



Jornadas Sarteco

17-19 de junio de 2024 - A Coruña

AVANCES EN ARQUITECTURA Y
TECNOLOGÍA DE COMPUTADORES



Sociedad
Científica
Informática
de España



EDITADO POR:

MARGARITA AMOR
ROBERTO R. EXPÓSITO
BASILIO B. FRAGUELA
ARTURO GONZÁLEZ-ESCRIBANO
MARÍA J. MARTÍN
GABRIEL RODRÍGUEZ



GRUPO ARQUITECTURA
DE COMPUTADORES
UNIVERSIDADE DA CORUÑA
GAC.UDC.ES



UNIVERSIDADE DA CORUÑA

Avances en arquitectura y tecnología de computadores

Actas de las Jornadas SARTECO 2024

A Coruña, 17 a 19 de Junio de 2024

Avances en arquitectura y tecnología de computadores
Actas de las Jornadas SARTECO 2024

Editores: Margarita Amor, Roberto R. Expósito, Basilio B. Fraguera, Arturo González-Escribano,
María J. Martín, Gabriel Rodríguez

(c) 2024, Jornadas SARTECO

ISBN: 978-84-09-61749-4

A Coruña, 2024



Prólogo

La Sociedad de Arquitectura y Tecnología de Computadores (SARTECO) presenta una vez más las Jornadas SARTECO, las cuales integran las XXXIV Jornadas de Paralelismo (JP2024) y las VIII Jornadas de Computación Empotrada y Reconfigurable (JCER2024). Este año las Jornadas SARTECO se celebran en el marco del VII Congreso Español de Informática, CEDI 2024.

Además, en el contexto de las jornadas SARTECO se realizarán las siguientes actividades:

- Octava edición del Concurso “Tu Tesis en Tres Minutos” (“Your PhD Thesis in Three Minutes” competition) que pretende premiar los mejores trabajos de tesis recientes en el área
- Cuarta edición del Concurso “Tu Trabajo Fin de Máster en Tres Minutos” que pretende hacer lo propio con los mejores Trabajos Fin de Máster.
- Dos *keynotes* impartidas por Rosa Badia (Barcelona Supercomputing Center-BSC) y Ignacio M. Llorente (OpenNebula).
- Mesa Redonda: “La informática en la computación cuántica” - Organizada por QSpain, moderada por Carmen G. Almudéver presidenta de QSpain.
- Presentación del computador cuántico del CESGA: “QMIO. Infraestructura de 32 cubits para la investigación” por Andrés Gómez Tato, Administrador de aplicaciones y proyectos del CESGA.

En esta edición de las Jornadas contamos con un excelente programa de sesiones técnicas con un total de 119 artículos repartidos entre 89 en las JP y 30 en las JCER. A fecha de hoy, el número de asistentes sin incluir los voluntarios es de 190 personas. Un gran porcentaje de los asistentes son profesores, y un 41 % son estudiantes.

Aprovecho también para mostrar mi sincero agradecimiento a nuestros sponsors por su generosa contribución:

- Las Jornadas SARTECO-CEDI han contado con una ayuda para la organización de congresos por parte del Vicerrectorado de Investigación y Transferencia de la Universidade da Coruña.

La celebración conjunta de estas actividades y eventos constituye un referente nacional imprescindible para la comunidad científica agrupada en SARTECO. Estas jornadas facilitan colaboraciones y sinergias entre grupos y dan a nuestra comunidad una mayor visibilidad social para potenciar nuestras oportunidades de transferencia tecnológica a la industria.

Desde la organización de estas nuevas Jornadas SARTECO, os deseamos a todos y a todas, un fructífero trabajo y una placentera estancia en A Coruña.

