} \mathbf{x}(n)e(n) \quad (1)$$

donde $\mathbf{w}(n+1)$ es el vector de coeficientes en la iteración $n+1$, μ es el tamaño del paso, $\mathbf{x}(n)$ es la señal de entrada al filtro, ρ es una constante pequeña usada para evitar la indeterminación y $e(n)$ es la señal de error, la cual se obtiene mediante:

$$e(n) = d(n) - \mathbf{w}^T(n)\mathbf{x}(n) + \eta(n) \quad (2)$$

donde $d(n)$ es la señal deseada y $\eta(n)$ representa ruido aditivo. La función de costo del algoritmo NLMS se define como:

$$L_1(n) = \frac{1}{2} e^2(n) \quad (3)$$

La actualización de los coeficientes del filtro y la función de costo del algoritmo NLMF se presenta en (4) y (5), respectivamente.

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu}{\mathbf{x}^T(n)\mathbf{x}(n)+\rho} \mathbf{x}(n)e^3(n) \quad (4)$$

$$L_2(n) = \frac{1}{2} e^4(n) \quad (5)$$

Uniendo las funciones de costo de los algoritmos LMS y NLMF, se obtiene la función de costo de la combinación, la cual es llamada algoritmo NLMS/F y se expresa como:

$$L_3(n) = \frac{1}{2} e^2(n) - \frac{1}{2} \varepsilon \ln(e^2(n) + \varepsilon) \quad (6)$$

donde $\varepsilon > 0$ es un parámetro que regula la velocidad de convergencia y el rendimiento del algoritmo. De esta forma, la ecuación de actualización de coeficientes del algoritmo NLMS/F se expresa como:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu}{\mathbf{x}^T(n)\mathbf{x}(n)+\rho} \mathbf{x}(n)\beta(n) \quad (7)$$

donde $\beta(n)$ se expresa como:

$$\beta(n) = \frac{e^3(n)}{e^2(n)+\varepsilon} \quad (8)$$

donde a ε se le asigna un valor pequeño para evitar una indeterminación. Como se puede observar en (8), cuando $\varepsilon \gg e^2(n)$, el algoritmo NLMS/F se comporta como un algoritmo NLMF con un tamaño de paso de μ/ε , y cuando $\varepsilon \ll e^2(n)$, se comporta como un algoritmo NLMS.

III. FILTRADO ADAPTATIVO POR CONJUNTO DE MEMBRESÍAS

La estrategia denominada conjunto de membresías establece una regla para la actualización de los coeficientes del filtro, en la cual se requiere que la magnitud de la señal del error sea menor o igual a un umbral preestablecido, $\bar{\gamma}$, de modo que los coeficientes del filtro pertenezcan a un conjunto de soluciones factibles definidos mediante:

$$\Theta = \bigcap_{(x,d) \in S} \{\mathbf{w} \in \mathcal{R}^N: |d - \mathbf{w}^T \mathbf{x}| \leq \bar{\gamma}\} \quad (9)$$

donde S denota el conjunto de todos los posibles valores de \mathbf{x} y d . En algunas aplicaciones no es posible acceder a d , por lo que es necesario definir un conjunto de restricciones denominado $\mathcal{H}(n)$, el cual se expresa como:

$$\mathcal{H}(k) = \{\mathbf{w} \in \mathcal{R}^N: |d(n) - \mathbf{w}^T \mathbf{x}(n)| \leq \bar{\gamma}\} \quad (10)$$

Los límites de $\mathcal{H}(n)$, son considerados como un conjunto de hiperplanos paralelos $e(n) = \pm\bar{\gamma}$, donde la intersección entre el conjunto de restricciones y los hiperplanos es denominado conjunto de membresías $\psi(n)$.

$$\psi(n) = \bigcap_{i=0}^k \mathcal{H}(i) \quad (11)$$

Específicamente, si el conjunto $\psi(n)$ se encuentra en $\mathcal{H}(i)$, no se requiere calcular los coeficientes del vector \mathbf{w} , ya que estos se encuentran dentro del conjunto solución.

IV. ALGORITMO SM-NLMS/F

Para aplicar la teoría SM en el algoritmo NLMS/F, primeramente, se sustituye el término del error, $\beta(n)$, en la ecuación del tamaño del paso del algoritmo SM-NLMS convencional [5], lo cual se expresa como:

$$\mu(n) = \begin{cases} 1 - \frac{\bar{\gamma}}{|\beta|} & \text{si } |\beta| > \bar{\gamma} \\ 0 & \text{Caso contrario} \end{cases} \quad (12)$$

Al aplicar la regla expresada en (12), se reduce el costo computacional del algoritmo ya que cuando el tamaño del paso es igual a 0 no se requiere realizar la actualización de los coeficientes del filtro.

Por otra parte, se propone que ε se ajuste automáticamente mediante:

$$\varepsilon(n) = \begin{cases} \beta^2(n) & \text{si } |\beta| > \bar{\gamma} \\ \varepsilon(n) & \text{Caso contrario} \end{cases} \quad (13)$$

El valor $\bar{\gamma}$ se calcula como el método convencional SM, $\bar{\gamma} = \sqrt{5}\sigma_{e_0}$, donde $\sigma_{e_0}^2$ denota la varianza del ruido aditivo.

Por otra parte, cabe mencionar que el comportamiento de la señal $e^2(n)$ en las primeras iteraciones del algoritmo NLMS/F suele ser mayor a ε , debido a que a ε se le asigna un valor pequeño. Al establecer que $\beta^2(n)$ sea más grande que $\beta(n-1)$, el algoritmo tendrá una tendencia a comportarse como un algoritmo NLMF y al aplicar la técnica de conjunto de membresías, el número de actualizaciones se reducirá considerablemente debido a que solo actualizará al algoritmo cuando $|\beta(n)|$ supera el umbral $\bar{\gamma}$.

V. RESULTADOS

Con la finalidad de mostrar el funcionamiento de la propuesta, se realizaron dos experimentos en el software MATLAB, en los cuales se comparó el funcionamiento del algoritmo SM-NLMS/F con el algoritmo NLMS/F en una estructura de identificador de sistemas.

En la Fig. 1 se muestra la estructura de identificador de sistemas que se utilizó para las simulaciones. En el primer experimento se usó como señal de entrada ruido blanco Gaussiano con varianza unitaria. El sistema desconocido es un filtro de respuesta finita al impulso (FIR, del inglés: finite impulse response) pasa bajas. La señal deseada es afectada por un ruido Gaussiano aditivo con una relación señal-ruido SNR de 33dB. El filtro adaptativo y el sistema desconocido cuentan con 16 coeficientes. Para el algoritmo NLMS/F se eligió $\mu = 0.4$, dicho valor se obtuvo a prueba y error, y se estableció cuando se encontró la mayor velocidad de convergencia. Para evaluar el funcionamiento de los algoritmos se obtuvo el nivel de MSE al promediar 200 experimentos. Además, a la mitad del proceso se realizó un cambio abrupto al multiplicar los coeficientes del sistema desconocido por -1, esto con la finalidad de evaluar las capacidades de seguimiento del algoritmo.

Como se muestra en la Fig. 2, el algoritmo propuesto presenta la mayor velocidad de convergencia tanto al inicio del experimento como al momento de realizar el cambio de coeficientes en el sistema desconocido. Mientras que el algoritmo NLMS/F presenta un mejor nivel de MSE. Sin embargo, al aplicar la técnica de conjunto de membresías el algoritmo propuesto no requiere de actualizar los coeficientes del filtro durante todo el proceso. Para este experimento solamente se realizó 484 actualizaciones, es decir reduce el número de actualizaciones hasta un 76% mientras que el algoritmo LMS/F requiere de realizar el cálculo en cada una de las iteraciones.

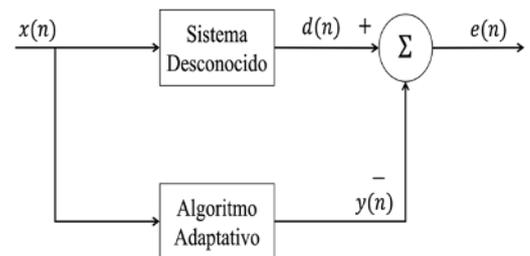


Fig. 1. Diagrama general de un identificador de sistemas.

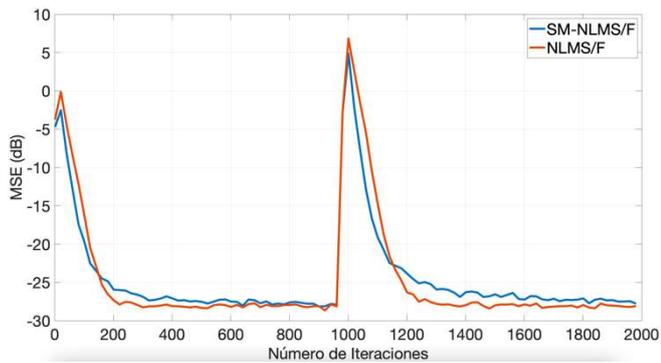


Fig. 2. Nivel de MSE de los algoritmos NLMS/F y SM-NLMS/F para el primer experimento.

En el segundo experimento se utilizó como señal de entrada el ruido de una turbina de avión. El sistema desconocido es un filtro FIR pasa bajas, en este caso tanto el filtro adaptativo como el desconocido contaron con 32 coeficientes. A la salida del sistema desconocido se añadió ruido aditivo con una SNR de 33 dB. Para el algoritmo NLMS/F se estableció $\mu = 0.5$, ya que este valor entregó el mejor desempeño del algoritmo. Al igual que en la primera prueba, las curvas de aprendizaje se obtuvieron al promediar 200 experimentos y también se realizó un cambio abrupto a la mitad de las iteraciones para probar las capacidades de seguimiento.

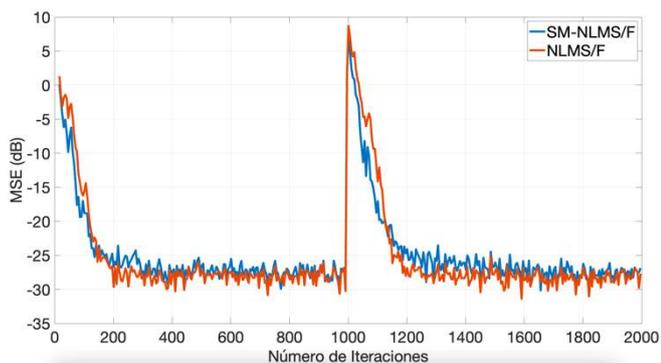


Fig. 3. Nivel de MSE de los algoritmos NLMS/F y SM-NLMS/F para el segundo experimento.

Como se muestra en la Fig. 3, el algoritmo propuesto presenta una mayor velocidad de convergencia con respecto al algoritmo NLMS/F. Cabe mencionar, que en esta prueba el

nivel de MSE de ambos algoritmos es muy similar. Por otra parte, el algoritmo SM-NLMS/F solamente requirió calcular sus coeficientes en 5,430 iteraciones, es decir se reduce en un 79% el número de actualizaciones.

VI. CONCLUSIONES

En este trabajo, se presentó un algoritmo adaptativo NLMS/F al cual se le aplicó la teoría del conjunto de membresías. Además, se propuso un método para establecer de manera automática el tamaño del paso del algoritmo. Los resultados experimentales demuestran que el algoritmo propuesto alcanza una mayor velocidad de convergencia y un nivel de MSE similar con respecto al algoritmo NLMS/F. Además, el número de multiplicaciones y sumas realizadas al ejecutar el algoritmo se reducen ampliamente, debido a que no se requiere calcular los coeficientes del filtro durante todo el proceso adaptativo. Como trabajo a futuro se realizará la implementación del algoritmo en un dispositivo FPGA (del inglés: Field Programmable Gate Array) para validar su funcionamiento, así como su consumo de área/ potencia.

REFERENCIAS

- [1] S. Haykin, *Adaptive Filter Theory*. Pearson, 2014.
- [2] A. Zerguine, "Convergence and steady-state analysis of the normalized least mean fourth algorithm," *Digital Signal Processing*, vol. 17, no. 1, pp. 17–31, Jan. 2007, doi: [10.1016/j.dsp.2006.01.005](https://doi.org/10.1016/j.dsp.2006.01.005).
- [3] E. Eweda and N. J. Bershad, "Stochastic Analysis of a Stable Normalized Least Mean Fourth Algorithm for Adaptive Noise Canceling With a White Gaussian Reference," in *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6235–6244, Dec. 2012, doi: [10.1109/TSP.2012.2215607](https://doi.org/10.1109/TSP.2012.2215607).
- [4] A. Zerguine, "Convergence behavior of the normalized least mean fourth algorithm," *Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers (Cat. No.00CH37154)*, 2000, pp. 275–278 vol.1, doi: [10.1109/ACSSC.2000.910958](https://doi.org/10.1109/ACSSC.2000.910958).
- [5] B. Mohanty, H. K. Sahoo, y B. Patnaik, "Block NLMS/F-based equalizer design and channel capacity analysis for indoor IEEE 802.11 fading wireless channels", *Signal, Image and Video Processing*, vol. 13, núm. 4, pp. 693–701, 2019 doi: [10.1007/s11760-018-1398-2](https://doi.org/10.1007/s11760-018-1398-2).
- [6] B. Mohanty, H. K. Sahoo and B. Patnaik, "NLMS/F Based Adaptive Beam former for Indoor Wireless Channel," *2017 Ninth International Conference on Advanced Computing (ICoAC)*, 2017, pp. 87–91, doi: [10.1109/ICoAC.2017.8441186](https://doi.org/10.1109/ICoAC.2017.8441186).
- [7] P. S. R. Diniz, *Set-Membership Adaptive Filtering*. In: *Adaptive Filtering*. Boston, MA: Springer US, 2020. doi: [10.1007/978-3-030-29057-3_6](https://doi.org/10.1007/978-3-030-29057-3_6)

Categoría

FORO TECNOLÓGICO

Protocolos y Comunicaciones

Área Temática

Sistema celular alternativo y de emergencia basado en OpenBTS y SDR

Alian Ernesto Matos Rodríguez, Ing.
Universidad Tecnológica de la Habana
La Habana, Cuba
aematos@nauta.cu

Marcelino Sánchez Posada, Dr.C.
Centro de Investigación Naval
La Habana, Cuba
msanchez82@nauta.cu

Resumen—El Sistema Global de Comunicaciones Móviles (GSM) ha tenido una gran aceptación alrededor del mundo. Para implementar estos sistemas era necesario utilizar equipos de costo elevado, sin embargo, gracias al desarrollo de sistemas de radios definidos por software, es posible desarrollar este tipo de redes con un costo reducido. El acceso a servicios de telecomunicaciones en comunidades rurales de Cuba es hoy en día un gran reto, propiciado principalmente, por la ubicación geográfica y el reducido número de habitantes que hacen que el despliegue de infraestructura no sea económicamente factible. Las contribuciones principales del presente trabajo son un sistema celular alternativo y software de administración de red GSM. Quedó demostrado que es posible la construcción de una red telefónica móvil usando componentes de software libre y hardware de bajo costo, que pueden ser empleados en comunidades rurales de poca población, así como en situaciones de emergencia.

Palabras claves—GSM, OpenBTS, SDR, Asterisk.

I. INTRODUCCIÓN

El Sistema Global de Comunicaciones Móviles (GSM, siglas en inglés) es sin duda una de las mayores obras de la ingeniería en telecomunicaciones por su aporte técnico a la arquitectura celular [1]. Anteriormente para implementar los sistemas GSM, ha sido necesario utilizar equipos de costo elevado, sin embargo gracias al desarrollo de sistemas de radios definidos por software (SDR, siglas en inglés) y OpenBTS, es posible desarrollar este tipo de redes con un costo reducido [1, 2]. Los SDR permiten realizar las funciones vía software típicamente implementadas mediante hardware, mientras que OpenBTS es un punto de acceso de GSM basado en software que posibilita el acceso de los teléfonos móviles compatibles con el estándar GSM.

El acceso a servicios de telecomunicaciones en las comunidades rurales de Cuba es hoy en día un gran reto. Esto es propiciado principalmente, por la ubicación geográfica y el bajo número de habitantes que hacen que el despliegue de infraestructura no sea económicamente factible para el operador de telecomunicaciones tradicional. Además, no se cuenta con un sistema alternativo que pueda ser utilizado en situaciones de emergencia, ante daños en la infraestructura de comunicaciones y saturación de los servicios de las estaciones base en determinadas áreas.

La implementación de estaciones base GSM mediante SDR y OpenBTS ha sido abordada por diferentes autores en el transcurso de los años. En las propuestas analizadas [1-5] se obtuvieron como resultados sistemas GSM de bajo costo y fácil implementación con el empleo herramientas de hardware USRP (Universal Software Radio Peripheral, en lengua inglesa) y software de licencia libre. Se evidencia que en todas las propuestas identificadas emplean como sistema operativo Ubuntu y como dispositivos hardware USRP de diferentes modelos: N200, N210 y B210.

Todos los trabajos analizados carecen de un software para la gestión de la red GSM, realizan de forma manual el registro de terminales móviles, de extensiones, la inicialización de servicios, entre otros, mediante comandos en la terminal del sistema operativo.

Las contribuciones principales del presente trabajo son un sistema celular alternativo y software de administración de red GSM, que pueden ser empleados en comunidades rurales de poca población, así como en situaciones de emergencia, donde se necesiten soluciones de bajo costo, portables y de pronta instalación.

II. DESARROLLO

El sistema OpenBTS, es definido como una aplicación de software libre desarrollado bajo el sistema operativo multitarea Unix y que utiliza mayormente el hardware USRP para construir la interfaz inalámbrica de radio, para emular al estándar de comunicaciones móviles GSM. Lo que posibilita que los teléfonos celulares circundantes detecten una completa red GSM, y a su vez, estos sean vistos como extensiones del Protocolo de Inicio de Sesión (SIP, siglas en inglés), permitiendo montar un sistema de conmutación o central telefónica gracias al software de licencia libre Asterisk. Ver Fig.1.

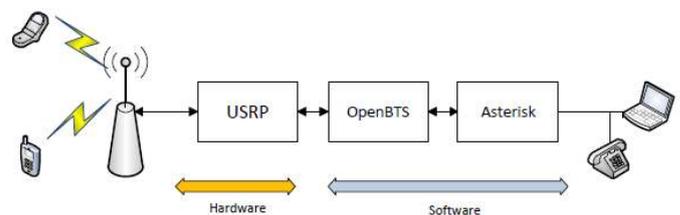


Fig. 1. Sistema OpenBTS.

La implementación de la red celular propuesta se realizó en el Sistema Operativo Ubuntu 14.04 y se utilizó como hardware SDR un USRP B200.

A. Asterisk

Asterisk es un proyecto de código abierto que agrupa la funcionalidad de una central telefónica privada (PBX, siglas en inglés) dentro de un paquete de software, permite realizar el registro, conexión y comunicación de terminales telefónicos conectados al servidor Asterisk. Ofrece las mismas características y servicios que un moderno equipo de hardware PBX.

B. USRP B200

Se trata de un SDR diseñado por la empresa “ETTUS Research” como propuesta de hardware libre, donde los microprocesadores convencionales pueden actuar como dispositivos de radio bajo un gran ancho de banda,

convirtiéndose en una plataforma flexible de bajo costo que permite implementar y diseñar potentes sistemas de radiocomunicaciones con aplicaciones en tiempo real. En esencia, sirve como procesador digital de banda base y convertidor de frecuencia intermedia FI en los sistemas de radiocomunicaciones.

C. Antenas

En las pruebas de laboratorio del sistema OpenBTS se utilizó la antena VERT900 mostrada en la Fig. 2(a), y en las pruebas en el terreno una antena de polarización vertical como se observa en la Fig. 2(b).



Fig. 2. (a) Antena VERT900 (b) Antena de polarización vertical.

Las características de las antenas antes mencionadas se muestran en la Tabla I.

TABLA I. CARACTERÍSTICAS DE LAS ANTENAS UTILIZADAS

Antena	Frecuencia	Ganancia
VERT 900	824-960 MHz, 1710-1990 MHz	3 dBi
Polarización vertical	906-960 MHz, 1710-2500 MHz	12.5/15 dBi

D. Banda de operación

OpenBTS puede usar un amplio rango de frecuencias de GSM como por ejemplo GSM850, GSM900, GSM1800 y GSM1900. Para la implementación del sistema se seleccionó la frecuencia de 900 MHz. De la misma forma se establece una banda de operación para OpenBTS, esta es definida por el Número de Canal de Radio-Frecuencia Absoluta (ARFCN, siglas en inglés), el cual define el método de acceso múltiple en GSM y provee los canales downlink y uplink de la estación móvil para establecer la radiocomunicación con la Estación Base Celular (BTS, siglas en inglés). Ver Tabla II.

TABLA II. BANDAS DE FRECUENCIA GSM Y NÚMEROS DE CANAL ARFCN

	GSM-850	GSM-900	GSM-1800	GSM-1900
Rango de frecuencia uplink	824-849 (MHz)	890-915 (MHz)	1710-1785 (MHz)	1850-1910 (MHz)
Rango de frecuencia downlink	869-894 (MHz)	935-960 (MHz)	1805-1880 (MHz)	1930-1990 (MHz)
ARFCN	128-251	1-124	512-885	512-810
Offset	45 MHz	45 MHz	95 MHz	80 MHz

E. Características del sistema OpenBTS

Algunas de las características principales del sistema implementado:

- Soporta funciones propias GSM 2G y 2.5G.
- Opera en las bandas de 850, 900, 1800 y 1900 MHz.
- Multiplexación en tiempo TDMA y en frecuencia FDMA.
- Soporta servicios de voz, envío de mensajes SMS y de datos GPRS.
- Traslada el tráfico GSM a VoIP.
- Ofrece servicios de gestión del sistema, encriptación y anonimato.
- Puede emplearse como terminal GSM cualquier móvil comercial.
- Bajo costo y rápido despliegue.
- Permite estructura multi-Estación Base y configuración simple o múltiples canales ARFCN.
- El área de cobertura según la densidad del área y características del terreno puede alcanzar los 20Km.

F. Software para la gestión de la red GSM

Con el objetivo de facilitar el empleo del sistema celular basado en OpenBTS se desarrolló un software en el lenguaje de programación C++ mediante el Entorno de Desarrollo Integrado (IDE, siglas en inglés) Qt Creator. En la Fig. 3 se muestra la Interfaz Gráfica de Usuario (GUI, siglas en inglés) de la aplicación.

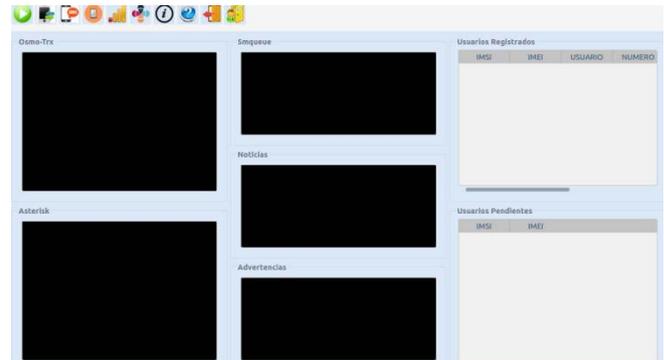


Fig. 3. Software para la gestión de la red GSM.

G. Principio de funcionamiento del software

El software realizado ejecuta pruebas de funcionamiento y conectividad del equipo de hardware USRP B200. Verifica el estado de todos los servidores que componen la estación celular, Asterisk, el servidor de registro, el servidor de mensajes de texto y el transceiver. Realiza el escaneo de las bandas de frecuencia GSM activas junto con los identificadores ARFCN usados por las redes móviles que operen dentro del área, muestra el número de canal ARFCN, junto con la potencia y la frecuencia de cada canal, con esta información se evita usar un mismo número de ARFCN en la red y causar una interferencia a la operadora nacional de telefonía móvil. En la Fig. 4 se muestra la aplicación en el proceso de comprobación de los componentes de hardware y software.

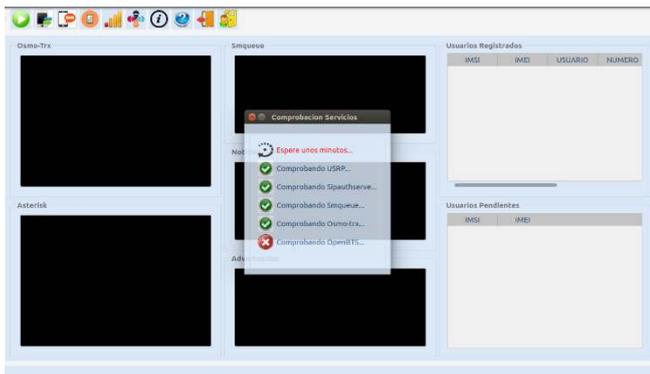


Fig. 4. Comprobación de los componentes de hardware y software.

Una vez concluido el proceso de verificación de hardware y software satisfactoriamente, los usuarios registrados pueden acceder a todos los servicios de la red de telefonía móvil.

III. PRUEBAS Y RESULTADOS

Para la realización de las pruebas se empleó el esquema de conexión básico que se muestra en la Fig. 5.

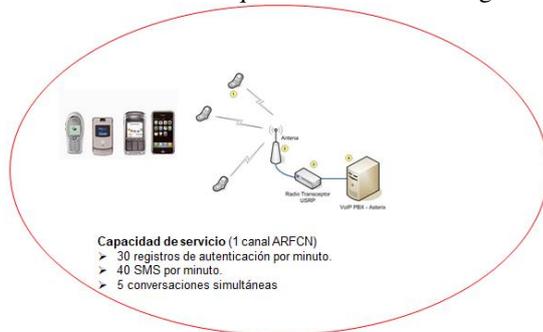


Fig. 5. Esquema de conexión básico del sistema OpenBTS.

Dicho esquema está compuesto por el sistema OpenBTS en una sola computadora de escritorio conectada al transceptor USRP B200. Primeramente, se verificó el correcto funcionamiento en un entorno controlado en el laboratorio con la antena VERT 900. Para ello, se registraron 30 usuarios en el sistema, dando la posibilidad de realizar llamadas y envío de mensajes de forma simultánea.

Con la certeza de los resultados de las pruebas de laboratorio, se realizaron experimentos en el terreno con la antena de polarización vertical con el fin de determinar: el máximo alcance de la señal en espacio libre, cantidad de registro de autenticación por minuto, cantidad de llamadas simultáneas, envío de mensajes de texto por minuto y el correcto funcionamiento del software de gestión de la red GSM.

Los resultados de las pruebas arrojaron un alcance del sistema de 200 metros en espacio libre, superior a lo logrado por los sistemas expuestos en [2, 3], siendo de 10 metros y 120 metros respectivamente. Se logaron 30 registros de autenticación por minuto, 5 conversaciones simultáneas y 40 mensajes de texto por minuto. Además, se validó el correcto funcionamiento del software de gestión de la red GSM implementado.

IV. CONCLUSIONES

Quedó demostrado mediante la práctica y experimentación que es posible la construcción de una red telefónica móvil usando componentes de software libre y hardware de bajo costo. El uso de Asterisk y las características del protocolo SIP, facilitó el manejo de los terminales de usuario y la conmutación de las llamadas de móvil a móvil, lo que propició un servicio telefónico de buena calidad tanto en voz como en tiempo de respuesta. Se logró satisfactoriamente la integración de la tecnología GSM, Asterisk y SDR, con la finalidad de despertar el interés en la utilización de las redes abiertas de telefonía celular como solución a la necesidad de comunicación en sectores rurales y escenarios de emergencia.

REFERENCIAS

- [1] M. Quiñones, J. P. Tene, K. Rohoden, and C. Carrión, "Sistema de conmutación y control de llamadas basado en OpenBTS y Asterisk," *Maskana*, vol. 5, 2014.
- [2] J. H. H. A. Garcia, C. J. P. Rodríguez, F. E. Ñ. Ruiz, and J. L. A. Alarcón, "Implementación de una red celular GSM mediante software OPENBTS," *Pueblo Continente*, vol. 30, pp. 101-108, 2019.
- [3] A. J. Ospino Polanco and C. A. Díaz Villadiego, "Implementación de una herramienta de radio definido por software que emule una red de telefonía celular móvil para ser usada en la enseñanza dentro del programa de ingeniería electrónica," Corporación Universidad de la Costa, 2020.
- [4] M. M. Macario, R. R. T. Martínez, L. A. S. Tamayo, C. A. Aguilar, J. R. R. Cruz, and C. G. Cervantes, "Diseño de una estación base GSM usando openbts y tarjeta de desarrollo usrp B210," *Investigación de las agujas hipodérmicas convencionales y su falta*, 2016.
- [5] H. A. P. Guerrero, D. M. J. M. García, M. A. C. Juárez, U. P. Rico, E. S. Navarro, and A. A. Casas, "Implementación de red celular de bajo costo para comunidades rurales basada en sdr y openbts (Implementation of a low cost cellular network for rural communities based on sdr and openbts)," *Pistas Educativas*, vol. 40, 2018.

Categoría

REPORTE

FPGAs, HDLs y ASICs

Área Temática

A Simple Low Voltage, Temperature Compensated MOSFET Dosimeter for embedded applications

Pablo Petrashin

Laboratorio de microelectrónica
Universidad Católica de Córdoba
Cordoba, Argentina
ppetrashin@gmail.com

Walter Lancioni

Laboratorio de microelectrónica
Universidad Católica de Córdoba
Cordoba, Argentina
walter.lancioni@gmail.com

Agustin Laprovitta

Laboratorio de microelectrónica
Universidad Católica de Córdoba
Cordoba, Argentina
agustin.laprovitta@gmail.com

Fortunato Dualibe

Faculté polytechnique de Mons (FP)
Service d'électronique et de
microélectronique (SEMi)
Mons, Belgium
Fortunato.DUALIBE@umons.ac.be

Juan Castagnola

Laboratorio de microelectrónica
Universidad Católica de Córdoba
Cordoba, Argentina
juanluiscastagnola@gmail.com

Abstract— It is introduced in this paper a new concept circuit for using in radiation measurement. With only 4 transistors it is achieved an output proportional to radiation and temperature independent, compensated using transistor geometry. Its small size makes this circuit especially suited for embedded applications. Simulations and measurements have been carried out in order to prove the behavior of the proposed configuration, using a widely extended and well known circuit as is the cd4007. The idea has proven to be reliable, achieving very good performance with a minimum area and complexity.

Index Terms— CMOS Dosimeters, Radiation measurement, Temperature compensation, embedded applications.

I. INTRODUCTION

NOWADAYS it is becoming more and more usual the usage of CMOS sensors in environments where the level of ionizing radiation is high, such as space, therapy and radiology systems, power plants, etc. These sensors are actually called dosimeters in radiation therapy applications, giving a readout of total ionizing dose (TID), using some radiation dependent parameter for sensing, such as, typically, the transistors threshold voltage, among others such as transconductance variations and leakage current increasing [1].

There are basically two types of radiation: particle radiation and photon radiation [2, 3]. Particle radiation is due to impacts of charged or neutral particles such as protons, electrons, ions, alpha particles and neutrons. The other type, photon radiation, consists of high energy x-rays and gamma rays. This kind of radiation is the one received when working with cancer treatment or diagnosis. It is common practice to measure the TID in Grey [Gy], which is defined as the absorption of one joule of radiation energy per kilogram of matter.

Pairs electron-hole are generated in the oxides and insulators due to ionizing radiation [3]. This effect can lead to device degradation and failure, but also it can be used for determining radiation dose through the measuring of TID, that is called a long term effect. As well, there are some single event effects that are short term, called Single Event

Effects (SEE) that are present for a small interval of time, causing momentary, yet sometimes permanent, changes in the device properties [4, 5].

Traditionally, the shift in threshold voltage V_t with the radiated dose has been commonly used as the measuring parameter [4-6]. One of the main problems when using MOS circuits for dosimetry is the shifting of V_t due to temperature, effect that can lead to incorrect readings [6]. While this is not a serious problem for high dose measurements, special care should be taken to avoid this effect when measuring low doses of radiation. Different techniques were proposed for this purpose [7, 8].

A novel configuration to perform TID measurements is presented in this work, introducing a very simple circuit that performs temperature cancellation taking advantage of different threshold voltages temperature coefficients. The paper goes first explaining the physical principle, then the proposed circuit, an implementation using commercial CD4007 along with simulations and measurements.

II. TOTAL IONIZING DOSE EFFECTS

As it was mentioned before, a high energy particle or photon can generate even thousands of electron-hole pairs in the oxide, which is the main reason of almost all TID effects [3].

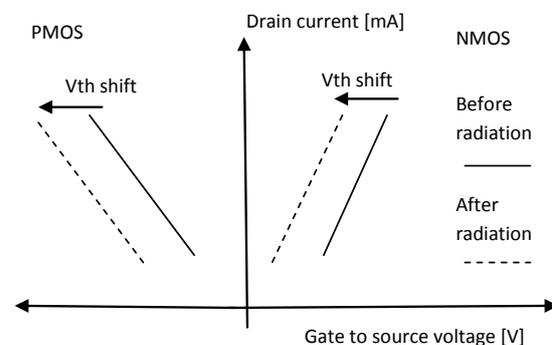


Fig. 1. Threshold voltage shift V_{th} due to the oxide trapped charges, after [1].

The threshold voltage shift ΔV_{th} is negative. This means that the shifting produces a decrement of V_{th} for NMOS transistor and an increment of the absolute value of V_{th} for the PMOS devices. This can be seen in Figure 1. This effect is the key idea behind this proposal.

III. PROPOSED CONFIGURATION

For achieving the function proposed at the beginning, circuit in figure 2 is presented, followed by mathematical analysis.

As it can be seen in Fig. 2, the circuit consists in two branches, one with pMOS transistors, that is M1 and M2, and the other with nMOS transistors, being M3 and M4. This allows to cancel the mobility per branch, being the mobility a parameter that varies with temperature and radiation. For this first approach, it will be not taken into account the body effect of transistors M2 and M3.

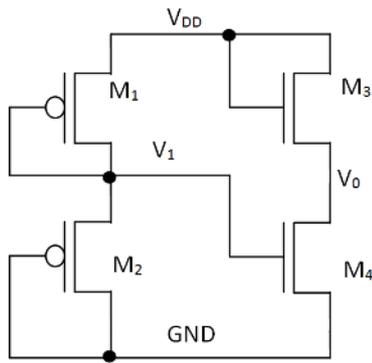


Fig. 2. Proposed configuration for the embedded CMOS dosimeter

By analyzing M1 and M2 it can be stated that:

$$V_1 = (\beta_2(V_{DD} - V_{TP}) + V_{TP}) / (1 + \beta_2) \quad (2)$$

In a similar way the second branch is analyzed, with the following result:

$$V_0 = V_{DD} [1 - \beta_1 \beta_2 / (1 + \beta_2)] - V_{TN} (1 - \beta_1) + V_{TP} (\beta_2 - 1) \beta_1 / (1 + \beta_2) \quad (3)$$

Where:

$$\beta_1 = [(W/L)_4 / (W/L)_3]^{1/2} \text{ and } \beta_2 = [(W/L)_1 / (W/L)_2]^{1/2}$$

As it can be seen from eq. (3), V_0 has three terms, one depending on V_{DD} , and the others related to V_{TN} and V_{TP} . As it is well known, threshold voltages temperature coefficients varies in opposite directions [3]; adjusting the factors related to V_{TN} and V_{TP} compensation is possible achieving, to the first order, a temperature independence. For doing this, the following equation must be solved:

$$(\delta V_{TN} / \delta T) (1 - \beta_1) = |(\delta V_{TP} / \delta T)| (\beta_2 - 1) \beta_1 / (1 + \beta_2) \quad (4)$$

With

$$k = (\delta V_{TN} / \delta T) / |(\delta V_{TP} / \delta T)|$$

The following relation can be obtained:

$$k(1 - \beta_1) = (\beta_2 - 1) \beta_1 / (1 + \beta_2) \quad (5)$$

Eq. (5) can be solved for β_1 giving:

$$\beta_1 = k(\beta_2 + 1) / (2\beta_2) \quad (6)$$

Representing graphically eq. (6) gives the curve of Fig. 4.

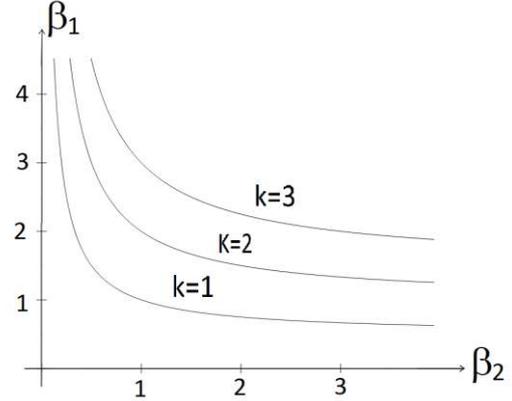


Fig. 3. Graphical representation of eq. (6) for different values of k

Figure 3 or eq. (6) can be used for designing in order to eliminate temperature independence from the dosimeter. In this way, the output voltage will look like:

$$V_0 = V_{DD} [1 - \beta_1 \beta_2 / (1 + \beta_2)] + (V_{TN} - k V_{TP}) (\beta_1 - 1) \quad (7)$$

Equation (7) presents the output voltage including temperature compensation. As it was stated before, shifting of threshold voltages due to radiation is induced in the same direction. This means that, for radiation, there is no cancelling; on the contrary, the effect is additive. In this way, the output voltage will show dependence on radiation, but it will be independent of the temperature variations. The reader may have noticed that the output will also depend on V_{DD} . However, the biasing can be easily kept constant for a sufficiently large period of time to complete the desired measurement.

IV. MEASUREMENTS

In order to have a first approach about the practical behavior of the proposed configuration, a physical implementation was performed using the well-known IC CD4007. Many authors have proposed dosimeters using this circuit. The preferred configuration seems to be the simplest one, with one MOS transistor connected in diode configuration, attached to a constant current source, as in figure 4 [4, 5]. However, this is not temperature compensated and there appear then various proposals to avoid this problem; i.e. some authors have proposed to have a replica not submitted to radiation in order to cancel the temperature effect [6], or other complex mathematical solutions such as in [7].

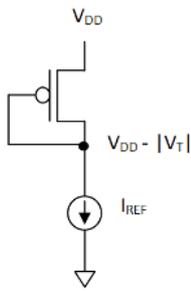


Fig. 4. Constant current VT extractor used commonly for dosimetric applications

The proposed circuit shown in figure 2 is intended to cope with the problem of temperature compensation while giving an output proportional to radiation. It was implemented using CD4007 and submitted to radiation - using an x-ray machine Truebeam-varian®, generously borrowed by the Zunino cancer treatment Institut in Cordoba, Argentina, and temperature variations in a temperature controlled electric oven, showing very good concordance with calculations. The first results for temperature are shown in table I, and radiation measurements are shown in figure 5.

TABLE I
OUTPUT VOLTAGE AS A FUNCTION OF TEMPERATURE

Temperature [°C]	Output voltage [V]
28	2.6140
40	2.6166
50	2.6189
60	2.6222
70	2.6254
80	2.6293
90	2.6333

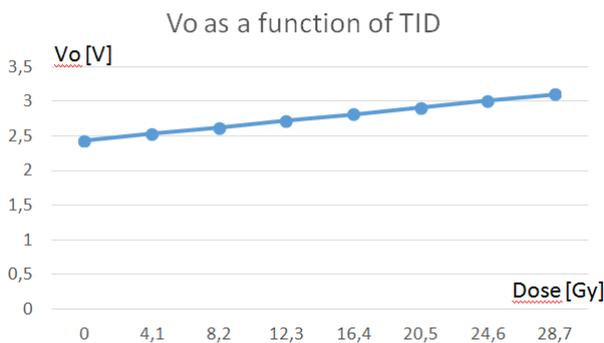


Fig. 5. Output voltage as a function of TID

The measured results are remarkably similar to those found with simulation. Looking at table 1, a calculation between the extremes gives a variation of 16.7mV for a variation of 50 degrees. In percentage, this is less than 1% (0.64%) for 50 degrees variation.

V. CONCLUSIONS

A very simple yet effective CMOS configuration has

been presented for using in dosimeter applications. Its small size and compact structure makes it specially suited for being integrated in embedded applications. Compensation of temperature coefficient is performed and a single output voltage proportional to TID is obtained, using a simple relationship between transistors geometry. The idea was tested using CD4007, showing similar behavior between simulation and laboratory measurements for a wide range of temperatures. As this is only a first step in our research, other measurements have to be made in order to corroborate the first data shown in the paper.