Basilio B. Fraguera
23 de mayo de 2024
A Coruña

Comités

Comité de Dirección de las Jornadas SARTECO

Francisco Quiles Flor (UCLM, Presidente)
Víctor Viñals Yúfera (UNIZAR, Vicepresidente)
Gracia Ester Martín Garzón (UAL, Secretaria)
Francisco Tirado Fernández (UCM, Presidente de Honor)

Comité de Organización de las Jornadas SARTECO

Margarita Amor (UDC)
José M. Andión (UDC)
Roberto L. Castro (UDC)
Ramón Doallo (UDC)
Roberto R. Expósito (UDC)
Basilio B. Fraguela (UDC)
Patricia González (UDC)
Arturo González-Escribano (UVa)
María J. Martín (UDC)

Comité de Coordinación JCER'2024

Jesús Barba Romero (UCLM)
Ignacio Bravo (UAH)
David Castells (UAB)
Sergio Cuenca-Asensi (UA)
Antonio Martínez-Álvarez (UA)
Gabriel Mújica (UPM)
Joaquín Olivares (UCO)
José Manuel Palomares (UCO)
Juan Suardiaz Muro (UPCT)
Miguel Ángel Vega Rodríguez (UNEX)
Gustavo D. Sutter (UAM)
José Torres País (UV)

Comité de Programa JP'2024

Diego Andrade (UDC)
Margarita Amor (UDC)
Diego Darriba (UDC)
Jonatan Enes (UDC)
Basilio B. Fraguela (UDC)
Patricia González (UDC)
María J. Martín (UDC)
Roberto R. Osorio (UDC)
Xoán C. Pardo (UDC)
Gabriel Rodríguez (UDC)
Guillermo L. Taboada (UDC)
Diego Teijeiro (UDC)

Comité de Programa JCER'2024

Jesús Barba Romero (UCLM)
Francisco Barranco Expósito (UGR)
Manuel Alejandro Barranco González (UIB)
Francisco Jesús Bonin Font (UIB)
Julián Caba Jiménez (UCLM)
Daniel Carreres Prieto (CUD)
Joaquín Cuéllar (UCO)
José Antonio de la Torre las Heras (UCLM)
Juan Antonio Gómez Pulido (UNEX)
Joaquín Gracia (UPV)
José María Granado Criado (UNEX)
Juan Antonio Holgado Terriza (UGR)
Miren Illarramendi Rezabal (MU)
José Manuel Lanza (UAH)
Fernando León (UCO)
Iago Martínez (UCO)
Màrius Montón (UAB)
Alberto Ortiz Rodríguez (UIB)
José Manuel Palomares (UCO)
Fernando Pardo Carpio (UV)
María Begoña del Pino Prieto (UGR)
Jorge Portilla (UPM)
Julián Proenza Arenaz (UIB)
Lluís Ribas Xirgo (UAB)
Fernando Rincón Calle (UCLM)
Antonio Rodríguez Almeida (ULPGC)
Roberto R. Osorio (UDC)
Jesús Rubio Aparicio (UPCT)
Juan Carlos Ruiz (UPV)
Miguel Ángel Vega Rodríguez (UNEX)
Félix Jesús Villanueva Molina (UCLM)

Comité T3M24 y TFM3M24

Francisco Quiles Flor (UCLM) (Presidente)
Gracia Ester Martín Garzón (UAL) (Secretaria)
Francisco Tirado Fernández (Presidente de Honor)
Dora Blanco Heras (USC)
Vicente J. Blanco Pérez (ULL)

Indice

Ponencias invitadas

- Simplifying the life-cycle management of complex application workflows 3
Rosa M. Badia
- Tecnológica Europea para el Continuo Cloud-Edge de Próxima Generación 5
Ignacio M. Llorente