REFERENCES

- [1] Tatjana Pešić-Brđanin, “Spice Modelling of Ionizing radiation effects in CMOS Devices” FACTA UNIVERSITATIS Series: Electronics and Energetics Vol. 30, No 2, June 2017, pp. 161 - 178 DOI: 10.2298/FUEE1702161P
- [2] Y. H. Lho, K. Y. Kim, “Radiation Effects on the Power MOSFET for Space Applications,” ETRI Journal, vol. 27, no. 4, Aug. 2005, pp. 449-452. DOI: 10.4218/etrij.05.0205.0031
- [3] Deepkumar R. Shah, “Radiation Effects on MOSFETs”, Book 2016/05/01 – DOI: 10.13140/RG.2.2.12167.29602
- [4] O. F. Siebel, M. C. Schneider and C. Galup-Montoro, “Low power and low voltage VT extractor circuit and MOSFET radiation dosimeter”, 10th IEEE International NEWCAS Conference, 17-20 June 2012, DOI: 10.1109/NEWCAS.2012.6329016
- [5] M.A. Carvajal, S. Garcia-Pareja, M. Vilches, D. Guirado, M. Anguiano, A.J. Palma, A.M. Lallena, “Simulated and experimental angular response of a commercial MOSFET used as dosimeter”, Proceedings of the 2009 Spanish Conference on Electron Devices - Feb 11-13, 2009. Santiago de Compostela, Spain.
- [6] O.F. Siebel, J.G. Pereira, M.C. Schneider, and C. Galup-Montoro, “A MOSFET dosimeter built on an off-the-shelf component for in vivo radiotherapy applications”, 2014 IEEE 5th Latin American Symposium on Circuits and Systems 25-28 Feb. 2014, DOI: 10.1109/LASCAS.2014.6820261
- [7] M. G. Buehler, B. R. Blaes, G. A. Soli, and G. R. Tardio, “ON-CHIP p-MOSFET DOSIMETRY”, IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 40, NO. 6, DECEMBER 1993

Implementación de técnicas SPWM en FPGA de código abierto

Benjamín Chavarría Domínguez
Departamento de Ingeniería
Electrónica

Tecnológico Nacional de
México/Centro Nacional de
Investigación y Desarrollo Tecnológico
Cuernavaca, Morelos; México.
d18ce092@cenidet.tecnm.mx

Jesus Aguayo Alquicira
Departamento de Ingeniería
Electrónica

Tecnológico Nacional de
México/Centro Nacional de
Investigación y Desarrollo Tecnológico
Cuernavaca, Morelos; México.
ORCID: 0000-0002-0567-0033

Susana Estefany De León Aldaco
Departamento de Ingeniería
Electrónica

Tecnológico Nacional de
México/Centro Nacional de
Investigación y Desarrollo Tecnológico
Cuernavaca, Morelos; México.
ORCID: 0000-0002-1189-4228

Resumen— Ningún inversor de CD-CA es capaz de replicar una señal senoidal perfecta, por tal motivo, es necesario cumplir un conjunto de criterios para poder entregar señales bajo estándares de: menor distorsión armónica, menores pérdidas por conmutación, mayor procesamiento de potencia y máximo aprovechamiento del bus de CD. Una topología recomendada para cumplir con estos criterios es el inversor multinivel de puentes H o CHB-MLI, esta topología requiere de una gran cantidad de pulsos de comando para su correcta ejecución.

El objetivo de este trabajo es presentar una metodología para la generación y programación de los pulsos de comando de un CHB-MLI: Se explica el método de discretización utilizado, sus parámetros y criterios a tener en cuenta. Desde el programa Matlab-Simulink se desarrolla en un esquema de bloques las técnicas de modulación SPWM para generar los pulsos de comando. Se muestra la programación llevada a cabo dentro del entorno Icestudio para integrar los pulsos de comando y generarlos de forma física desde una FPGA. El valor de este trabajo radica en la descripción de los procedimientos necesarios para determinar adecuadamente las características de los pulsos de comando y su integración en el FPGA.

Palabras clave— FPGA, Inversores multinivel, Modulación, SPWM.

I. INTRODUCCIÓN

En el rubro del procesamiento de energía eléctrica uno de los dispositivos que resulta de mayor interés es el inversor. Este dispositivo se encarga de convertir corriente directa a corriente alterna. Se han propuesto diferentes topologías de inversores, las más importantes son las siguientes: el inversor de puente H, el inversor de medio puente H, el inversor de capacitor flotante y el inversor de diodos enclavados [1].

Debido a sus prestaciones, los inversores de puentes H son una de las topologías más empleadas. Existen variantes para cada aplicación, por ejemplo, para motores de tres fases se utilizan inversores de puente H trifásicos, mientras que para reducir el contenido armónico se usan los CHB-MLI (del inglés: Cascaded H-Bridge Multilevel Inverter) [2].

Uno de los inconvenientes que presentan los CHB-MLI es que para controlar la conversión de corriente efectuada en sus múltiples celdas de puentes H, requiere de una gran cantidad de pulsos de comando fiables y sin tiempos de retardo. Un método recomendado para la generación de estos pulsos de comando es mediante el uso de técnicas de modulación basadas en múltiples portadores, por ejemplo, la técnica SPWM (del inglés: Sinusoidal Pulse Width Modulation) [3].

Este trabajo aborda una metodología para el ajuste de parámetros y la programación de los pulsos de comando en

una FPGA (del inglés: Field-Programmable Gate Array). Estos dispositivos digitales poseen la capacidad de efectuar con mucha precisión la ejecución de múltiples trenes de pulsos sin que se presenten retardos o desfases [4].

Existen plataformas como la tarjeta NEXYS 3 con una FPGA Spartan 6 de la empresa Xilinx, o la tarjeta DEO Nano con una FPGA Cyclone IV de la empresa Altera. Los fabricantes proveen versiones gratuitas del IDE, pero existen funciones reservadas a versiones sujetas a licencia.

Por tal motivo, Se empleó una tarjeta de desarrollo de código abierto denominada TinyFPGA BX que cuenta con una FPGA iCE40 de la empresa Lattice [5], [6]. Esta FPGA puede ser programada mediante software libre, como es el caso de los entornos de desarrollo Atom e Icestudio [7].

II. MÉTODOS

A continuación, se describe el procedimiento utilizado para la generación de los pulsos de comando, su discretización y programación dentro de la FPGA.

A. Métodos de discretización de señales y pulsos de comando

Debido a la naturaleza analógica de las técnicas de modulación SPWM, es necesario recurrir a estrategias de discretización que permitan el desarrollo del generador de pulsos de comando dentro de la FPGA, que es un dispositivo digital. Se pueden considerar dos métodos principales para realizar esta discretización.

- Discretización de las señales analógicas SPWM.

Se realiza una simulación de la técnica SPWM para discretizar sus señales portadoras y moduladora [8], [9]. Estas señales son muestreadas a partir de dos criterios: El primero asigna un determinado número de muestras al periodo de la señal muestreada. El segundo criterio consiste en representar el valor de amplitud de la señal mediante palabras binarias (representación de un número decimal en sistema binario), se asigna una palabra binaria a cada muestra del periodo.

- Discretización directa de los pulsos de comando.

Nuevamente se realiza una simulación de la técnica de modulación SPWM, pero ahora se discretizan los pulsos de comando resultantes de la modulación [10], [11]. El periodo de los pulsos es muestreado de igual manera que en método anterior. Mientras que la amplitud de los pulsos es representada mediante un bit (estados alto o bajo). Dicho de otro modo, solo se utiliza una palabra binaria extendida a lo largo de todo el tiempo que duran los pulsos de comando.

La discretización directa de los pulsos de comando emplea menos componentes lógicos de una FPGA en comparación con los recursos utilizados por una discretización de señales analógicas SPWM. Se eligió el método de discretización directa por su sencillez y por qué su proceso de sintetizado e implementación en la FPGA es más rápido.

B. Parámetros de simulación, discretización y divisor de frecuencia

Los datos listados en la tabla 1 se toman como base para efectuar los cálculos y ajustes necesarios en las simulaciones, discretización y programación de los pulsos de comando.

TABLA 1. CARACTERÍSTICAS DE LA FPGA Y PARÁMETROS DE LAS MODULACIONES SPWM.

Características del FPGA	Parámetros de las modulaciones
- FPGA Lattice, modelo ICE40 LP/HX1K.	- Moduladora de 50 Hz y 6 Vpp.
- Compatibilidad con el IDE de Icestudio y lenguaje HDL Verilog.	- 6 portadoras de 2.5 kHz y 1 Vpp para las técnicas LS-PWM, PWM Rotativo y PWM Distribuido.
- Sistema embebido basado en la tarjeta TinyFPGA BX.	- 6 portadoras de 2.5 kHz y 6 Vpp para la técnica PS-PWM.
- Reloj oscilador de 16 MHz.	- Índice de modulación en amplitud de 1 e índice de modulación en frecuencia de 50.
- 7,864 LUTs y 128 kb de RAM.	- 1,000 muestras por ciclo.
- 31 salidas o entradas digitales.	- Simulación de 20 ms para un ciclo de moduladora (equivalente a 50 Hz).

- Verificación de los valores de muestreo

Para integrar los pulsos de comando a la FPGA, es necesario obtener de cada simulación seis vectores de datos correspondientes a los seis pulsos de comando. Cada vector de datos almacena mil muestras del respectivo pulso de comando. Una correcta discretización de estos pulsos, considera que la división del periodo por ciclo de los pulsos entre el número de muestras para cada vector no dé como resultado un número irracional. Esta operación se denomina “Factor de no irracionalidad”. La ecuación 1 efectúa esta comprobación con el periodo de 20 ms y las mil muestras de los vectores de datos.

$$F_r = \frac{t_p}{N_m} = \frac{20 \text{ ms}}{1,000 \text{ muestras}} = 20 \mu\text{s/muestra} \quad (1)$$

Donde F_r es el Factor de no irracionalidad, t_p es el Periodo por ciclo de los pulsos de comando y N_m corresponde al Número de muestras de los vectores de datos.

También se verifica que la división del valor de frecuencia del oscilador entre el número de muestras de cada vector de datos resulte en un valor no irracional y submúltiplo de la frecuencia del oscilador, se asigna el término “Factor submúltiplo” para esta operación. La ecuación 2 corrobora esta operación considerando al reloj oscilador de 16 MHz de la tarjeta TinyFPGA BX y las mil muestras de cada vector.

$$F_s = \frac{f_o}{N_m} = \frac{16 \text{ MHz}}{1,000 \text{ muestras}} = 16 \text{ kHz/muestra} \quad (2)$$

Donde F_s es el Factor submúltiplo, f_o es la Frecuencia del oscilador en la tarjeta de desarrollo y N_m es el Número de muestras de los vectores de datos [12].

- Cálculo del divisor de frecuencia

Es necesario que los pulsos de comando generados por la FPGA tengan una frecuencia por ciclo de 50 Hz, esto se consigue al desarrollar una etapa conocida como “Divisor de frecuencia”. Esta etapa adapta la frecuencia del reloj oscilador principal a un valor de frecuencia adecuado, que permite a los vectores de datos integrados en la FPGA generar los pulsos de comando a la frecuencia deseada.

Para desarrollar el divisor de frecuencia son necesarios dos cálculos: El primero es denominado “Valor del divisor de frecuencia” y se obtiene al multiplicar la frecuencia por ciclo de los pulsos de comando a la salida del FPGA con el número de muestras de los vectores de datos. La ecuación 3 realiza esta operación considerando la frecuencia de 50 Hz y las mil muestras de los vectores de datos.

$$V_d = f_c * N_m = 50 \text{ Hz} * 1,000 \text{ muestras} = 50 \text{ kHz} \quad (3)$$

Donde V_d es el Valor del divisor de frecuencia, f_c es la Frecuencia por ciclo para los pulsos de comando a la salida del FPGA y N_m es el Número de muestras.

El segundo cálculo denominado “Factor del divisor de frecuencia”, es el número empleado en la sección del código HDL llamada divisor de frecuencia. Se obtiene mediante la división de la frecuencia del oscilador montado en la FPGA sobre el valor del divisor de frecuencia obtenido previamente. La ecuación 4 realiza esta operación considerando los 16 MHz del reloj montado en la tarjeta de desarrollo y el resultado obtenido en la ecuación 3.

$$F_d = \frac{f_o}{V_d} = \frac{16 \text{ MHz}}{50 \text{ kHz}} = 320 \quad (4)$$

Dónde F_d es el Factor del divisor de frecuencia, f_o es la Frecuencia del oscilador en la tarjeta de desarrollo y V_d es el Valor del divisor de frecuencia [13].

C. Simulación y discretización de los pulsos de comando

Las simulaciones se desarrollaron en el entorno de bloques Matlab-Simulink. Como se expuso en la tabla 1, para las simulaciones se consideró un tiempo de 20 ms con mil muestras. Se empleó una señal moduladora senoidal de 50 Hz y amplitud de 6 Vpp. Y dos grupos de seis señales portadoras triangulares de 2.5 KHz: el primer grupo con amplitud de 1 Vpp (para las modulaciones LS-PWM, PWM Rotativo y PWM Distribuido) y el segundo grupo con amplitud de 6 Vpp (para la modulación PS-PWM). La Fig. 1 muestra los bloques empleados para la generación de las técnicas de modulación.

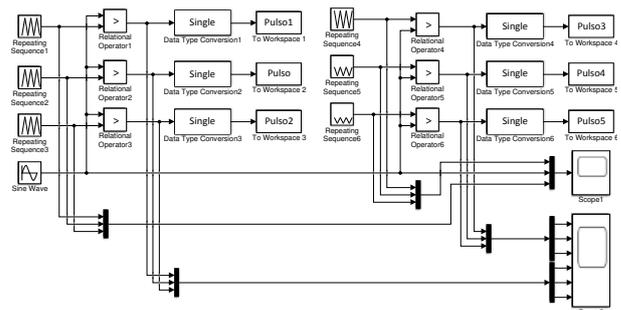


Fig. 1 Bloques usados en Simulink.

D. Tratamiento y programación de los pulsos de comando discretizados

De cada simulación efectuada en el esquema de bloques de la Fig. 1, se obtuvieron seis vectores de datos con la información de los pulsos de comando discretizados gracias a los bloques “Data Type Conversion” y “To Workspace”. Dentro de la ventana “Workspace” de Matlab cada vector de datos se representa como una tabla de mil celdas conformada por una columna que almacena el valor binario de los pulsos de comando (estado alto o bajo).

El entorno de programación de Icestudio permite trabajar en una interfaz modular e híbrida que combina bloques de código escrito en lenguaje HDL tipo Verilog [14], en conjunto con simbología de componentes digitales. La información de los vectores de datos es transferida del entorno de Matlab al entorno de Icestudio. En este se desarrolla un esquema con diferentes bloques de código que conforman al generador de pulsos de comando, presentado la Fig. 2.

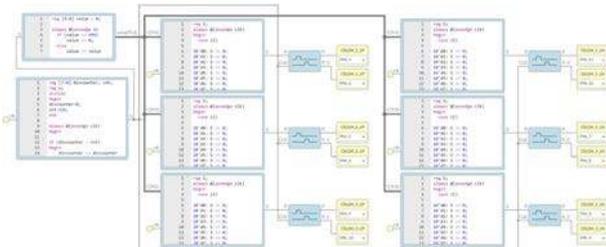


Fig. 2 Esquema de programación dentro del entorno Icestudio.

Las principales secciones del esquema son: El divisor de frecuencia que ajusta los 16 MHz del oscilador principal. El contador de 10 bits que genera mil pasos durante un periodo de 20 ms. Y las memorias de instrucciones que almacenan la información de los vectores de datos (los pulsos de comando).

III. RESULTADOS

El porcentaje de recursos utilizados de la FPGA iCE40 al sintetizar el esquema de la Fig. 2, es el siguiente:

- FFs: 0.16%.
- LUTs: 6.22%.
- PIOs: 7.76%.
- PLBs: 7.91%.

Una vez que los pulsos de comando fueron implementados en la FPGA, se verificó su correcta operación. Con el analizador digital del osciloscopio Agilent modelo DSO-X 3054A se midieron los pines de la tarjeta de desarrollo asignados para la emisión de los pulsos. A continuación, de la Fig. 3 a la Fig. 6 se presentan las lecturas obtenidas.

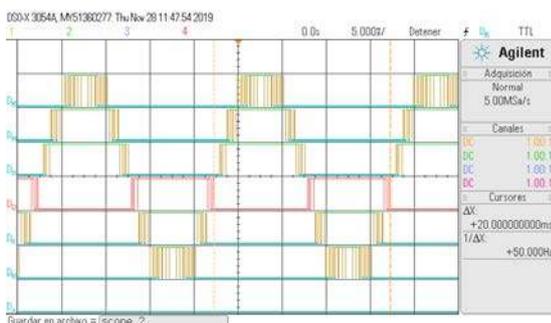


Fig. 3 Pulsos resultantes de la modulación SPWM convencional.

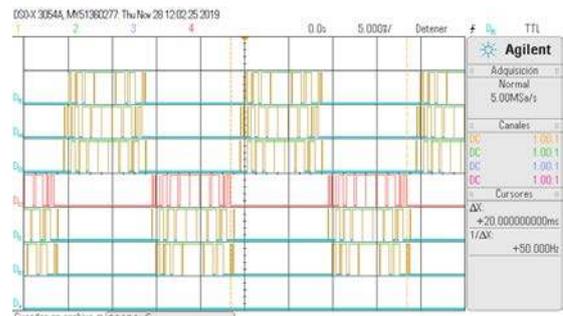


Fig. 4 Pulsos resultantes de la modulación PWM Rotativo.

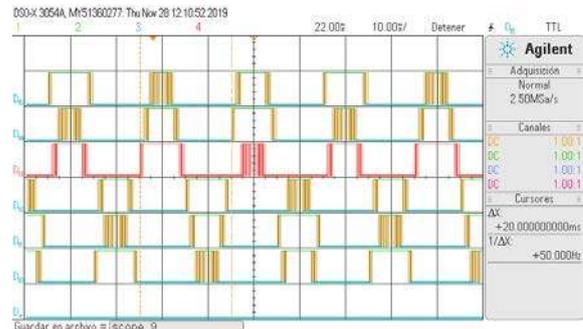


Fig. 5 Pulsos resultantes de la modulación PWM Distribuido.



Fig. 6 Pulsos resultantes de la modulación PS-PWM

Empleando los cursores del osciloscopio se corroboró que los pulsos de comando resultantes tienen un periodo de 20 ms (equivalente a una frecuencia de 50 Hz). Dicho periodo se encuentra en concordancia con los valores definidos en la tabla 1. Cabe destacar que se visualizaron los pulsos de conmutación sin sus contrapartes complementarias.

IV. DISCUSIÓN

Antes de elegir un método de discretización, es necesario tomar en cuenta las siguientes consideraciones para cada uno de las propuestas planteadas:

La técnica de discretización de las señales analógicas SPWM permite desarrollar un circuito digital capaz de variar la configuración de la modulación. Brindando la posibilidad de modificar los pulsos resultantes, una función que sería de gran utilidad para sistemas de control de lazo cerrado; debido a que esta característica brinda la capacidad de reajustar y modificar la respuesta a la salida del CHB-MLI en función de lo que ordene la acción de control del lazo cerrado.

Es necesario que la discretización de las señales SPWM posea una gran cantidad de muestras (y por tanto un mayor uso de componentes lógicos dentro de la FPGA) que le

permitan tener una representación cercana a su forma analógica. De otro modo los pulsos de comando generados por esta discretización no corresponderán a su equivalente analógico, lo que puede ocasionar la generación de una señal deficiente con repercusiones (como un mayor contenido armónico) a la salida del CHB-MLI.

La discretización directa de los pulsos de comando solo requiere de una palabra binaria para representar los estados altos y bajos de los pulsos de comando, el número de componentes lógicos utilizados en su implementación dentro de una FPGA es menor que los componentes necesarios para la discretización de señales analógicas. Sin embargo, al guardar directamente los pulsos generados por una configuración fija de la modulación SPWM no pueden ser usados por sistemas de control de lazo cerrado, ya que solo están capacitados para entregar una forma de onda cuasi-senoidal correcta a la salida del CHB-MLI.

El desarrollo de un código en lenguaje HDL con el primer método es más voluminoso, complejo y el compilador requiere asimilar una mayor cantidad de información. El proceso de sintetizado llega a tardar más de una hora y dependiendo del código se utilizan en promedio los siguientes recursos de la FPGA: 1.27% de sus FFS, 33.19% de sus LUTs, 10.67% de sus PIOs y 40% de sus PLBs.

Debido a que este trabajo de investigación, no requería de la aplicación de un lazo cerrado, se optó por usar el método de discretización directa de los pulsos de comando.

V. CONCLUSIONES

La discretización de señales debe ser lo más cercano a las formas de onda originales para evitar distorsiones. Es necesario seleccionar adecuadamente el número de muestras que conformaran a cada señal discretizada, estas deben ser múltiplos entre sí. También se debe cuidar el redondeo de valores decimales al discretizar.

Al definir las frecuencias de las señales discretizadas dentro del FPGA, se tiene que prever que los valores empleados por los divisores de frecuencias (requeridos por las señales discretizadas) sean submúltiplos exactos del valor del reloj principal.

Es importante considerar la extensión de bits que será asignada a las palabras binarias donde se guardan las señales discretizadas, ya que impactarán directamente en el número de componentes que serán empleados al implementar el código final dentro del FPGA

REFERENCIAS

- [1] J. Rodriguez, Jih-Sheng Lai, and Fang Zheng Peng, "Multilevel inverters: a survey of topologies, controls, and applications," *IEEE Trans. Ind. Electron.*, vol. 49, no. 4, pp. 724–738, 2002.
- [2] L. G. Franquelo, J. Rodriguez, J. I. Leon, S. Kouro, R. Portillo, and M. A. M. Prats, "The age of multilevel converters arrives," *IEEE Ind. Electron. Mag.*, vol. 2, no. 2, pp. 28–39, 2008.
- [3] S. Sankarakumar et al., "Performance Analysis of Multicarrier Sine PWM Based Cascaded H-Bridge Multi Level Inverter," in *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, May 2018, pp. 1018–1023.
- [4] N. M. Salgado-Herrera, A. Medina-Ríos, A. Ramos-Paz and J. R. Rodríguez-Rodríguez, "Generation of a multilevel SPWM technique of 3, 9 and 21 levels with FPGAs," *2013 North American Power Symposium (NAPS)*, 2013, pp. 1-5.
- [5] Valenty, L., *TinyFPGA BX User Guide*. 2015.
- [6] Lattice Semiconductor. (2018). *iCE40 LP/HX Family Data Sheet*.
- [7] J. Barreto, "Inversor Trifásico de Baja Potencia para Aplicaciones de Tracción, CENIDET, Cuernavaca, Morelos, 2020.
- [8] H. Hussin, A. Saparon, M. Muhamad and M. D. Risin, "Sinusoidal Pulse Width Modulation (SPWM) Design and Implementation by Focusing on Reducing Harmonic Content," *2010 Fourth Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation*, 2010, pp. 620-623.
- [9] M. S. N. Romli, Z. Idris, A. Saparon and M. K. Hamzah, "An area-efficient Sinusoidal Pulse Width Modulation (SPWM) technique for Single Phase Matrix Converter (SPMC)," *2008 3rd IEEE Conference on Industrial Electronics and Applications*, 2008, pp. 1163-1168.
- [10] A. B. Afarulrazi, M. Zarafi, W. M. Utomo and A. Zar, "FPGA implementation of Unipolar SPWM for single phase inverter," *2010 International Conference on Computer Applications and Industrial Electronics*, 2010, pp. 671-676.
- [11] M. S. Sur, S. N. A. Singh and P. Kumari, "Simulation and generation of SPWM waveform using VHDL for FPGA interfaced H bridge power inverter," *2010 International Conference on Industrial Electronics, Control and Robotics*, 2010, pp. 39-42.
- [12] J. Quintero, M. Sanz, A. Barrado and A. Lazaro, "FPGA based Digital Control with High-Resolution Synchronous DPWM and High-Speed Embedded A/D Converter," *2009 Twenty-Fourth Annual IEEE Applied Power Electronics Conference and Exposition*, 2009, pp. 1360-1366.
- [13] De Oro Arenas, G. de Azevedo e Melo and C. A. Canesin, "A Methodology for Power Quantities Calculation Applied to an FPGA-Based Smart-Energy Meter," in *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-11, 2021.
- [14] LaMeres, B. J., *Introduction to Logic Circuits & Logic Design with Verilog*. Springer International Publishing. 2019.

Assessment of image resolution reduction on position estimation in visual odometry

Carlos Fernandez
GridTICs

National University of Technology,
Mendoza, Argentina,
Email: cfernandez@inaut.unsj.edu.ar

Rodrigo Gonzalez
GridTICs

National University of Technology,
Mendoza, Argentina,
Email: rodriguez@frm.utn.edu.ar

Héctor Daniel Patiño
Instituto de Automática

Universidad Nacional de San Juan
Email: dpatino@inaut.unsj.edu.ar

Abstract—In this work, we evaluate how the performance of a VO system changes when the resolution of images are decreased. We have used a very common technique in VO. We detect the characteristic points of successive images such as corners, lines, regions or edges and we look for correspondences of these points with the points detected in the previous images. We decrease the resolution of these images to establish a comparison between the original resolution of the images and the images of lower resolution. We find that RMSE increases when resolution is decreased when applying in a VO system.

Index Terms—Visual Odometry, Embedded Systems, FPS, ORB

I. INTRODUCTION

In recent years, visual odometry (VO) has gained importance in robotics applications, such as ground vehicles moving over rough terrain or unmanned aerial vehicles (UAVs). VO is a process for estimating the position of a body using only the input from one or more cameras attached to it. VO operates by gradually estimating the position of the vehicle by analyzing the variations that motion induces in the frames of its onboard cameras. Compared to the classical odometry of wheeled vehicles, the VO has the advantage that it is not affected by wheel slippage on uneven terrain and other kinds of disturbances due to adverse conditions. In environments where there is no access to GPS measurements (underwater environments, UAVs in indoor spaces) VO becomes a relevant option.

VO systems use the classical SFM (Structure From Motion) principles where a 3D points cloud of the environment are obtained by triangulation along the camera's trajectory, thus making it possible to simultaneously estimate both the camera movement and the depth of the points that have been used for tracking [1].

[2] describes that the methods of VO can be classified in different ways:

- 1) Depending on the number of cameras used, the VO can be Monocular (1 camera) or Stereo (2 cameras).
- 2) Because of the type of sensors used, the VO can be either only visual or integrated with

supplementary information from sensors such as lasers or IMUs, among others.

- 3) Because of the way the relationship between successive images is obtained, the VO can be based on the extraction and matching of points using matching methods as SURF, SIFT, FAST or by point tracking with Optical Flow (OF).

VO has been a breakthrough in robotics. However, its use in resource-constrained applications, such as small-scale mobile robotic platforms, is limited due to the high computational complexity involved. About the image resolutions used to run the VO algorithm, there are some studies in which the image resolution is reduced to achieve a fast execution of the algorithm, depending on whether it will be run on a Raspberry, Arduino or FPGAs. For example, [3] in 2013 presented PX4FLOW, an ARM Cortex M4 microcontroller based sensor system that performs optical flow processing at 250 fps at a subsampled resolution of 64×64 pixels using a CMOS machine vision sensor. In 2016,[4] proposed a method to evaluate the performance of a VO algorithm using an industrial robot, the parameters necessary for the comparison are estimated by reducing the problem to a Constrained Quadratic Program. The method is applied to an experimental setup consisting of an industrial robot with six degrees of freedom and a commercial high-definition monocular camera capturing images with a resolution of 1280×720 . In 2019 [5] presented a paper aiming to propose and compare a fast monocular approach, called ARM-VO, with two state-of-the-art algorithms, LibViso2 and ORB-SLAM2, on Raspberry Pi 3. The approach is a sequential frame-by-frame scheme that extracts a sparse set of well-distributed features and tracks them in the next frames using the Kanade-Lucas-Tomasi tracker, the method was implemented in ROS using the KITTI dataset with images of 1226×370 resolution. [6] in 2021 presented a paper where benchmark tests of several Visual Inertial Odometry (VIO) algorithms are performed on NVIDIA Jetson platforms using 640×480 resolution images obtained from the Intel Realsense D435i camera. The compared algorithms include mono and stereo, covering VO and VIO: VINS-Mono, VINS-Fusion, Kimera, ALVIO,

Stereo- MSCKF, ORB-SLAM2 stereo and ROVIO. The evaluation is performed in terms of estimated odometry accuracy, CPU usage and memory usage on various Jetson boards.

In this work we performed a series of experiments using an ORB feature detector (configured to detect 6000 features) to test the performance of a VO algorithm using high to low resolution images. If in our experiments the algorithm has a bad performance, we could say that the better resolution (and also brightness) has the set of images of our dataset, the algorithm will work much better, it would also allow us to say that the higher the resolution of these images will probably be complicated the execution of the VO algorithm in embedded systems as Raspberry and Arduino. So, it would be advisable to make use of FPGAs, which have been of great help for processing images at large scales.

II. EXPERIMENT

The VO algorithm used is based on the feature extraction method, which extracts feature points from the image and tracks them in the image sequence and has been evaluated on the KITTI dataset, which has become the most popular benchmark for optical flow, VO and VSLAM. It includes rural landscapes, road sequences, provides grayscale images, and color images. The dataset contains more than 4000 grayscale images. Also comprises the following information, captured and synchronized at 10 Hz:

- 1) 3D Velodyne point clouds, 100k points per frame, stored as binary float matrix.
- 2) 3D GPS/IMU data: location, speed, acceleration, meta information, stored as text file.
- 3) Calibration: camera, camera-to-GPS/IMU, camera-to-velodyne, stored as text file.

Three different trajectories have been selected from the KITTI dataset: the first trajectory is a straight line, the second is a right turn and the last one is a left turn. Each trajectory contains a maximum of 200 images to achieve fast algorithm execution. In addition, the resolution of each image is gradually reduced. Then, the ground-truth trajectories and those estimated by the VO algorithm are compared to see how the algorithm performs. The programming language used is Python version 3.9, we have used the Anaconda package and the OpenCV library, which is an open source library that contains implementations that cover many algorithms for computer vision.

III. RESULTS

The Root Mean Square Error (RMSE) metric is used to measure the difference between ground-truth and estimated points, given two sets of errors δx and δy , the $2RMSE$ is calculated through the following equation:

$$\delta_{2RMSE} = 2\sqrt{\frac{(\delta x_1^2 + \delta y_1^2) + (\delta x_2^2 + \delta y_2^2) + \dots + (\delta x_n^2 + \delta y_n^2)}{n}}$$

First, we test the VO algorithm using the 1226×370 original resolution of the image provided by the KITTI dataset. We observe in the Fig. 1a that the VO algorithm is quite efficient as it manages to be close to the ground-truth path. The $2RMSE$ value is 16.556 m.

Next, image resolution is reduced to 800×241 . It can be observed in Fig. 1b that the algorithm is less accurate. There is a large deviation from the prediction. The $2RMSE$ value is 30.771 m.

Finally, we reduced the resolution to 640×193 . Results for this image setting are quite similar to the second one, as we can see in Fig. 1c. The prediction of the algorithm continues to degrade, as it has deviated more than the previous setup. Thus, the error continues to increase and the $2RMSE$ value is 37.837 m.

In the second experiment, we have a path with a certain left turn. First, we take the original images from the KITTI dataset. As can be seen in Fig. 2a, the VO algorithm presents errors however it manages to get close to the truth path. The $2RMSE$ value is 17.543 m.

Next, the resolution of the image is reduced to 800×241 to observe the behavior of the VO algorithm with respect to the ground-truth. As can be seen in Fig. 2b, although the algorithm performs well in following the real path, the error has increased with respect to the error obtained when using the original image. The $2RMSE$ value is 24.849 m.

Finally, images are reduced for once again to a scale of 640×193 . As can be seen in Fig. 2c, the algorithm continues to draw a path similar to the one obtained in the original image but the error continues to increase. The $2RMSE$ value is 32.657 m.

The last experiment was performed with a trajectory that has a right turn. We repeated the steps mentioned above to verify the operation of the algorithm. The Fig 3a shows the original image, the $2RMSE$ value of the original image is 15.489 m; in Fig 3b images have a resolution of 800×241 , the $2RMSE$ is 35.944 m; and lastly in Fig 3c images have a resolution of 640×193 and the $2RMSE$ is 48.957 m.

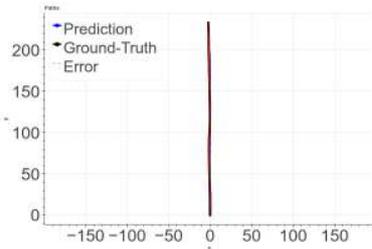
Table I shows a summary containing the three trajectories with the resolutions used in the experiments and their respective $2RMSE$ values.

TABLE I
COMPARISON AMONG THE THREE TRAJECTORIES WITH
DIFFERENT RESOLUTIONS AND $2RMSE$ RESULTS.

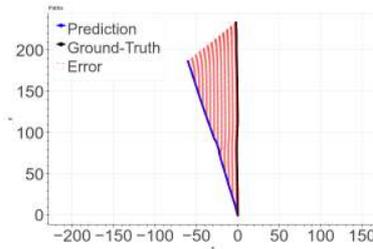
Trajectory 1		Trajectory 2		Trajectory 3	
Resolution	$2RMSE$	Resolution	$2RMSE$	Resolution	$2RMSE$
1226x370	16.556 m	1226x370	17.543 m	1226x370	15.489 m
800x241	30.771 m	800x241	24.849 m	800x241	35.944 m
640x193	37.837 m	640x193	32.657 m	640x193	48.957 m

IV. CONCLUSION

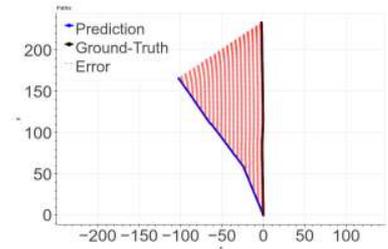
In this work we proposed a series of experiments based on reducing the image resolution provided by



a) Original resolution (1226x370)

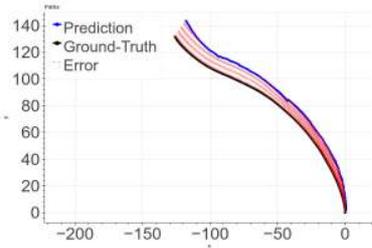


b) 800x241 resolution

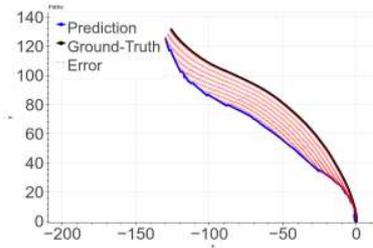


c) 640x193 resolution

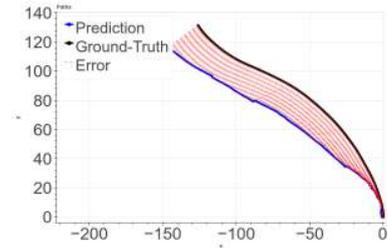
Fig. 1. First path



a) Original resolution (1226x370)

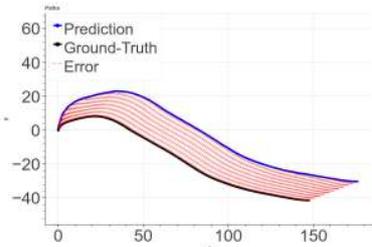


b) 800x241 resolution

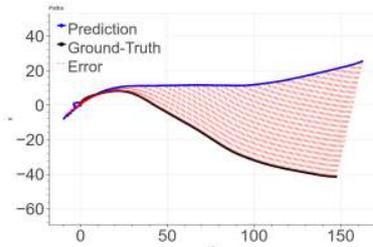


c) 640x193 resolution

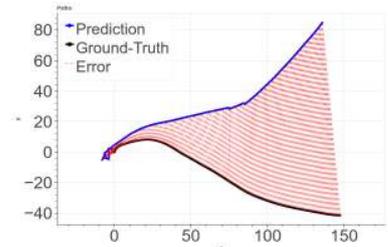
Fig. 2. Second path



a) Original resolution (1226x370)



b) 800x241 resolution



c) 640x193 resolution

Fig. 3. Third path

the KITTI dataset. The decision to reduce the images is based on testing the performance of the VO algorithm when these images suffer a resolution reduction. We have exposed the relevance of using the best possible image resolution when it comes to implementing a VO system. As shown in Table I, a resolution reduction of 35% (from 1223x370 to 800x241) can lead, on average, to an increase of 220% in 2RMSE, or in other words, to a x2 in performance degradation. Since most microprocessor-based embedded systems cannot cope to process images with high resolutions as shown in Section I, FPGA emerge as a suitable platform to implement VO systems.

REFERENCES

- [1] Wilbert Geovanny. “Reconocimiento y seguimiento de plataformas para el aterrizaje automático de un vehículo aéreo no tripulado basado en inteligencia artificial y odometría visual”. PhD thesis. 2020. ISBN: 9789896540821.
- [2] Davide Scaramuzza and Friedrich Fraundorfer. “Tutorial: Visual odometry”. In: *IEEE Robotics and Automation Magazine* 18.4 (2011), pp. 80–92. ISSN: 10709932.
- [3] Dominik Honegger et al. *An Open Source and Open Hardware Embedded Metric Optical Flow CMOS Camera for Indoor and Outdoor Applications*. Tech. rep.
- [4] Roberto Rossi, Giulio Melacarne, and Paolo Rocco. “Performance evaluation of visual odometry using an industrial robot as ground truth”. In: *IECON Proceedings (Industrial Electronics Conference)* (2016), pp. 761–766.
- [5] Zana Zakaryaie Nejad and Ali Hosseininaveh Ahmadabadian. “ARM-VO: an efficient monocular visual odometry for ground vehicles on ARM CPUs”. In: *Machine Vision and Applications* 30.6 (2019), pp. 1061–1070. ISSN: 14321769.
- [6] Jinwoo Jeon et al. “Run Your Visual-Inertial Odometry on NVIDIA Jetson: Benchmark Tests on a Micro Aerial Vehicle”. In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 5332–5339. ISSN: 23773766.

Categoría

REPORTE

Implementación de Sistemas Embebidos

Área Temática

Sistema de información visual para pasajeros de trenes argentinos

Carlos German Carreño Romano
Universidad de Buenos Aires
(UBA) Facultad de Ingeniería
Buenos Aires, Argentina
ccarreno@fi.uba.ar

Pablo Martín Gomez
Universidad de Buenos Aires
(UBA) Facultad de Ingeniería
(LSE) Laboratorio de Sistemas Embebidos
Buenos Aires, Argentina
pgomez@fi.uba.ar

Resumen—En los trenes existen distintos sistemas de control que se interconectan por medio de una red de comunicaciones del tren (TCN). Algunos ejemplos son el sistema de frenos, el control de puertas, el aire acondicionado y el sistema de información visual al pasajero (PIDS). El PIDS es el responsable de transmitir mensajes como el destino o la próxima estación usando carteles de matriz led y tiene su propia red. También incluye mapas con indicadores led, un sistema de audio y un circuito cerrado de cámaras. El propósito de este trabajo es desarrollar firmware y hardware necesarios para controlar los carteles de matriz LED de salón (IDU) de las formaciones ferroviarias de Trenes Argentinos. El principal valor que aporta es generar herramientas de mantenimiento para extender la vida útil de los trenes. Si bien existen carteles led comerciales de propósito general, en los trenes hace falta conectarlos a la red, interpretar datos y protocolos del PIDS, presentar mensajes en función de los datos y considerar restricciones eléctricas. En el diseño e implementación del sistema embebido de este trabajo se aplicaron patrones de software concurrente (máquinas de estado y objeto activo) usando un sistema operativo de tiempo real (RTOS) sobre una plataforma EDU-CIAA. También se desarrollaron piezas de hardware para adaptar el sistema a la red y se realizaron capturas de tramas de datos en formaciones ferroviarias operando en vivo.

Palabras clave—TCN, PIDS, RTOS, EDU-CIAA, Trenes Argentinos

I. INTRODUCCIÓN

El uso de carteles de matriz led, además de ser usados en trenes, está extendido a aplicaciones como los sistemas de información en aeropuertos, en paradas de ómnibus, en señalamiento vial y en la industria del entretenimiento. Las dimensiones del cartel, la densidad de píxeles por unidad de área, la cantidad de colores o leds por píxel, los niveles de intensidad lumínica, el brillo y contraste, la potencia eléctrica, son algunas de las especificaciones típicas. Los controladores pueden basarse en circuitos digitales, en microcontroladores de 8, 16 o 32 bits, en FPGA, y existe diversidad en la implementación del software embebido. El envío de datos y el formato de los mismos dependen de la implementación.

En [1] se utiliza el chip AT89C52 para enviar caracteres chinos sobre matrices de 32 x 192 leds de un solo color; en [2] se implementa una pantalla led RGB de 320 x 240 píxeles que rota 360° permitiendo visualizar imágenes en color por

persistencia de visión; en [3] se desarrollan algoritmos sobre FPGA usando búferes de datos para controlar una pantalla LED de 160 x 32 píxeles alcanzando 32,768 colores; en [4] se presenta el control de un micro display de transistores de película delgada (TFT) usando modulación por ancho de pulso (PWM) alcanzando 256 niveles de color a una frecuencia de refresco de 60 Hz, basado también en FPGA; en [5] se presenta el control de píxeles virtuales para matrices led multicolor usando flip-flops tipo D.