Parte 1.- XXXIV Jornadas de Paralelismo

Aplicaciones

- Aceleración de la fase de preproceso del flujo de trabajo MVS mediante procesadores gráficos 11
Roberto Díaz-Cano Lozano, Francesc Folch, Enrique S. Quintana-Ortí, Pedro Alonso-Jordá
- Arquitectura de Software Paralela Optimizada para la Clasificación de Materiales en Sistemas Industriales 21
Otoniel López-Granado, Miguel Martínez-Rach, Héctor F. Migallón, Carlos Pérez Vidal, Ricardo Morales
- Arquitectura de generación de modelos para Gemelos Digitales 29
Fernando Vázquez-Novoa, Riccardo Cecco, Jorge Ejarque, Rosa M. Badia
- cuElSe y cuExCuSe: seguimiento pupilar de alto rendimiento mediante GPUs 37
Arturo Vicente-Jaén, Juan Mompeán, Juan L. Aragón, Pablo Artal
- Algoritmo ACO-VNS aplicado a problemas de optimización combinatoria binarios en biología de sistemas 45
Roberto Prado-Rodríguez, Patricia González, Julio R. Banga, Ramón Doallo
- Aceleración del Ajuste Automático y Personalizado de Modelos gEUD para Planificación en IMRT 51
Juan José Moreno, Savíns Puertas-Martín, María L. Padilla, Juana L. Redondo, P.M. Ortigosa, Ester M. Garzón
- Accelerating Long Sequence Alignment Using Processing-In-Memory 59
Alejandro Alonso-Marín, Ivan Fernandez, Santiago Marco-Sola, Juan Gómez-Luna, Quim Aguado-Puig, Miquel Moreto
- Desarrollo de una Implementación Distribuida en el Framework Hive para Detección de Catástrofes Naturales 69
Carlos Cañada-Rostro, María B. García-Flores, Mercedes E. Paoletti, Javier Guillén, Juan M. Haut
- Aceleración de la técnica Parallel Tempering para métodos Markov Chain Monte Carlo 75
Aingeru Ramos, Jose A. Pascual, Javier Navaridas
- GAN-based data augmentation for the classification of remote sensing multispectral images 83
Víctor Barreiro, Álvaro G. Dieste, Dora B. Heras, Francisco Argüello

Adaptación del proceso de transformación wavelet para la detección de DMRs en el ADN	89
<i>Ricardo Olanda, Carlos Reaño, Mariano Pérez, Juan M. Orduña</i>	
Diseño de un sistema AIoT para la gestión de Ecosistemas Lagunares	95
<i>Virginia C. Sánchez, José M. Cecilia, Carlos T. Calafate, Martín González, José J. López-Espín</i>	
Aceleración de la codificación de vídeo mediante VVC. Predicción Intra-Frame: codificador base	103
<i>Alberto Jiménez-Ruiz, Gerardo Fernández-Escribano, Pedro Cuenca</i>	
Metasistema para Modelar el Servicio de Urgencias Hospitalarias	111
<i>Francisco Mesas, Manel Taboada, Dolores Rexachs, Alvaro Wong, Emilio Luque, Francisco Epelde</i>	
Métodos de alineación jerárquica aplicados a la inspección superficial de productos largos	119
<i>F.J. de la Calle, D.G. Lema, L. Magadán, R. Usamentiaga</i>	

Aceleradores hardware y arquitecturas heterogéneas

Rendimiento y limitaciones de diseño en la integración de un módulo IME basado en FPGA en el codificador de vídeo HEVC	129
<i>Otoniel López-Granado, Héctor F. Migallón, Manuel P. Malumbres, Estefanía Alcocer, Roberto Gutierrez</i>	
Cálculo Eficiente de la Transformada Rápida de Fourier en GPUs usando Técnicas de Comunicación Intra-Warp	137
<i>Xián García Veiga, Margarita Amor López, Adrián Pérez Diéguez</i>	
Esquemas de Orquestación Heterogénea para Inferencia Filogenética	145
<i>Sergio Santander-Jiménez, Miguel A. Vega-Rodríguez</i>	
Hardware scheduler for balanced co-execution on Integrated GPUs	153
<i>Borja Pérez, Jose Luis Bosque</i>	
Caracterización de mapeos multi-dataflow para modelos DNN dispersos en el acelerador Flexagon	161
<i>Adrián Navarro, Francisco Muñoz-Martínez, José Cano, José L. Abellán, Manuel E. Acacio</i>	
Segmentación de modelos para acelerar la inferencia en sistemas multi-TPU	171
<i>Jorge Villarrubia, Luis Costero, Francisco D. Igual, Katzalin Olcoz</i>	
Programación de múltiples dispositivos heterogéneos: El caso de Optical Flow	181
<i>Sergio Alonso Pascual, Arturo Gonzalez-Escribano</i>	
Optimización de la lectura de archivos EDF en C++: Avances para el procesamiento eficiente en GPU de grandes conjuntos de datos de EEG	189
<i>F. Romero, M. Lupión, N.C. Cruz, L.F. Romero, P.M. Ortigosa</i>	
Acelerando la multiplicación de matrices sobre un Array Reconfigurable de Grano Grueso	197
<i>María José Belda, José Miranda, Fernando Castro, Miguel Peón, Katzalin Olcoz, David Atienza</i>	