En el diseño e implementación del presente trabajo, los carteles son de matriz led de un solo color y de distintas dimensiones (8x64, 32 x 64, 32 x 128). El control de los carteles tiene como factor común el uso del conjunto de chips digitales 74HC138, 74HC595 y 74HC245. La topología permite interconectar paneles en serie para construir carteles led de distinto tamaño usando la misma lógica de control.

A. Introducción específica

El sistema PIDS de las formaciones de Trenes Argentinos (SOFSE) adquiridas en 2015 tiene una arquitectura propietaria que usa una red física RS485 más una serie de protocolos definidos por el fabricante para cada interfaz. Cabe mencionar que desde 2008 [6] la IEC busca desarrollar estándares para TCN usando redes Ethernet de tiempo real. En 2016 se publica IEC62580-2 que incluye al sistema de información al pasajero (ahora llamado PIS), el CCTV y los protocolos asociados en el estándar. Sin embargo, este nuevo estándar no estaría disponible en dispositivos comerciales hasta finales de 2016-2017. El estudio y desarrollo de herramientas de mantenimiento para la red PIDS existente es de gran relevancia para el personal de trenes argentinos, porque permite extender la vida útil de los trenes planificada en no menos de 20 años y además reducir la dependencia tecnológica con los fabricantes. En este trabajo se estudia entonces el sistema PIDS basado en redes RS485 con protocolos.

El propósito de este trabajo es desarrollar el firmware y hardware necesarios para controlar los carteles LED de salón de formaciones ferroviarias de Trenes Argentinos. Los

carteles forman parte del sistema PIDS, una red RS485 que se interconecta a la red de comunicaciones del tren TCN. La red TCN presenta una arquitectura de buses jerárquicos de dos niveles, el bus de datos WTB y el MVB [7]. En la figura 1 se representan los buses de la red TCN, y el sistema embebido propuesto para conectar los carteles LED al sistema.

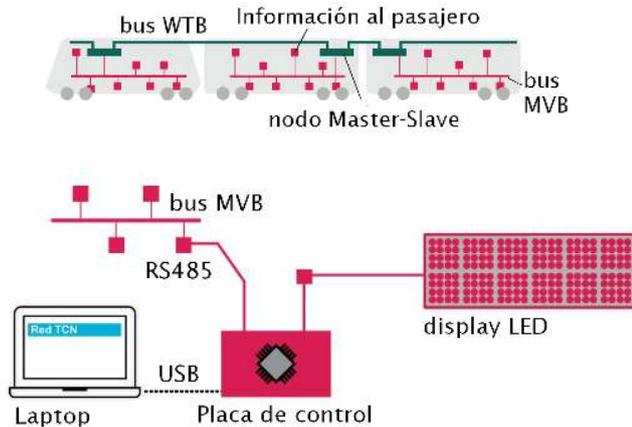


Fig. 1. Arriba: diagrama de los buses de datos WTB y MVB de la red TCN. Abajo: diagrama del sistema embebido propuesto para controlar los carteles LED de salón. El sistema se compone de la placa de control y su firmware, las interfaces con el cartel led, con la red RS485 y con una laptop.

II. DESARROLLO

Para estudiar la comunicación entre estos buses de datos se desarrollaron piezas de hardware y software que permiten capturar paquetes de la red RS485. En la figura 2 se puede observar la conexión realizada en las formaciones de SOFSE para la captura de datos de la red PIDS en operación.

Estas tramas de datos se estudiaron posteriormente logrando identificar patrones. Las mediciones buscan identificar la comunicación entre los carteles de matriz led y el resto del sistema.

A. Diseño e implementación de sistema embebido

El sistema embebido del presente trabajo debe resolver la comunicación entre la red RS485 del sistema PIDS y los carteles LED de salón. La placa de control está basada en la plataforma EDU-CIAA. La misma se puede comunicar con una computadora personal y tiene la función de recibir datos de la red RS485 y presentar información al pasajero a través de los carteles LED.

B. Casos de uso

Los casos de uso requeridos del sistema involucran al tren, al operador o conductor de tren, al sistema y a los pasajeros. Cuando el tren arriba a una estación se debe presentar información visual al pasajero. Si el conductor presiona



Fig. 2. Arriba: Fotografía de las mediciones realizadas sobre la red PIDS. Abajo: Cartel LED de salón inicializándose.

un botón, el sistema debe también presentar información predeterminada al pasajero, por ejemplo las estaciones cabecera del recorrido que se visualizan en los carteles LED de frente y contrafrente del tren. Cierta información, por ejemplo de prevención, se puede presentar al pasajero periódicamente. Este caso de uso implica eventos temporales mientras el tren está en circulación.

Por último, un caso de uso de particular interés es decodificar las tramas de datos que viajan por la red

PIDS para alimentar un sistema externo, por ejemplo un sistema de transmisión de datos del tren a la nube, o una herramienta local de chequeo de datos. Este es un caso de uso en el que el cliente no es un pasajero sino otro sistema.

C. Patrones

Los casos de uso plantean problemas de diseño a resolver que presentan similitudes. En los casos de presentar información visual al pasajero el problema a resolver implica observar variables de estado y generar mensajes de salida en base a algún criterio. Se utiliza para estos casos el patrón observar y reaccionar [13]. En sistemas embebidos también es usual utilizar el patrón máquina de estados finitos (fsm) [9]. Este es un patrón de comportamiento que permite actualizar el estado interno de un objeto como respuesta a entradas. En cada momento la fsm puede estar en un sólo estado posible, y las transiciones entre estados están normalmente asociadas a eventos. En este trabajo se realiza una implementación de fsm utilizando un arreglo de estructuras y handlers usando punteros a función [10].

Dado que el sistema tiene como requerimientos dar respuesta a eventos de forma asíncrona, se hará uso también del patrón objeto activo (AO) [11]. Este es un patrón que permite desacoplar los métodos de invocación y ejecución para mejorar la concurrencia y simplificar el acceso a datos de un objeto que tiene su propio hilo de control. El patrón AO se utiliza en este trabajo usando una implementación en C sobre RTOS que permite a cada tarea ejecutar una máquina de estados en un ciclo infinito [12], estandarizando la interfaz entre tareas por medio de una cola FIFO. Las tareas esperan por nuevos eventos a ser procesados usando esta interfaz, y mientras no reciban eventos se mantienen en estado inactivo. Cuando se recibe un evento, RTOS despierta la tarea y ésta ejecuta una actualización en la máquina de estados asociada.

En este sistema se hará uso del patrón AO para simplificar el acceso a distintas máquinas de estado que deberán funcionar de forma concurrente. La solución que se plantea en este documento hace uso de estos patrones de forma combinada.

D. Firmware

Los datos de entrada de la red RS485 se reciben usando la interfaz UART. Se ha programado esta interfaz usando interrupciones (ISR UART) como se indica en la figura 3(a). La interrupción está configurada para usar un callback (IRQ UART) que otorga un semáforo de RTOS. Este semáforo es tomado siempre por una tarea de RTOS (vTask UART) que se encarga de leer los datos de entrada y encolar mensajes de eventos para actualizar una máquina de estados. Este mecanismo logra que la rutina de interrupción sea muy corta, permitiendo una buena performance en términos de tasa de datos, modularidad y simplicidad. La tarea de leer los

bytes de entrada y de resolver una acción concreta quedan desacopladas usando objeto activo.

En la figura 3 se presentan los diagramas de dos de los objetos activos implementados en el firmware del sistema embebido desarrollado.

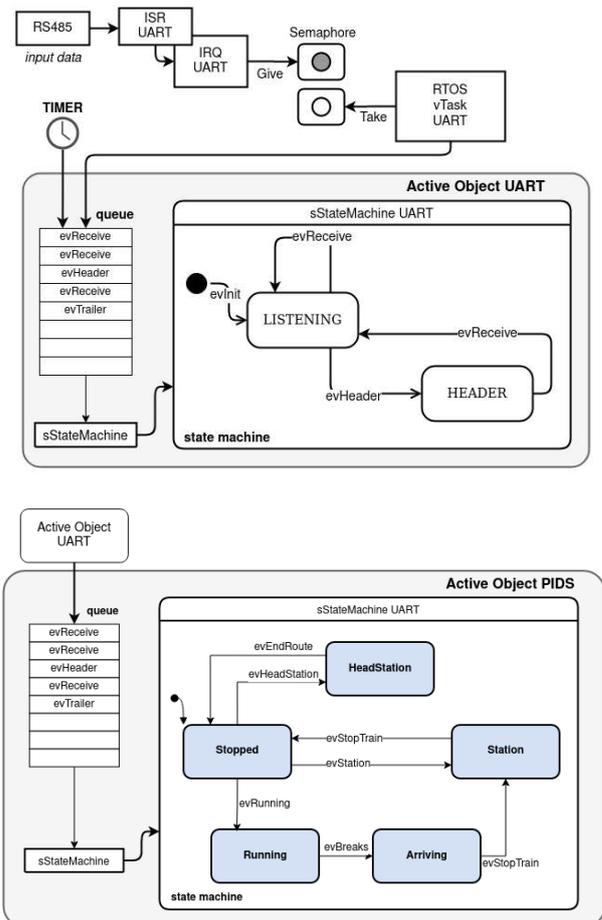


Fig. 3. (a) Arriba: diagrama del objeto activo que maneja la UART y usando RTOS. (b) Abajo: el diagrama del objeto activo que administra el contexto, es decir el estado del tren.

El manejo de la lógica asociada al tren también se resuelve con un objeto activo que administra otra máquina de estados. En particular, esta última se puede ver en la figura 3(b) y representa los estados de circulación del tren, siendo la que guarda relación directa con el contexto. Cuando el tren está en circulación, una serie de sensores envían datos de entrada que serán procesados y se recibirá un mensaje por la cola FIFO. Si el tren frena, la máquina de estados debe pasar al estado detenido (Stopped). En cada uno de estos estados se guarda un mensaje de información visual predeterminado. Estos mensajes se pueden configurar modificando un archivo en el firmware.

E. Pruebas de validación

Para realizar pruebas del sistema se ha montado una maqueta que se pueda observar en la figura 4. Esta maqueta está formada por la EDU-CIAA, un adaptador de conectores de 2x8 pines y la placa del display led. La EDU-CIAA está conectada a una computadora personal, recibiendo datos por la interfaz UART y enviando datos al cartel usando pines de salida GPIO.

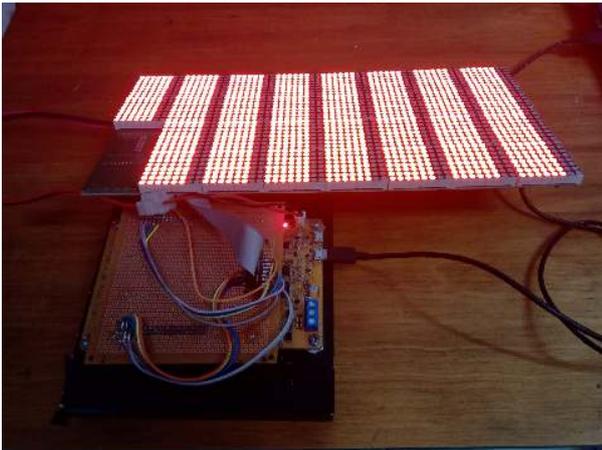


Fig. 4. Fotografía de la maqueta para testear el control del cartel de matriz led usando la plataforma EDU-CIAA y un adaptador.

Las pruebas realizadas permiten validar el firmware desarrollado para controlar carteles de matriz led compatibles con los existentes en las formaciones de SOFSE.

III. CONCLUSIONES

Este trabajo en progreso en colaboración con Trenes Argentinos permite estudiar los protocolos de la red PIDS y la comunicación con los carteles led de salón. Se han desarrollado piezas de hardware y software que funcionan como herramientas de mantenimiento. El control de los carteles de matriz led usando el conjunto de chips descrito es escalable, de bajo costo, y de complejidad media en el manejo de señales digitales. Estos carteles pueden ser candidatos a fabricación en serie. Se ha logrado controlar estos carteles usando la plataforma EDU-CIAA. Se ha observado que por las tramas de datos de la red PIDS no viajan mensajes, sino códigos asociados a dispositivos. Cada dispositivo, como un cartel led, está nombrado con una dirección. Las placas de control que funcionan como interfaz entre la red RS485 y los carteles led de salón se alimentan de entradas de corriente continua de 110 V. Distintos módulos adaptadores de tensión permiten el funcionamiento de un circuito de datos basado en un microcontrolador de 32 bits. Esta observación es la que permite y da sentido a este desarrollo ya que el prototipo desarrollado permitiría reemplazar las placas del fabricante por hardware de fabricación nacional, reduciendo de esta

manera la dependencia tecnológica y acortando las cadenas de suministro.

REFERENCIAS

- [1] SONG, Yongxian, et al. Design of LED Display Control System Based on AT89C52 Single Chip Microcomputer. *J. Comput.*, 2011, vol. 6, no 4, p. 718-724.
- [2] LIU, Shih-Mim; CHEN, Ching-Feng; CHOU, Kuang-Chung. The design and implementation of a low-cost 360-degree color LED display system. *IEEE transactions on consumer electronics*, 2011, vol. 57, no 2, p. 289-296.
- [3] KURDTHONGMEE, W. Design and implementation of an FPGA-based multiple-colour LED display board. *Microprocessors and Microsystems*, 2005, vol. 29, no 7, p. 327-336.
- [4] LIN, Yi-Zhen, et al. Active-matrix micro-LED display driven by metal oxide TFTs using digital PWM method. *IEEE Transactions on Electron Devices*, 2021, vol. 68, no 11, p. 5656-5661.
- [5] GAGO, Alfonso; FERNÁNDEZ, José; BOHÓRQUEZ, Alfonso G. Control architecture of a virtual matrix LED display without current drivers. En *2009 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*. IEEE, 2009. p. 53-56.
- [6] Feng, Jianghua, et al. "Survey of development and application of train communication network." *Proceedings of the 2015 International Conference on Electrical and Information Technologies for Rail Transportation*. Springer, Berlin, Heidelberg, 2016.
- [7] Schifers, C., and Gernot Hans. "IEC 61375-1 and UIC 556-international standards for train communication." *VTC2000-Spring*. 2000 IEEE 51st Vehicular Technology Conference Proceedings (Cat. No. 00CH37026). Vol. 2. IEEE, 2000.
- [8] Sommerville, Ian. "Software Engineering 10." Harlow: Pearson Education Limited (2016).
- [9] Lavender, R. Greg, and Douglas C. Schmidt. "Active object—an object behavioral pattern for concurrent programming." (1995).
- [10] Samek, Miro. *Practical UML statecharts in C/C++: event-driven programming for embedded systems*. CRC Press, 2008.
- [11] <https://aticleworld.com/state-machine-using-c/>, consultado en Junio 2022.
- [12] <https://www.sinelabore.de/doku.php/wiki/howto/rtos>, consultado en Junio 2022.
- [13] *Introduction to Driving LED Matrices, Application Note 1216*, Avago Technologies.

Diseño de un transmisor ADS-B embebido utilizando tecnología de Radio Definida por Software (SDR)

Francisco Martín ESCOBAR
Grupo de Sistemas de Tiempo Real
Universidad Nacional de Río Cuarto
Río Cuarto, Córdoba, Arg.
mescobar@ing.unrc.edu.ar

Damian Hector PRIMO
Grupo de Sistemas de Tiempo Real
Universidad Nacional de Río Cuarto
Río Cuarto, Córdoba, Arg.
dprimo@ing.unrc.edu.ar

Juan Octavio OVIEDO
Centro I+D de Tec. Aeronáuticas
Fuerza Aérea Argentina
Las Higueras, Córdoba, Arg.
joviedo@faa.mil.ar

Juan Pablo RUMIE VITTAR
Centro I+D de Tec. Aeronáuticas
Fuerza Aérea Argentina
Las Higueras, Córdoba, Arg.
jvittar@faa.mil.ar

Dario Walter DIAZ
Grupo de Sistemas de Tiempo Real
Universidad Nacional de Río Cuarto
Río Cuarto, Córdoba, Arg.
ddiaz@ing.unrc.edu.ar

Diego Daniel BADINO
Grupo de Sistemas de Tiempo Real
Universidad Nacional de Río Cuarto
Río Cuarto, Córdoba, Arg.
dbadino@ing.unrc.edu.ar

Resumen— Desarrollo de un prototipo de transmisor ADS-B embebido en una mini-pc y se utiliza un SDR para transmitir en una frecuencia de 1090MHz. El sistema recibe la señal del GNSS (Global Navigation Satellite System) y un sensor de presión barométrica para luego generar el protocolo ADS-B y así transmitirlo con una modulación PPM (Pulse Position Modulation) en 1090MHz. Este sistema embebido sirve para realizar una análisis y estudio del estándar de comunicación inalámbrico que utilizan las aeronaves civiles y militares

Palabras claves—ADS-B out, SDR, GNSS, Data link inalámbrico, Seguridad aérea, Control de tráfico aéreo

I. INTRODUCCIÓN

ADS-B (Automatic Dependent Surveillance - Broadcast)[1], es un sistema de vigilancia aérea que transmite información en forma cooperativa, y de modo automático, ya que emite los mensajes sin intervención humana y es dependiente porque utiliza información brindada por sistemas de geolocalización. A diferencia de otros sistemas de radar, en los sistemas ADS-B no hay necesidad de realizar interrogaciones desde las estaciones terrenas. ADS-B informa periódicamente datos de posición obtenidos por sistemas satelitales de posicionamiento global (Global Navigation Satellite System, GNSS), velocidad, altitud, número de vuelo, estado de la aeronave, entre otros. Esta información emitida en forma de broadcast es recibida por receptores en superficie o por aeronaves dentro del área de vuelo de la aeronave emisora.

La información, que aporta el sistema ADS-B de la aeronave, contiene una identificación, posición, ruta y velocidad, a las estaciones terrestres ADS-B y aeronaves circundantes. Ofrece a los controladores aéreos información de situación en tiempo real de las aeronaves durante su vuelo dando la posibilidad de manejar con mayor seguridad la separación entre aeronaves. En el sistema ADS-B, existen equipos con capacidad "ADS-B IN"[1] en el avión, que permite que la información pueda visualizada por el Piloto en una pantalla o también información que envían las estaciones terrestres de ADS-B, que incluyen METAR(Informe meteorológico aeronáutico de rutina - METeorological Aerodrome Report), NOTAMS (Avisos a los Aviadores - Notice to Airman), información de radar y más, de modo que los "ADS-B IN" muestren toda esta información. Además, los equipos "ADS-B IN" reciben las señales de aviones "ADS-B OUT"[1] sin que haya de por medio una estación terrestre. O sea, permite que haya comunicación de manera directa entre aeronaves, de forma que los aviones se vean entre ellos con

sus posiciones en tiempo real. Para este desarrollo ha sido realizado solamente la etapa de un transponder ADS-B out[1][2] de baja potencia de RF (10mW).

II. OBJETIVOS

A. Implementación en el Hardware

Para el desarrollo, se hizo uso de un sistema embebido compuesto por una placa adquisidora de GNSS y sensores, una Mini-PC que ejecuta todos los procesos de generación del protocolo ADS-B y por último un hardware de SDR (Software Defined Radio)[12] (Fig. 1).

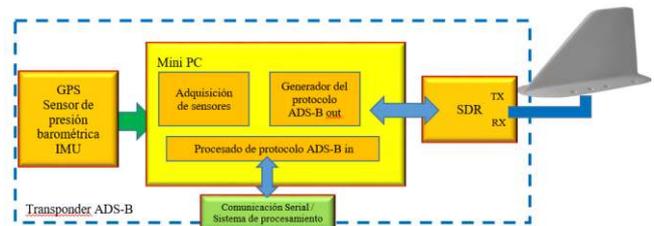


Fig. 1: Esquema del transponder ADS-B embebido

B. Descripción del Hardware

Para la adquisición de los datos de posición (Latitud, Longitud, Altitud), velocidad (Ground speed), tiempo (UTC), altitud (Barométrica); se utilizó una placa desarrollada en el Centro I+D de Tecnologías Aeronáuticas de Fuerza Aérea Argentina (CITEA); la cual tiene una interfase serial para comunicar sus datos a la mini PC.

Dicha placa consta de sensores como GNSS (Modelo: Quectel L80)[3], IMU (BNO055) [4] de 9 grados de libertad y sensor de presión barométrica (BMP280)[5]; todo controlado por un ESP8266[6]. En la Fig. 2, se muestran la interconexión de todo el hardware involucrado en este desarrollo.

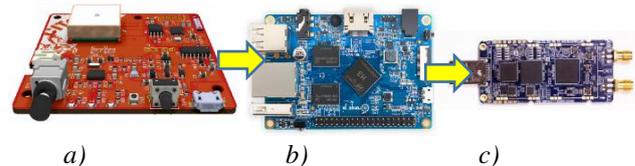


Fig. 2: a) Registrador de parámetros de vuelo, b) mini PC, c) LimeSDR

Como componente principal de procesamiento, se utilizó una Mini-PC [11]; la cual ejecuta el programa principal que recibe los datos de la placa de sensores, procesa la información

y arma la trama según protocolo ADS-B. En dicha placa principal, se ejecutan los scripts desarrollados en Python generados desde el software GNURadio [7][8].

Por último para la modulación y transmisión de la señal de RF en 1090MHz se utilizó un Hardware SDR[9] [12].

III. DESCRIPCIÓN DEL PROTOCOLO ADS-B

Un mensaje del protocolo ADS-B[1] tiene una longitud de 112 bits, y está compuesto por cinco partes, las cuales varían en el contenido, según la información que se quiera transmitir.

El mensaje ADS-B típico tiene una duración de 120 μ s y utiliza una Modulación por Posición de Pulso (PPM) para transmitir los bits. Teniendo en cuenta que, para ADS-B, el intervalo de tiempo es igual a un microsegundo, un "1" o "0" se indica con un pulso en la primera o segunda mitad (0,5 μ s) del intervalo de tiempo respectivamente, para ello se utiliza una Codificación Manchester[13].

Los primeros cuatro pulsos se consideran el preámbulo. El preámbulo sirve como indicador y sincronización de tiempo, notificando al receptor que un mensaje comenzará a 8 μ s del primer pulso de preámbulo.

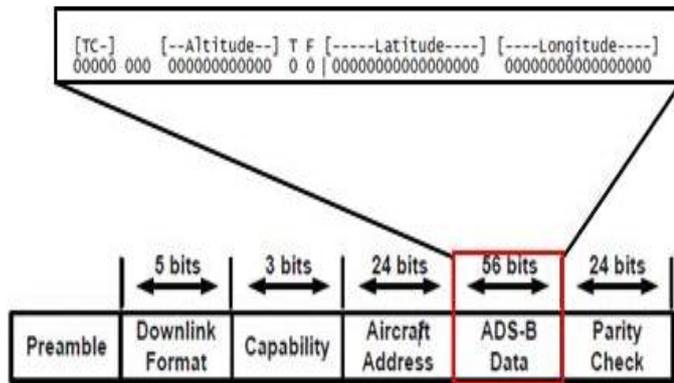


Fig. 3: Estructura completa de la trama ADS-B

TABLA I- BLOQUES DEL PROTOCOLO ADS-B

Nº Bits	Bits	Abreviatura	Nombre
5	1-5	DF	Formato de Bajada
3	6-8	CA	Capacidad (identificación adicional)
24	9-32	ICAO	Dirección de la Aeronave
56	33-88	DATA	Bloque de Datos
	[33-37]	[TC]	Tipo del Código del Mensaje
24	89-112	PI	Código de Paridad del Mensaje

En la Fig. 3 se muestra la estructura del protocolo ADS-B, en donde primero se envía el preámbulo para el receptor; luego del preámbulo se envían los bloques de la Tabla I, en el que se realizará una descripción de los mismos a continuación.

A. Bloques del protocolo ADS-B

- DF (Formato de Bajada): un mensaje ADS-B debe comenzar con el formato descendente DF=11 (Respuesta Modo S) o DF=17 (Modo S Formato Extendido)

- CA (Capacidad): Es un código de 3 bit que sigue a continuación de los 5 bits de DF, en este bloque se indica la capacidad de la estación de transmisión.

- ICAO (Identificación de la Aeronave): en cada mensaje ADS-B la aeronave que transmite el mensaje se identifica utilizando el ICAO (por sus siglas del inglés, International Civil Aviation Organization), este bloque tiene una longitud de 24 bits. Se asigna una dirección de ICAO única a cada transpondedor en Modo S de una aeronave

- Data (Bloque de datos): si el DF es igual a 17, el mensaje contendrá un bloque de información de 56 bits, estos

bits ocupan la posición 33 al 88 del mensaje ADS-B y dependiendo de los primeros 5 valores de este bloque o del valor del TC (tipo de código).

- PI (Código de Paridad): ADS-B utiliza una verificación de redundancia cíclica para validar la exactitud del mensaje recibido, son los últimos 24 bits del protocolo

B. Bloques Datos

El cuarto bloque de la trama ADS-B es utilizado para transmitir la información, esta información es codificada en 56 bits que ocupan la posición 33 al 88 de la trama ADS-B.

1) Posición en Aire

Este mensaje se utiliza para transmitir la posición y la altitud de la aeronave Hay 8 campos dentro de un mensaje que transmite posición en aire, el mensaje que transmite cada bloque, así como también la posición dentro del bloque

2) Codificación CPR de Longitud y Latitud

La información de posicionamiento en ADS-B es codificada utilizando un formato de codificación llamado Reporte de Posición Compacta o conocido como CPR (por sus siglas en inglés, Compact Position Reporting). El principio de funcionamiento de este formato de codificación es codificar más decimales, utilizando menos bits. Para informar una posición, las fracciones de latitud y longitud se codifican utilizando 17 bits y estos bits son transmitidos en los mensajes de posición; según se describe en el estándar DO260B.

3) Codificación de la Altitud

la altitud se codifica en un total de 12 bits. El proceso que se utiliza para la codificación es el siguiente:

El octavo bit de la altitud o lo que es lo mismo el número 48 de la trama indica si la altitud está codificada con resolución de 25 pies o 100 pies, este bit suele ser llamado Q-bit. Para resolución de 100 pies Q-bit es igual a '0' y para resoluciones de 25 pies, Q-bit es igual a '1'. Por defecto se utiliza la resolución de 25 pies para altitudes menores a 50175 pies

4) Codificación de la Velocidad

En ADS-B existen diferentes tipos de mensajes para transmitir información de velocidad en el aire, pero todos ellos son transmitidos con el mismo tipo de código (TC=19),

5) Codificación Rumbo Magnético o Heading

La representación binaria del bloque heading está compuesto por 10 bits, para la codificación se utiliza el siguiente proceso. Se realiza una parametrización, para aprovechar de mejor manera los 10 bits, en vez de representar el heading en valores que van de 0° a 360°, se utiliza una conversión de 0 a 1024

IV. DESARROLLO

Todo el desarrollo de software se realizó en GNURadio, utilizando los bloques nativos del programa y para esta aplicación se generaron bloques propios que procesan la información adquirida y generan la trama ADS-B. [1][2][10], para luego generar la modulación PPM y posteriormente utilizar el Hardware SDR para su transmisión.

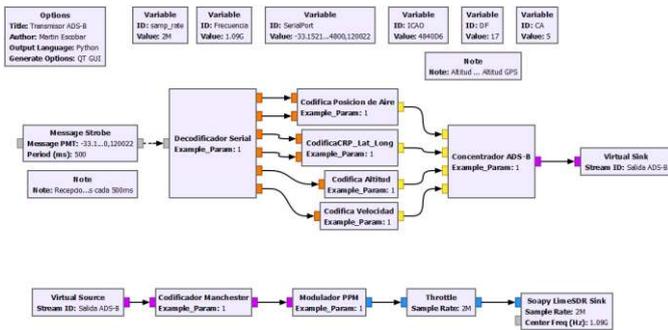


Fig. 4: Diagrama en bloques del transmisor ADS-B

En la Fig. 4, se muestra el diagrama en bloques completo del transmisor ADS-B o transponder ADS-B Out

A. Descripción funcional de los bloques

Para la prueba funcional del sistema se necesita adquirir los datos de la placa de sensores, la cual, se recibe a través de una cadena de caracteres ordenados de la siguiente manera; Latitud, Longitud, Altitud GPS, Altitud Barométrica, Velocidades, Hora UTC. Los datos van separados con una coma, y se envían a una tasa de 115200bps dos veces por segundos; estos son ingresados al bloque “Decodificador Serial” que se encarga de separar los datos en canales diferentes para ingresar a cada uno de los codificadores.

Los bloques que le siguen al “Decodificador Serial”, serían los distintos codificadores (“Codifica Posición de Aire”, “CodificaCRP_Lat_Long”, “Codifica Altitud”, “Codifica Velocidad”). Cada uno de estos bloques implementa los algoritmos necesarios para cumplir con el estándar D0260B, para compactar la información y generar el protocolo ADS-B.

Una vez que la información fue codificada se envía al bloque “Concentrador ADS-B”, este mismo es el que genera todo el protocolo; comenzando por el preámbulo luego el código DF, el CA, el ICAO, el bloque de datos de 56 bits y en función de eso genera el CRC en los últimos 24 bits.

Antes de realizar la transmisión inalámbrica se debe realizar la codificación Manchester para luego ingresar al último bloque que es “Modulador PPM”, este mismo toma el protocolo que es una secuencia de 1 y 0; para convertirlos a una señal del tipo “analogica” compleja que será la utilizada para modular y transmitir en 1090MHz por la LimeSDR a una potencia de 1mW.

B. Resultados

Las pruebas se realizaron con los siguientes datos: Latitud = -31.418059 S, Longitud = -64.292926 O, Altitud = 5750 pies, Velocidad = 159 nudos, ICAO = 4840D6, DF = 17, CA = 4, Rumbo = 23°, Velocidad vertical = 2 nudos; para corroborar la trama ASD-B en la estación receptora

Estos datos pasan por los bloques descriptos anteriormente y queda como trama resultante, que se envía al modulador PPM, “8D4840D6202CC371C32CE0576098”.

Esta trama que se envía al SDR, como se muestra en la Fig. 5; será validada, utilizando una estación receptora ADS-B[13] [1] activa y con cobertura local. Esta estación tiene todo el protocolo ADS-B implementado y tiene la capacidad de representar en un mapa georeferenciado la aeronave que está transmitiendo en tiempo real

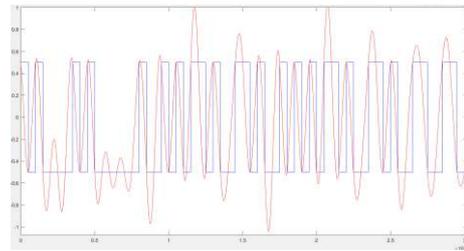


Fig. 5: Datos generados y acondicionados para el transmisor

Una vez verificada la integridad de la trama ADS-B con datos conocidos, se procedió a realizar ensayos con los datos adquiridos del registrador de parámetros en vuelo en un recorrido en proximidades de la estación receptora

V. CONCLUSIONES

Se logró implementar y desarrollar un prototipo funcional de un transponder ADS-B out, con las características mínimas y funcionales como requiere en el estándar DO260B; aunque todavía se requiere realizar ciertos ajustes desde el punto de vista de la modulación ya que se generan muchas señales espurias que generan señales mayores a 5MHz de ancho de banda y pueden producir interferencias en otros servicios cercanos a la frecuencia de 1090MHz. Además, como trabajos a posterior se podría instalar en algún Vehículo Aéreo no tripulado (UAV) y realizar una transmisión con una potencia por debajo de 1W, para evitar realizar interferencias en el tráfico aéreo, y así probar el sistema en vuelo para mejorar el desarrollo

REFERENCIAS

- [1] RTCA DO-260 “Minimum Operational Performance Standards for 1090 MHz Extended Squitter Automatic Dependent Surveillance – Broadcast (ADS-B) and Traffic Information Services – Broadcast (TIS-B)”, 17 December 2020
- [2] Federal Aviation Administration, 14 CFR 91.227 Automatic Dependent Surveillance-Broadcast (ADS-B) Out equipment performance requirements, 2016
- [3] Quectel , “Quectel L80, Compact GPS Module Integrated with Patch Antenna”, <https://www.quectel.com/product/gps-only-l80>, 2017
- [4] Bosch, “BNO055 Intelligent 9-axis absolute orientation sensor”, <https://www.bosch-sensortec.com>, Oct. 2021
- [5] Bosch, “BMP280 Digital Pressure Sensor”, <https://www.bosch-sensortec.com>, 2021
- [6] Espressif Inc, “ESP8266EX”. www.espressif.com , 2020
- [7] Eduardo Gonzalez, Carlos Zerbini, Guillermo Riva, Daniel Rosso, Fernanda Suarez, Luis Guanuco; “SDR con GNU Radio: de la teoría a la aplicación” , Congreso de Microelectronica Aplicada uEA 2016
- [8] GNURadio, https://wiki.gnuradio.org/index.php/Main_Page , 2022
- [9] G. Goavec-Merou, P.-Y. Bourgeois, J.-M Friedt “Embedded GNU Radio running on Zynq/PlutoSDR”; FEMTO-ST Time & Frequency, Besancon, France 2021
- [10] R. Michael Guterres, Stanley R. Jones, Gregory L. Orrell, Robert C. Strain; “ADS-B Surveillance System Performance with Small UAS at Low Altitudes” The MITRE Corporation, 7525 Colshire Drive, McLean, VA 22102, January 2017
- [11] Orange Pi Plus; https://geekmatic.in.ua/pdf/OrangePi_PC_user_manual_v0.9.1.pdf, June 2020
- [12] LimeSDR mini V1.0; https://wiki.myriadrf.org/LimeSDR-Mini_v1.1_hardware_description, March 2022
- [13] Abdulrazaq Abdulaziz, Abdulmalik Shehu Yaro, Ashraf Adam Ahmad, Habeeb Bello, SalauHabeeb Bello Salau; “Optimum Receiver for Decoding Automatic Dependent Surveillance Broadcast (ADS-B) Signals”, January 2015

Sistema embebido escalable para el monitoreo de parámetros en invernaderos mediante IoT

Mardegan Matías N.
Línea CeTaD - GCA
Instituto LEICI
(UNLP-CONICET)
La Plata, Argentina
matiasmardegan@gmail.com

Salina Mauro
Programa TICAPPS, UNAJ
msalina@unaj.edu.ar

Salvatore Juan E.
Programa TICAPPS, IlyA, UNAJ
jsalvatore@unaj.edu.ar

Osio Jorge R.
Línea CeTAD, GCA, Instituto LEICI
(UNLP- CONICET)
Programa TICAPPS, IlyA, UNAJ
Jorge.osio@ing.unlp.edu.ar

Cappelletti Marcelo
Línea CeTAD, GCA, Instituto
LEICI (UNLP- CONICET)
Programa TICAPPS, IlyA, UNAJ
mcappelletti@unaj.edu.ar

Resumen— El trabajo se centra en el sensado y control de un conjunto de magnitudes físicas con el objeto de mantenerlas en un rango de valores adecuado para el correcto desarrollo y crecimiento de cultivos. Mediante la comunicación inalámbrica sobre el protocolo LoRa se comunican los nodos y un gateway conformando una red local escalable. El gateway, que funciona como punto de conexión entre la nube y el resto del sistema, recibe la información de cada nodo y la redirecciona a una base de datos para su almacenamiento y posterior procesamiento. Los datos provenientes del nodo, así como también los recolectados por los sensores del gateway pueden visualizarse en una dashboard de forma amigable para el usuario final.

Entre los parámetros de interés a monitorear se tiene humedad del suelo, humedad ambiente, temperatura, iluminación y presión atmosférica. Con esta información se pueden monitorear los parámetros principales del cultivo y realizar las acciones de control necesarias (iluminación y riego) para obtener un microclima controlado dentro del invernadero. Además, el conocimiento de estos parámetros podrá ser utilizado para predecir la presencia de diversas plagas que pueden afectar la producción.

Palabras clave—Sistema embebido, red de sensores, sistema IoT, monitoreo de cultivos

I. INTRODUCCIÓN

En Argentina, la agricultura es una de las actividades más importantes de la economía, en este sentido en un alto porcentaje de la producción no cuenta con las condiciones necesarias para optimizar el uso de los recursos [1].

Es por eso que se plantea como objetivo principal proporcionar un sistema de bajo consumo y con un impacto ambiental reducido que asegure las condiciones necesarias en diferentes invernaderos, mediante una red de sensores que obtengan un conjunto de magnitudes físicas de interés y emitan las acciones necesarias de control sin la intervención humana.

Entre los objetivos específicos se propone que el sistema sea capaz de mostrar la información recopilada de los sensores y almacenarla de forma local (para garantizar la resiliencia de los datos). Otro de los objetivos particulares consiste en implementar una aplicación IoT que permita subir la información a un servidor para su almacenamiento y posterior tratamiento, posibilitando el acceso de forma remota.

El artículo está organizado en una introducción, en donde se describe la problemática y los objetivos del

trabajo. Luego la implementación, que contiene detalles del sistema tanto de hardware (microcontroladores, sensores y actuadores) como de software (software embebido para lectura de sensores y el accionamiento de actuadores, software para la comunicación RF en la red local y software para mostrar y almacenar los datos de forma local y en un servidor remoto). Por último, los resultados y conclusiones.

II. IMPLEMENTACIÓN

En la Fig. 1 se representan las distintas etapas del sistema completo. Por un lado, se tienen los nodos del sistema, los cuales permiten la conexión con el mundo físico y están constituidos por sensores de temperatura y humedad (DHT22), presión (BMP280), humedad del suelo (sensor capacitivo V2.0), iluminación (TEMT6000) y concentración de dióxido de carbono (MQ-135). Dichos nodos también incluyen diversos actuadores para controlar la iluminación, ventilación y humedad del suelo mediante la activación/desactivación de relés. Además, cuentan con módulos de comunicación RF LoRa [2]. El tratamiento de las medidas de los sensores, el control de los actuadores, el seguimiento de la fecha y hora, y el envío de los datos se lleva a cabo mediante el microcontrolador ATmega 328p que se encuentra en el kit Arduino uno.

Por su parte, el gateway realiza una serie de tareas por medio del ESP32 incluido en la placa “HELTEC LoRa 32 V2” [3], que proporciona una interfaz para la comunicación vía LoRa con los nodos, la conectividad WiFi para la comunicación con el servidor y una pantalla OLED. Al recibir los datos provenientes de los nodos, el gateway los muestra en la pantalla, los almacena en una memoria microSD y los envía a un servidor utilizando la conectividad WiFi para su almacenamiento en la base de datos. Lo mencionado también se lleva a cabo con los datos recopilados por los sensores del propio gateway. Se realiza también un seguimiento de la fecha y la hora por medio de un servidor NTP.

El diseño del sistema fue pensado para un invernadero de grandes dimensiones por lo que se requiere un método adecuado para el intercambio de información entre los nodos y el gateway. Para ello se utilizaron los módulos LoRa, que permiten implementar una red local de largo alcance empleando modulación en radiofrecuencia, a la frecuencia de 433 MHz. Además, se implementó un protocolo tipo daisy chain de forma que, si un dispositivo recibe un paquete que tiene la dirección de otro nodo como destinatario, en lugar de almacenarlo lo retransmite, para

conseguir abarcar grandes extensiones durante la implementación de la red local.

Por su parte, El gateway se alimenta de la red eléctrica y posee conectividad Wifi, pero los nodos deben ser alimentados a batería debido a su ubicación.

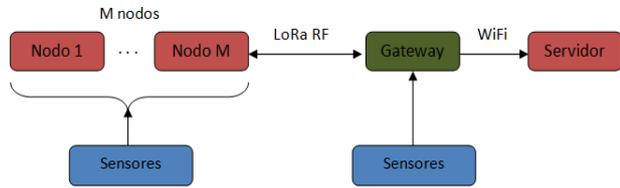


Fig. 1 Diagrama del sistema completo

Para el diseño del HW se utilizó el software de desarrollo KiCAD. De esta manera, se diseñaron los esquemáticos para ambos casos; luego los PCB's y por último se utilizó el software FreeCAD en conjunto con KiCAD para el armado de los modelos 3D de las placas del nodo y el gateway.

El esquemático diseñado para el hardware de los nodos y el gateway fue realizado en kicad, al igual que los circuitos y su modelo 3D.

La Fig. 2 muestra la placa de los nodos, en donde se muestra el RTC que es necesario para la temporización y la fecha de las muestras. En este sentido se debe aclarar que el RTC se sincroniza con la hora que obtiene el compilador del sistema operativo, que a su vez se encuentra sincronizada con los servidores de Microsoft y es la misma que la de la base de datos. Adicionalmente, el nodo cuenta con el core formado por el microcontrolador ATmega328 (incluido en el kit de Arduino uno), el módulo LoRa SX1278 para la comunicación inalámbrica [5], los conectores de los sensores y su alimentación.