Arquitecturas del procesador, multiprocesadores y chips multinúcleo

Integración de la aritmética posit con quire en un core RISC-V	207
<i>David Mallasén, Alberto A. del Barrio, Manuel Prieto-Matias</i>	

Generación de Aceleradores Basados en Aritmética Posit con HLS <i>Raul Murillo, Alberto A. del Barrio, Guillermo Botella</i>	215
Predicción de dependencias de memoria eficaz y sensible al contexto <i>Sebastián S. Kim, Alberto Ros</i>	225
Análisis de Políticas de Reemplazo en Branch Target Buffers para Servidores <i>Pascual Abellán-Pérez, Agustín Navarro-Torres, Josué Feliu, Alberto Ros, Manuel E. Acacio</i>	233
Comparación eficiente de secuencias genómicas usando extensiones AVX-512 <i>Elena Espinosa, Ricardo Quislan, Rafael Larrosa, Oscar Plata</i>	241
Comparativa de la sensibilidad frente a soft errors según las extensiones ISA RISC-V en un procesador COTS de Western Digital <i>Daniel León, Juan C. Fabero, Juan A. Clemente</i>	251
Simulación de un algoritmo de planificación dinámica de instrucciones sin etiquetas ni renombramiento dinámico. <i>Belen López-Salamanca, Pablo Sánchez-Cuevas, Manuel Rivas-Pérez, J. L. Sevillano-Ramos, Fernando Díaz-del-Río</i>	259

Arquitecturas del subsistema de memoria y almacenamiento

Tolerancia a fallos en el sistema de ficheros paralelo Expand Ad-Hoc <i>Darío Muñoz-Muñoz, Diego Camarmas-Alonso, Félix García-Carballeira, Alejandro Calderón-Mateos, Jesús Carretero</i>	271
iDA: Preservando el acceso a bases de datos a muy largo plazo <i>Eladio Gutiérrez, Sergio Romero, Oscar Plata</i>	281
Evaluación del rendimiento de un sistema de ficheros para sistemas IoT <i>Elías del-Pozo-Puñal, Félix García-Carballeira, Diego Camarmas-Alonso, Alejandro Calderón-Mateos</i>	289
Modelo de canales paralelos para la gestión eficiente de información en flujos de datos <i>Catherine A. Torres-Charles, J. L. González-Compeán, Jesús Carretero</i>	299
Optimización de la Caché en el Producto Matricial mediante Inteligencia Artificial <i>José I. Mestre, Pablo Pallarés, Héctor Martínez, Enrique S. Quintana-Ortí, Mar Catalán</i>	309
Evaluación sistemática de un Sistema distribuido de archivos ad hoc sobre aplicaciones HPC <i>Genaro Sánchez-Gallegos, Javier García-Blas, Cosmin Petre, Jesús Carretero</i>	317
Malleable I/O Applications Modeling with ElastiSim <i>Njoud Omar Almaaitah, David E. Singh, Taylan Özden, Raffaele Montella, Jesús Carretero</i>	327
Predicción de Accesos y Precarga de Memoria Caché mediante SVM y Aprendizaje Online. <i>Pablo Sánchez-Cuevas, Daniel Casanueva-Morato, Antonio M. Pérez-Peña, Enrique Piñero-Fuentes, Antonio Ríos-Navarro, Fernando Díaz-del-Río</i>	337

Educación

Utilización de Minecraft para la configuración y evaluación de servidores <i>José Luis Ávila Jiménez, Francisco Javier Rodríguez Lozano, Manuel Agustín Ortiz López</i>	349
--	-----