En la Fig. 3 se muestra el modelo 3D para el caso del gateway, en donde se observa la microSD para el almacenamiento local de los datos, el core con el microcontrolador ESP32, la pantalla OLED y el módulo LoRa integrado. Adicionalmente, cuenta con los conectores para la conexión de sensores y actuadores para el caso en que el sistema se implemente en un único invernadero.

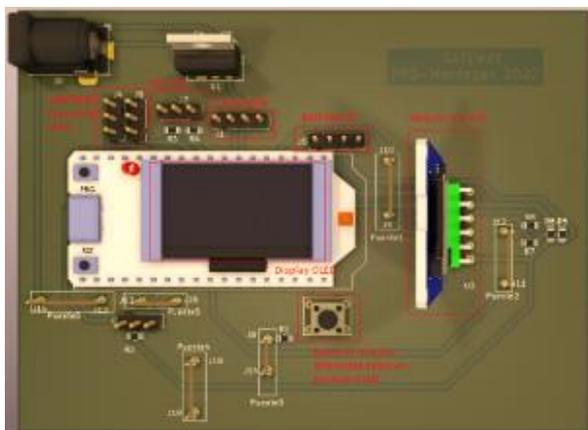


Fig. 2 Modelo 3D de los nodos

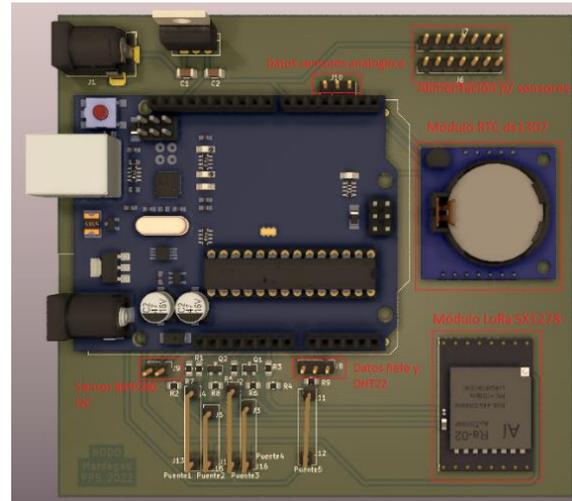


Fig. 3 Modelo 3D del Gateway

Para la programación del software embebido se utilizó el IDE de Arduino que brinda la posibilidad de cargar código tanto para los microcontroladores de Atmel (nodos) como para los de Espressif (gateway).

En la Fig. 4 se muestra el diagrama de flujo del soft embebido del nodo. Respecto a la funcionalidad de los nodos, simplemente responden a las peticiones del gateway leyendo los parámetros para cierto tiempo y enviándolos conjuntamente con la marca de tiempo. A su vez analizan si los parámetros superan los umbrales definidos para determinar si es necesario realizar alguna acción local. Respecto al gateway puede o no tener conectados los sensores y actuadores, por lo que en caso afirmativo realiza la lectura correspondiente, almacena los datos en la memoria SD y procede a encuestar todos los nodos de la red local. A medida que va recibiendo los paquetes de cada nodo, almacena dicha información en la memoria SD y la envía mediante Wifi hacia la base de datos del servidor remoto [4].

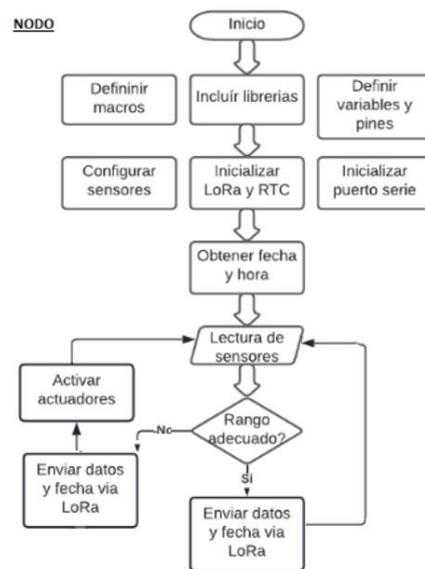


Fig. 4 Diagramas de flujo del nodo

En relación con la base de datos, esta fue

implementada en firebase por las facilidades que provee para vincularla a una aplicación web en donde los datos cambian constantemente (debido a que todos acceden a la misma base de datos almacenada en la nube). Además, posibilita escalamiento automático.

En referencia a la seguridad, Firebase Authentication facilitó la creación de un sistema de autenticación seguro, a la vez que mejoró la experiencia de integración y acceso para los usuarios finales. Esto proporcionó una solución de identidad de extremo a extremo, compatible con cuentas de correo electrónico y contraseñas.

III. RESULTADOS

En relación con los nodos, se realizó un análisis del consumo debido a que funcionan a batería y se determinó que en modo sleep consumen 11 mA. Durante la lectura y transmisión de datos consumen entre 70 y 100 mA (esto es por un muy corto lapso). El sensor Mq135 consume alrededor de 150mA, es por eso que solo se instaló en el gateway.

Respecto a la comunicación entre el gateway y los nodos se realizó mediante el protocolo LoRa en el modo de comunicación nodo/gateway, en donde se utilizó el modo invert-IQ para transmitir desde el gateway y disable-invert-IQ para recibir [5]. En el caso de los nodos se debe usar la configuración opuesta. Respecto a las pruebas de alcance realizadas, se realizó una prueba entre gateway y nodos con obstáculos y se logró un alcance inicial de unos 200 m, luego de ajustar algunos parámetros del protocolo se logró un alcance de 250 m a cambio de un incremento notable en el consumo y el tiempo de comunicación. Afortunadamente, en el lugar de aplicación del sistema no hay obstáculos, por lo que las pruebas realizadas permitieron lograr alcances cercanos a los 1000 m en un ambiente con pocos obstáculos [6].

Respecto al envío de datos a la nube, se realizaron dos implementaciones, una mediante la plataforma comercial Thingspeak y la otra personalizada mediante una base de datos NoSQL en Firebase integrada con una interfaz web (ver Fig. 5). En ambos casos se configuraron dashboards para la visualización de la información a la que el usuario accede de forma remota. En el caso de la plataforma comercial de Thingspeak se observó una latencia de entre 15 y 20 segundos durante el envío de los datos al servidor y en algunos casos hasta pérdida de paquetes. Respecto a la implementación en Firebase, las pruebas arrojaron latencias de entre 1 y 2 segundos y no se registró pérdida de información. Teniendo en cuenta que la latencia es aceptable debajo de los 5 s para accionamiento remoto, los resultados con Firebase son más que satisfactorios.

Se debe destacar que las prestaciones logradas con Firebase son muy buenas teniendo en cuenta que por lo general se trabaja con fenómenos físicos cuya variación es lenta. En ese sentido se realizaron pruebas de transmisión a intervalos de tiempo de entre 1 s y 10 s. obteniendo muy buenos resultados sin pérdida de datos.

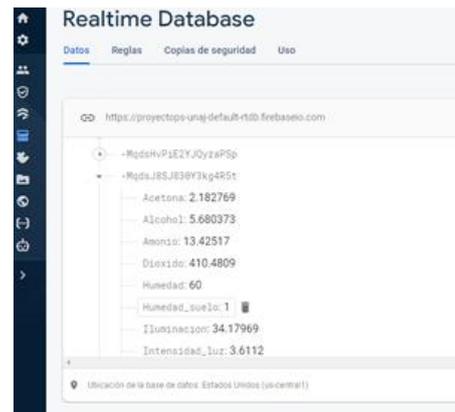


Fig. 5 Parámetros de un nodo en Firebase

En comparación con algunas plataformas comerciales, se pudo constatar que, para similares exigencias, se produjeron pérdidas de datos, o recepción fuera de orden. Por lo que la aplicación definitiva se realizó con Firebase.

IV. CONCLUSIONES

En primer lugar, se cumplió el objetivo principal de implementar una red de sensores escalable con interfaz IoT a una base de datos con acceso remoto. Además, se logró un funcionamiento eficiente y confiable en la red local y la interfaz de comunicación con Firebase.

Las plataformas de IoT resultan muy útiles cuando se cuenta con una base de datos que permita el almacenamiento de gran cantidad de información. Dado que esta información servirá al usuario final para hacer un análisis de los parámetros en diferentes periodos y realizar las acciones correspondientes.

Respecto a los beneficios conseguidos, se logró automatizar el riego, logrando minimizar el uso de ese recurso. En definitiva, se posibilitó el acceso a la información de forma remota y la automatización del sistema incorporando las tecnologías de IoT.

REFERENCIAS

- [1] Diego Coatz, "ANÁLISIS DE DIAGNÓSTICO TECNOLÓGICO SECTORIAL". Editor Lic. Sebastián Spector, Año 2013. https://www.argentina.gob.ar/sites/default/files/ats_-_fruticola.pdf
- [2] Raghavendra, C. S., Sivalingam, K. M., Znati, T.: Wireless sensor networks. Springer (2006)
- [3] ESP32 Technical Reference Manual Version 3.4, Espressif Systems. Año 2018. <https://cutt.ly/4KWuniS>
- [4] Acquarone Cesar, Osio Jorge R, Hromek Erik, Salvatore Juan, Montezanti Diego, Morales D. Martín, "Plataforma de desarrollo para IoT", IV conaiisi, 2017. <https://acortar.link/piQ7II>
- [5] Datasheet del módulo LoRa SX1272 Rev 3.1, Semtech. Año 2017. <https://acortar.link/gKvKTy>
- [6] J. Rios Urquidí, J osio, W Aróztegui, "Diseño de un sistema de monitoreo y prevención basado en IoT mediante el protocolo LoRa", Congreso uEA, UNSAM, 2019.

Computación paralela en sistemas de radar mediante Computadoras de Placa Reducida

Alian Ernesto Matos Rodríguez, Ing.
Universidad Tecnológica de la Habana
La Habana, Cuba
aematos@nauta.cu

Leandro Zambrano Méndez, Dr. C.
Universidad Tecnológica de la Habana
La Habana, Cuba
lzambrano@ceis.edu.cujae.cu

Resumen—*Los radares son dispositivos dedicados comúnmente a la detección y rastreo de objetivos próximos. El procesamiento y la visualización de sus datos son operaciones computacionalmente muy exigentes. En el pasado, se requería hardware altamente especializado para realizar las operaciones, las cuales en la actualidad se realizan mediante computadoras personales. Las soluciones mediante computadoras disminuyen los costos en comparación con los equipos antiguos, pero la rentabilidad económica es menor teniendo en consideración otras tecnologías como las Computadoras de Placa Reducida (SBC). La principal contribución de este trabajo es el empleo de las SBC y la computación paralela para el procesamiento y visualización de los datos en los sistemas de radar, lo que permitió hacer un uso eficiente de los recursos de cómputo y aumento de la rentabilidad económica en comparación con las computadoras personales.*

Palabras claves—Radar, SBC, Computación Paralela, OpenMP.

I. INTRODUCCIÓN

El Radar es un acrónimo derivado de las palabras *Radio Detection and Ranging*, son dispositivos dedicados comúnmente a la detección y rastreo de objetivos próximos en ambientes terrestres, marítimos y aéreos [1]. Estos irradian energía electromagnética y detectan el eco devuelto por los objetos reflectantes. Desde su surgimiento hasta la actualidad, los sistemas de radar se han expandido tanto en capacidad como en aplicabilidad, y dentro de sus aplicaciones se incluyen la observación de la situación meteorológica, la seguridad de los barcos, la exploración espacial, control de tráfico aéreo y con fines de defensa, tanto naval, aéreo como marítimo en el área militar [2].

El procesamiento y la visualización de datos de radar son operaciones computacionalmente muy exigentes. En el pasado, se necesitaba de hardware altamente especializado para realizar estas operaciones. El resultado fue que los equipos de procesamiento y visualización de datos de radar tenían altos costos de desarrollo y mantenimiento, eran grande en tamaño y difícil de modificar [3].

Los últimos avances en los sistemas de radar, son consecuencia de los adelantos en el mundo de las computadoras y la electrónica de alta velocidad. En la actualidad la capacidad de procesamiento de las computadoras personales (PC, siglas en inglés) ha ido significativamente en aumento [3]. Disímiles autores en el transcurso de los años han abordado el procesamiento y visualización de datos de radar mediante computadoras personales [3-9], a pesar de que en sus propuestas disminuyeron considerablemente los costos en comparación a los equipos antiguos, la rentabilidad económica es menor

en comparación a tecnologías de hardware modernas como las Computadoras de Placas Reducidas (Single Board Computer, SBC).

La principal contribución de este trabajo es el empleo de las SBC y la computación paralela para el procesamiento y visualización de los datos en los sistemas de radar, lo que permitió hacer un uso eficiente de los recursos de cómputo y un aumento de la rentabilidad económica en comparación con las computadoras personales.

II. COMPUTADORA DE PLACA ÚNICA

Las SBC son computadoras construidas en una sola placa de circuito, con uno o más microprocesadores, memoria, periféricos de entrada / salida (E / S) y otras características requeridas de una computadora funcional. Su uso ha aumentado de forma vertiginosa en los últimos años debido principalmente a su bajo costo, alto rendimiento, bajo consumo de potencia y fácil implementación [10]. Los bajos precios de las SBCs es una de las características más importantes de estos dispositivos, están relacionados a los avances en la electrónica y tecnologías de los semiconductores [11].

Debido a los grandes niveles de integración y reducción de componentes y conectores, suelen ser más pequeñas, livianas y con un mejor manejo de la potencia eléctrica que las computadoras convencionales. A pesar de que las SBC por lo general tienen menor potencia computacional que las computadoras de escritorio estándar, son adecuadas para determinadas aplicaciones en específico [11].

A. Odroid XU4

El Odroid-XU4 es un potente, versátil y económico ordenador de placa reducida [12]. Posee una arquitectura ARM provista de un SoC Samsung Exynos 5422 big.LITTLE de ocho núcleos, formado por 4 núcleos Cortex-A15 a 2GHz y por 4 núcleos Cortex-A7 a 1.4 GHz [13, 14]. La Fig. 1 muestra la apariencia de esta computadora de placa única.



Fig. 1. SBC Odroid XU4.

B. Odroid XU4Q

El Odroid-XU4Q es una versión del Odroid-XU4 donde se reemplazó el ventilador de enfriamiento original por un disipador de calor pasivo, como se muestra en la Fig. 2. Esta computadora de placa única mantiene las mismas especificaciones que el Odroid-XU4.



Fig. 2. SBC Odroid XU4Q.

C. Odroid N2

El Odroid-N2 es un ordenador de placa reducida de nueva generación. Cuenta con una arquitectura multi-núcleo asimétrica basada en ARM, donde integra núcleos con distintas características en un mismo chip. Posee un GPUMali-G52 y la CPU principal se basa en una arquitectura big.LITTLE, que integra un clúster de cuatro núcleos Cortex-A73 y otro clúster de dos núcleos Cortex-A53 [15]. Los núcleos A73 funcionan a 1.8 GHz, mientras que los núcleos A53 a 1.7 GHz. La Fig. 3. muestra la apariencia de este dispositivo.



Fig. 3. SBC Odroid N2.

D. Raspberry Pi4 Modelo B

La Raspberry Pi 4 Modelo B es un producto de la popular gama de ordenadores Raspberry Pi. Posee un procesador quad-core de 64 bit de alto rendimiento y hasta 8GB de memoria RAM [16]. La Fig. 4 muestra el aspecto de esta computadora de placa única.



Fig. 4. SBC Raspberry Pi4 Modelo B.

III. COMPUTACIÓN PARALELA

La computación paralela es el uso simultáneo de múltiples recursos de cómputo para resolver un problema computacional [17], permitiendo que se ejecuten varios procesos concurrentes. Todas las computadoras modernas admiten el paralelismo en el hardware a través de al menos una función paralela, incluidas instrucciones vectoriales, núcleos multiproceso, núcleos de múltiples procesadores, procesadores múltiples, motores gráficos y coprocesadores paralelos [18]. Una de las tecnologías de software que permiten desarrollar aplicaciones paralelas es OpenMP, esta API ha sido ampliamente adoptada por la comunidad científica informática. Una manera de diseñar e implementar de manera eficiente los algoritmos paralelos es a través del uso de patrones de computación paralela y distribuida [18].

A. API OPenMP

Es una API multiplataforma para la programación paralela de memoria compartida. Está diseñado para los lenguajes Fortran, C y C++. Está formado por un conjunto de directivas y funcionalidades que permiten el desarrollo de aplicaciones paralelas [17-19]. OpenMP está basado en el modelo Fork-Join [18, 19]. Este modelo permite que el flujo de control de bifurque en múltiples flujos paralelos en puntos designados del programa, para fusionarse en un punto posterior y reanudar la ejecución secuencial [18].

B. Segmentación de cauce

El patrón de programación paralela conocido como *segmentación de cauce* consiste en una secuencia lineal de etapas (segmentos de código de un algoritmo mayor). Cada etapa realiza una transformación en los datos y la pasa a la siguiente para continuar en el procesamiento de los mismos [18].

IV. PRUEBAS Y ANÁLISIS DE RESULTADOS

Se realizó un experimento bajo el principio de repetición para minimizar el error experimental, donde se hicieron 10 ejecuciones del diseño del algoritmo secuencial y paralelo propuesto en [20], con el objetivo de obtener los tiempos de respuesta de los SBC con ambas versiones del algoritmo. Los tiempos resultantes en milisegundos de la versión del algoritmo secuencial y paralelo se muestran en la Fig. 5 y en la Fig. 6 respectivamente.

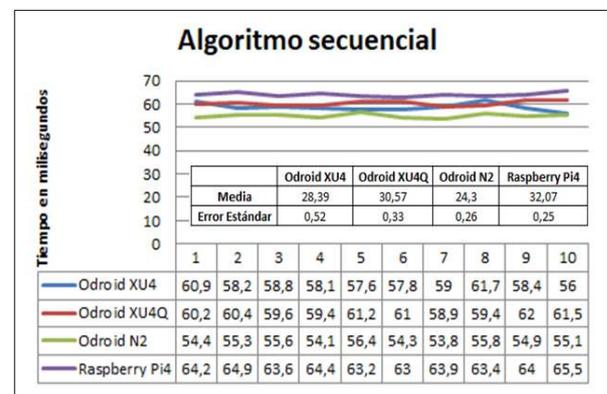


Fig. 5. Tiempos de respuesta del algoritmo secuencial.

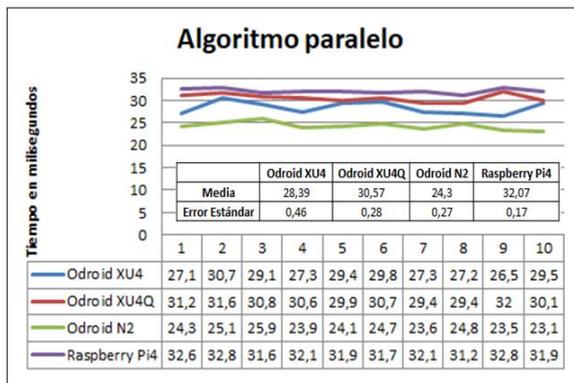


Fig. 6. Tiempos de respuesta del algoritmo paralelo.

En las gráficas mostradas en las Fig. 5 y Fig. 6 se aprecia como el Odroid N2 tiene los mejores tiempos de respuestas en ambas variantes del algoritmo, mientras que la Raspberry Pi 4 fue la que arrojó los peores tiempos de ejecución. Con el algoritmo paralelo todos los SBC logran una disminución evidente de los tiempos de respuesta.

Durante la realización del análisis de los resultados se procedió a generar el diagrama de Pareto mediante la herramienta Minitab 19 [21], el cual se muestra en la Fig. 7, con el objetivo es investigar los efectos de las variables de entrada, conocido como factores, sobre la variable de respuesta.

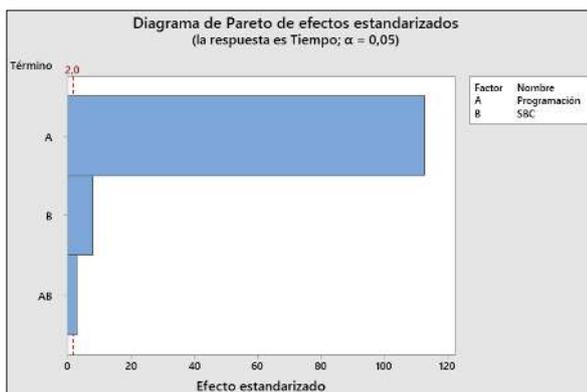


Fig. 7. Diagrama de Pareto.

Este diagrama permite detectar los efectos determinantes de los factores en el rendimiento del sistema, así como las variables que tienen un mayor impacto sobre la variable de respuesta. Como se aprecia en la Fig. 7 todos los factores controlables influyen significativamente en los tiempos de respuesta, siendo la variante de programación el factor más significativo.