Mejora de las clases de problemas en la asignatura ‘Introducción a los Computadores’ - Una única Excel para examinarlos a todos <i>Luis M. Ramos, Víctor Viñals</i>	357
Servicio de laboratorio remoto para la integración del simulador CREATOR con hardware RISC-V <i>Diego Camarmas-Alonso, Félix García-Carballeira, Alejandro Calderón-Mateos, Elías del-Pozo-Puñal</i>	365
Desarrollo de un programa generador de shellcodes para la enseñanza de ciberseguridad <i>Francisco Orts, Jordi Serra-Ruiz</i>	373
Ampliando la visión de la Computación de Altas Prestaciones: Caso de estudio en soluciones IoT <i>N.C. Cruz, M. Lupión, F. Romero, E.M. Ortigosa, J. Sanjuan-Estrada, P.M. Ortigosa</i>	381
Tablon: una herramienta para gamificación y autoevaluación de trabajos prácticos. <i>Francisco J. Andújar, Yuri Torres, Arturo Gonzalez-Escribano, Javier Bastida</i>	391

Evaluación de prestaciones

Rendimiento de máquinas virtuales y contenedores: estudio de la escalabilidad de la carga <i>Belén Bermejo, Carlos Juiz, Álvaro Bueno</i>	403
Sistema Inteligente de Medida de Energía para Monitorización Síncrona en Sistemas Informáticos Distribuidos <i>Antonio Francisco Díaz, Beatriz Prieto, Juan José Escobar, Alberto Prieto</i>	409
Splash-4: Una Benchmark Suite moderna con Constructos Lock-Free <i>Eduardo José Gómez Hernández, Juan M. Cebrian, Stefanos Kaxiras, Alberto Ros</i>	419
Evaluación de Rendimiento de Plataformas Data Lakehouse: Snowflake y Databricks <i>Manuel Corujo Muíña, Ángel Regueiro Feal, Adrián Xuíz García, Ignacio García Sánchez, Guillermo López Taboada, Roberto R. Rey, Juan Touriño</i>	429
Eficiencia energética y rendimiento en VASP: un simulador de materiales a escala atómica <i>Isidoro Nieves-Pérez, Alfonso Muñoz, Francisco Almeida, Vicente Blanco</i>	437

Lenguajes, compiladores y herramientas de programación y ejecución paralela

Representación del Pipeline a través de Algoritmos Paralelos Genéricos en la Programación Paralela Heterogénea. Un análisis Comparativo <i>Mario Rossain-López, Bárbara Sánchez-Rinza, Zuriel López Sosa, Manuel I. Capel-Tuñón</i>	449
Generación Automática de Micro-kernels ARM NEON para la Multiplicación de Matrices <i>Guillermo Alaejos, Pedro Alonso-Jordá, Adrián Castelló, Enrique S. Quintana-Ortí, Francisco D. Igual, Manuel F. Dolz, Héctor Martínez</i>	459
Herramienta de gestión de entrenamientos de TensorFlow en un clúster de HPC gestionado con SLURM <i>M. Lupión, N.C. Cruz, F. Romero, J. Sanjuan-Estrada, P.M. Ortigosa</i>	469
Un nuevo enfoque para el desarrollo de aplicaciones MPI maleables en entornos HPC gestionados por planificadores de trabajos <i>Javier Fernández Muñoz, Alberto Cascajo García, Jesús Carretero</i>	475