V. CONCLUSIONES

Las pruebas experimentales y el análisis de los resultados permitieron determinar la influencia de los factores tipo de programación y tipo de dispositivo en los tiempos de respuesta de la aplicación, todos los factores tienen influencia en el rendimiento del sistema, pero quedó la evidencia de que el factor más significativo es la variante de programación. Las computadoras de placa única en conjunto con un diseño paralelo del software permiten obtener una solución para el procesamiento y visualización de datos de

radar, de bajo costo, alto rendimiento, bajo consumo de potencia y fácil implementación.

VI. TRABAJOS FUTUROS

Modernización del sistema de procesamiento y representación de datos de un radar de seguimiento analógico con el empleo de computadoras de placa única.

REFERENCIAS

- [1] M. L. William and S. A. James, "Principles of Modern Radar, vol. III: Radar Applications," ed: SciTech Publishing, New Jersey, 2014.
- [2] N. Bhatta and M. Priya, "RADAR and its applications," *International Journal of Circuit Theory and Applications (IJCTA)*, vol. 10, pp. 1-9, 01/01 2017.
- [3] M. Jevtić and M. Stamatović, "Radar data processing and visualization on desktop platforms," in *Proc. 17th Telecommunications forum TELFOR*, 2009, pp. 1315-1318.
- [4] M. Manasa and M. Hemalatha, "Desing of generic radar data visualizer using model view controller pattern," *International Journal For Technological Research In Engineering*, vol. 2, August-2015 2015.
- [5] C. Ravindra, S. Rajkumar, and M. S. Babu, "Design and Implementation of Radar Console Displays for Multi Object Tracking Radar using Qt-IDE," in *11th International Radar Symposium India*, 2017.
- [6] V. Pezhgorski and M. Lazarova, "Real Time GPU Accelerated Radar Scan Conversion and Visualization," in *Proceedings of the 18th International Conference on Computer Systems and Technologies*, 2017, pp. 249-256.
- [7] Z. Miao, W. Xu, and H. Chen, "Technology of Radar Terminal on Researching Software Display," in *Proceedings of the 2nd International Conference on Vision, Image and Signal Processing*, 2018, pp. 1-5.
- [8] M. Syeda, N. Arjuman, T. Faiza, and E. Mohammed, "Enhancement of Flycatchers Using Radar for Tracking of Aircrafts Which Aviates at Lower Levels," *International Journal of Latest Engineering Research and Applications (IJLERA)*, vol. 02, pp. 37-42, April – 2017 2017.
- [9] O. Heriana, D. Kurniawan, A. N. Rahman, S. Hardiati, and E. J. Pristianto, "Implementation of Plan Position Indicator Display for Low Probability of Intercept Radar," in *2018 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, 2018, pp. 76-80.
- [10] S. Wazir, H. A. Imran, U. Latif, U. Mujahid, and M. Bilal, "Single Board Computers (SBC): The Future of Next Generation Pedagogies in Pakistan," *arXiv preprint arXiv:2008.06576*, 2020.
- [11] A. Pajankar, "Introduction to single board computers and Raspberry Pi," in *Raspberry Pi Image Processing Programming*, ed: Springer, 2017, pp. 1-24.
- [12] R. Roy and V. Bommakanti. (2017) User Manual Odroid XU4. *Odroid Magazine*.
- [13] D. Paul. (2019) ODRROID-XU4 Tweaks: A Collection of Popular Modications. *Odroid Magazine*.
- [14] J. Rondx. (2019) Cryptocurrency Mining: Earning Verium Coins With Your ODRROID. *Odroid Magazine*.
- [15] HardKernel. (2019). *ODROID-N2*. Available: <https://www.hardkernel.com/blog-2/odroid-n2/>
- [16] RaspberryPi. (2021). *Raspberry Pi 4 Computer Model B*. Available: www.raspberrypi.org
- [17] J. Diaz, C. Munoz-Caro, and A. Nino, "A survey of parallel programming models and tools in the multi and many-core era," *IEEE Transactions on parallel and distributed systems*, vol. 23, pp. 1369-1386, 2012.
- [18] M. McCool, J. Reinders, and A. Robison, *Structured parallel programming: patterns for efficient computation*: Elsevier, 2012.
- [19] R. Chandra, L. Dagum, D. Kohr, R. Menon, D. Maydan, and J. McDonald, *Parallel programming in OpenMP*: Morgan kaufmann, 2001.
- [20] L. G. Trujillo, A. M. Rodríguez, and L. Z. Méndez, "Representación en tiempo real de señales de radar empleando Odroid XU4," *Elektron*, vol. 4, pp. 87-92, 2020.
- [21] C. b. Minitab. (2020). *Introducción a Minitab 19 para Windows*. Available: www.minitab.com

Categoría

REPORTE

Internet de las Cosas (Internet of Things, IoT)

Área Temática

Diseño e implementación de una central operativa para el control y monitoreo en el material rodante

Fernando Iglesias, Matías Sambrizzi, Ariel Lutenberg

CONICET - GICSAFe

Laboratorio Sistemas Embebidos

Facultad de Ingeniería - UBA

Buenos Aires - Argentina

lse@fi.uba.ar

Resumen—Ante la falla de sistemas como el control de puertas o la cámara frontal de registro de accidentes viales, las formaciones ferroviarias activan automáticamente el corte de tracción y el freno de emergencia. En esas circunstancias la formación queda detenida y los pasajeros deben descender a las vías si la detención no se produce en una estación ferroviaria. En este trabajo se presenta el desarrollo para Trenes Argentinos de una arquitectura modular basada en el paradigma de Internet de las Cosas (IoT) que permite visualizar y gestionar los sistemas involucrados en esas situaciones.

Palabras claves—Sistemas ferroviarios, MQTT/TLS, Type-Script, Apache Kafka

I. MOTIVACIÓN Y CONTEXTO

El sistema ferroviario de la República Argentina cuenta con una gran cantidad de formaciones ferroviarias en las que se encuentran diferentes sistemas de seguridad a bordo. Estos equipos se encargan de supervisar el correcto funcionamiento de los subsistemas críticos. Ante una falla en uno de los subsistemas, una formación ferroviaria se detiene inmediatamente por la activación automática de las señales de corte de tracción (CT) y frenado de emergencia (FE). En esta situación, el conductor debe llevar la formación a un lugar seguro para que los pasajeros puedan descender y, posteriormente, trasladarla a un taller para que pueda ser reparada.

El SAL/T, según sus siglas, Sistema de Aislamiento Limitado / Total [1], es un dispositivo del cual se cuenta con una primera versión prototipada; que se presenta como solución a las contingencias descritas anteriormente. De esta manera, el maquinista de una formación ferroviaria cuenta con la posibilidad de activar y desactivar el *modo aislado limitado*. En este modo, el equipo permite la circulación de la formación al desactivar las señales de corte de tracción y freno de emergencia generadas por los subsistemas críticos. Para que esta operación se complete de forma segura, se debe monitorear la velocidad de la formación tal que sea posible garantizar que no se supere cierto valor máximo.

A partir del desarrollo previo del prototipo, se presentan los avances en la implementación de una central operativa que centraliza los dispositivos SAL/T para su respectiva administración, configuración y monitoreo en tiempo real de la información recibida y transmitida desde una plataforma digital. El proyecto se desarrolla por CONICET-GICSAFe [2] para la empresa Trenes Argentinos [3].

Los subsistemas asociados al SAL/T, como la seguridad de puertas, el sistema de hombre vivo y la protección de coche a la deriva, son críticos debido a que, en caso de fallar, pueden ocasionar lesiones o muertes de personas e incluso generar pérdidas materiales.

La central operativa permite la administración y configuración en forma remota de los dispositivos de supervisión de seguridad de cada formación ferroviaria, la visualización de los diferentes parámetros de interés involucrados por las personas asignadas dentro de una entidad y de este modo es posible optimizar la toma de decisiones.

La sección II presenta cada una de las capas con sus respectivas tecnologías en basándose en el *stack IoT* [4]. Luego, en la sección III se exponen las conclusiones del trabajo.

II. ARQUITECTURA PROPUESTA

A partir de los modelos más relevantes que se encuentran establecidos para el desarrollo e implementación de aplicaciones de software se consideran el *stack web* y el *stack IoT*. A pesar de las similitudes entre ambos, en la aplicación el *stack web* emplea el protocolo HTTP [5] mientras que el *stack IoT* utiliza el protocolo MQTT [6]; siendo este último el más adecuado para escenarios donde son limitados los recursos como el ancho de banda y el consumo de energía.

En especial, el protocolo MQTT dispone de mensajes más livianos y además es posible transmitir y/o recibir datos en formato binario sin la necesidad de una codificación previa. También, este protocolo permite asignar niveles de calidad de servicio (QoS) [7] a los mensajes transmitidos, resultando una característica primordial en aplicaciones donde la probabilidad de pérdidas de paquetes es considerable.

La figura 1 presenta la estratificación de las capas, según el *stack IoT*, que son detalladas a lo largo de esta sección.

A. Capa de dispositivos

Sobre la base de un prototipo del Sistema de Aislamiento Limitado o Total, SAL/T [1], se desarrolló mediante el kit de desarrollo *Nucleo F429* [8], un cliente MQTT que permite la publicación de la velocidad de la formación y de los parámetros provistos por los subsistemas de falla, entre otras variables; y llegado el caso, el dispositivo pueda realizar la lectura de diferentes parámetros configurables.

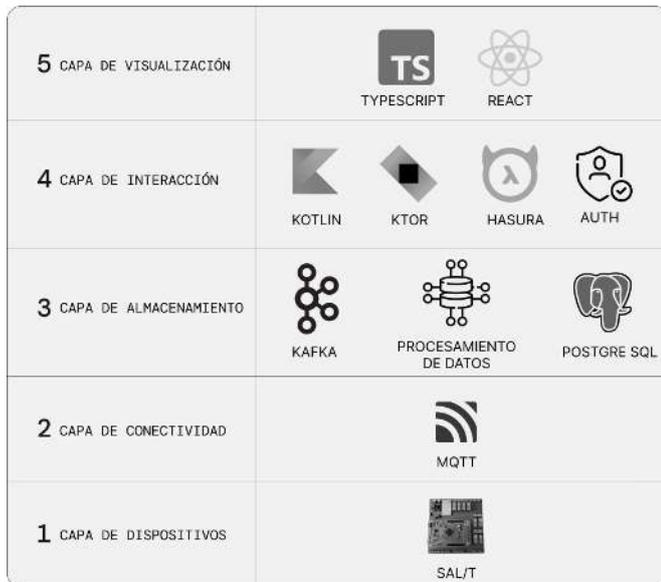


Fig. 1: Arquitectura de la Central Operativa SAL/T.

B. Capa de conectividad

Para la comunicación entre los dispositivos *SAL/T* y la capa de almacenamiento se emplea un *Broker MQTT* que oficia de orquestador entre el cliente, el dispositivo *SAL/T* que publica los mensajes y *Apache Kafka* [9], un sistema distribuido que efectúa la lectura de la información y su posterior procesamiento.

Dado que los mensajes intercambiados entre las partes contienen información sensible, se ha optado por agregar la capa de seguridad *TLS* [10]. En este sentido, resulta indispensable utilizar el certificado *X509* [11] para prevenir ataques del tipo *adversary-in-the-middle* [12] y los certificados emitidos por autoridades reconocidas.

C. Capa de almacenamiento

La capa de almacenamiento se encuentra constituida por el sistema distribuido *Kafka* que se encarga de almacenar los datos de la aplicación. Entre ellos se destacan los mensajes *MQTT* enviados por los dispositivos *SAL/T*, como también la información referida a las formaciones y a los *SAL/T* que las ocupan. Además, se tienen servicios de procesamiento de datos que se encargan de adaptar la información de los tópicos en datos que posteriormente serán consumidos por los servicios que integran la capa superior.

Por otro lado, se tiene una base de datos relacional *PostgreSQL* [13] para el almacenamiento de los datos provenientes del servicio de autenticación y el motor que facilita la interacción con la plataforma web. Los conectores se encargan de insertar los datos que se quieren disponer a la capa de visualización.

En la figura 2 se observa el modelo de datos que representa a los dispositivos *SAL/T*, las formaciones y las entidades. El primero cuenta con un identificador del dispositivo en el sistema, el estado de operación y una referencia al tren en el que se desplegó el artefacto. El modelo de la formación está enriquecido con la entidad a la que pertenece y el número

DISPOSITIVOS				
id	train_id	status	subsystems	train
b8ce177e-9aa8-4606-bb83-75a58c13c396	897ba9f3-35dd-4150-94c2-523c3919f164	disabled	Close	Close

TRENES			
id	entity_id	series_number	entity
897ba9f3-35dd-4150-94c2-523c3919f164	67bc0b10-9d1f-4986-a3da-89777021cac	AXW123	Close

ENTIDADES		
id	name	description
67bc0b10-9d1f-4986-a3da-89777021cac	BCYL	Belgrano Cargas

Fig. 2: Modelo de datos de los dispositivos, trenes y entidades.

DISPOSITIVOS				
id	train_id	status	subsystems	train
b8ce177e-9aa8-4606-bb83-75a58c13c396	897ba9f3-35dd-4150-94c2-523c3919f164	disabled	Close	Close

SUBSISTEMAS DE FALLA				
id	fail_system	port_number	device_id	status
9c667501-5d1b-4677-af09-7f5e8a3a47a	ALS	0	b8ce177e-9aa8-4606-bb83-75a58c13c396	disabled
28f4480c-cc44-d2677b-36e340512281	HASLER	2	b8ce177e-9aa8-4606-bb83-75a58c13c396	disabled
896bce05-0907-4332-8018-082a2085700	BRAXES	1	b8ce177e-9aa8-4606-bb83-75a58c13c396	disabled

Fig. 3: Modelo de datos de los dispositivos y sus subsistemas de fallas.

de serie del tren. Por último, las entidades están compuestas por nombre y descripción. En la figura 3 se aprecian los subsistemas de fallas relacionados con el dispositivo *SAL/T*. Su modelo está integrado con el nombre del subsistema, el identificador del dispositivo al que está asociado y el estado actual del mismo.

D. Capa de interacción

La capa de interacción se encuentra conformada por tres unidades. Por un lado, se ha desarrollado en el lenguaje *Kotlin* [14] en conjunto con el *framework Ktor* [15], un microservicio de autenticación donde es posible la gestión de los usuarios con sus respectivos roles y también el *backend*. Entre sus tareas principales se encuentran las relacionadas con el protocolo *MQTT* para la actualización de los certificados que brindan seguridad a las comunicaciones, mediante el uso de la capa de seguridad *TLS*, y la indexación de los eventos que se publiquen en tiempo real; como la configuración general que permite el funcionamiento integral de los servicios y módulos dispuestos en el sistema.

Además, se dispone del motor *Hasura* [16], el cual se encuentra conectado a una base de datos relacional (*PostgreSQL*) basado en el lenguaje de consulta estructurado *SQL* [17], que permite exponer una *API GraphQL* [18] a aquellos clientes que deseen obtener una visualización del modelo de datos propuesto en que se conjuga la información de cada formación ferroviaria con su correspondiente dispositivo *SAL/T* y diseño destinado al *frontend*. Más aún, el enlace permite realizar modificaciones y hasta remociones de las columnas propuestas en las tablas de cada esquema dentro de la base de datos.

E. Capa de visualización

En esta capa se ha utilizado el lenguaje de programación *TypeScript* [19] junto con la biblioteca gráfica *React* [20]

para brindar una web reactiva en la que se elaboraron los formularios, las tablas y los paneles a partir de la explotación de la información almacenada en la base de datos. Cabe destacar que la plataforma web se encuentra condicionada por el rol y la entidad a la que pertenece cada usuario, de modo que, cada formación ferroviaria tendrá un operador asignado, el cual podrá modificar los parámetros correspondientes con el dispositivo asociado.

En la figura 4 se puede apreciar el panel de control de la central operativa al que tendrán acceso los operadores de cada entidad. En el mismo se observan distintas tarjetas con el estado de cada subsistema de fallas y del tren, como así también un velocímetro que informa la velocidad de la formación.

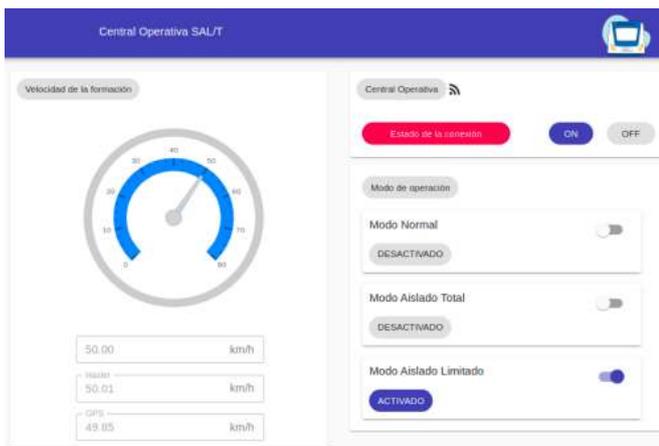


Fig. 4: Panel de control de la central operativa.

III. CONCLUSIONES

Los avances obtenidos permiten confirmar que el diseño propuesto ha sido el apropiado. La arquitectura implementada facilitó la disponibilidad y la explotación de la información proveniente de los dispositivos SAL/T, así como del microservicio correspondiente a la autenticación de usuarios. De este modo, se posibilitó la visualización de los integrantes y los

dispositivos de cada entidad, ofreciendo una solución sencilla y escalable.

En la actualidad, se encuentra en desarrollo propiciar al operario el envío de comandos para el control de los dispositivos activos que este tenga asignados, actualizar la configuración de los parámetros de cada dispositivo y finalmente, brindar la seguridad en las comunicaciones, agregando la capa *TLS* en el protocolo *MQTT*.

REFERENCIAS

- [1] 'Sistema de supervisión de la seguridad del material ferroviario utilizando patrones de diseño', Ivan Mariano Di Vito, Pablo Gomez, Ariel Lutenberg, Libro de trabajos del CASE2019, Congreso Argetino de Sistemas Embebidos, Santa Fe, Argentina. (2019).
- [2] CONICET-GICSAFe (June 2022). [Online]. Disponible: <https://sites.google.com/view/conicet-gicsafe/inicio>
- [3] Trenes Argentinos (June 2022). [Online]. Disponible: <https://www.argentina.gob.ar/transporte/trenes>
- [4] 'Internet of Things for Measuring Human Activities in Ambient Assisted Living and e-Health', Amine Rghioui, Sendra Sandra, Lloret Jaime, Oumnad Abedlmajid, Network Protocols and Algorithms. (2016).
- [5] HTTP (June 2022). [Online]. Disponible: <https://developer.mozilla.org/es/docs/Web/HTTP>
- [6] MQTT (June 2022). [Online]. Disponible: <https://mqtt.org/>
- [7] QoS (June 2022). [Online]. Disponible: <https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/>
- [8] NUCLEO-F429ZI (June 2022). [Online]. Disponible: <https://www.st.com/en/evaluation-tools/nucleo-f429zi.html>
- [9] Kafka (June 2022). [Online]. Disponible: <https://kafka.apache.org/>
- [10] 'TLS/SSL - MQTT Security Fundamentals' (2022). [Online]. Disponible: <https://www.hivemq.com/blog/mqtt-security-fundamentals-tls-ssl/>
- [11] X509 (June 2022). [Online]. Disponible: <https://www.ssl.com/faqs/what-is-an-x-509-certificate/>
- [12] Adversary-in-the-Middle (June 2022). [Online]. Disponible: <https://attack.mitre.org/techniques/T1557/>
- [13] PostgreSQL (June 2022). [Online]. Disponible: <https://www.postgresql.org/>
- [14] Kotlin (June 2022). [Online]. Disponible: <https://kotlinlang.org/>
- [15] Ktor (June 2022). [Online]. Disponible: <https://ktor.io/>
- [16] Hasura (June 2022). [Online]. Disponible: <https://hasura.io/>
- [17] SQL (June 2022). [Online]. Disponible: <https://www.w3schools.com/sql/>
- [18] API GraphQL (2022). [Online]. Disponible: <https://graphql.org/>
- [19] TypeScript (June 2022). [Online]. Disponible: <https://www.typescriptlang.org/>
- [20] React (June 2022). [Online]. Disponible: <https://reactjs.org/>

Congreso Argentino de Sistemas Embebidos
CASE 2022

Libro de trabajos en modalidad artículo, foro tecnológico y reporte

ISBN 978-987-46297-9-1



9 789874 629791