Redes y Comunicaciones

- QVal: a novel routing algorithm for Dragonfly networks 485
Javier Navaridas, Jose A. Pascual
- Planificación de paquetes multicanal para mejorar el rendimiento del streaming de video en redes vehiculares 493
P. Pablo Garrido Abenza, Pablo Piñol Peral, Manuel P. Malumbres, Otoniel López-Granado
- Análisis de prestaciones de primitivas colectiva de uno a varios en redes DragonFly con cargas de trabajo de exaescala 503
José Duro, Adrián Castelló, María Engracia Gómez, Julio Sahuquillo, Enrique S. Quintana-Ortí
- Análisis de Interferencias en la Red de Interconexión entre Aplicaciones Exascale 513
Alejandro Baviera, Max Kaidanov, José Duro, María Engracia Gómez, Julio Sahuquillo
- Estudio Comparativo de tecnologías LPWAN para entornos rurales 521
María Ángeles Amador, Celia Garrido-Hidalgo, Luis Roda-Sanchez, Teresa Olivares
- Modelado de algoritmos de encaminamiento multicamino en un simulador basado en OMNeT++ utilizando diferentes topologías de red 531
Carlos Medrano Navalón, Gabriel Gómez-López, Jesús Escudero-Sahuquillo, Pedro Javier García, Francisco J. Quiles
- Una solución híbrida para proporcionar control de flujo de extremo a extremo y control de congestión en redes de interconexión de altas prestaciones 539
Alberto Merino, Jesús Escudero-Sahuquillo, Pedro Javier García, Francisco J. Quiles, Fei Chen, Yunping Lyu, Long Yan, Jose Duato
- Redes LoRaWAN para Comunicación Extraterrestre 547
Alejandro Romero, María Ángeles Amador, Celia Garrido-Hidalgo, Luis Roda-Sanchez, Teresa Olivares

Plataformas distribuidas y Deep Learning

- Redes convolucionales basadas en escalado uniforme de dimensiones para la clasificación de imágenes multiespectrales 555
Nicolás Vilela-Pérez, Álvaro G. Dieste, Dora B. Heras, Francisco Argüello
- Despliegue de una herramienta de aprendizaje federado sobre una infraestructura distribuida de bajo coste 565
Víctor Hidalgo Izquierdo, Carmen Carrión, Blanca Caminero
- Mejora del Entrenamiento de Redes Neuronales Distribuidas a través de Comunicaciones entre Nodos 575
Sergio Moreno-Álvarez, Mercedes E. Paoletti, Juan M. Haut
- Estrategias para la mejora de gestión de memoria en algoritmos de aprendizaje profundo para el realce de contraste de imágenes radiológicas de alta resolución 581
Daniel A. Rodríguez López, Daniel Sanderson, Javier García-Blas, Manuel Desco, Mónica Abella

Computación en el borde

- Arquitectura para la orquestación de datos en el cómputo continuo 591
Dante D. Sánchez-Gallegos, Diana Carrizales-Espinoza, Jesús Carretero, J. L. González-Compeán

Infraestructura autoconfigurable de red de sensores inalámbricos para computación en el borde <i>Daniel Rodríguez, Andrés Rodríguez, Oscar Plata</i>	599
Cuantización y Poda en Edge Computing: Estrategias para Modelos de Deep Learning Eficientes <i>Nicolás Hernández, Francisco Almeida, Vicente Blanco</i>	609

Computación cuántica

Mapeo de algoritmos cuánticos en arquitecturas modulares de computación cuántica <i>Anabel Ovide, Santiago Rodrigo, Medina Bandic, Sebastian Feld, Sergi Abadal, Eduard Alarcón, Carmen G. Almudever</i>	621
Benchmarking Quantum Computers through Fourier Series Loaders <i>Oluwatosin Odubango, Diego Andrade, Basilio B. Fraguera, Andrés Gómez</i>	629
Algunos experimentos en la detección de enfermedades respiratorias con procesamiento de señales sonoras y quantum machine learning <i>Diego García-Vega, Samuel González-Castillo, Elías F. Combarro, José Ranilla</i>	643
Circuitos cuánticos para calcular la distancia de Hamming <i>Laura M. Donaire, Francisco Orts, Gloria Ortega, Elías F. Combarro, Ignacio F. Rúa, Ester M. Garzón</i>	649
Evaluación de técnicas de mitigación de errores cuánticos en procesadores NISQ <i>Laura Rodríguez-Soriano, Handy Kurniawan, Francisco García-Herrero, Carmen G. Almudever</i>	659
Fidelidad de la suma de errores independientes en computación cuántica <i>J. Lacalle, L.M. Pozo Coronado, R. Martín-Cuevas</i>	667
Contracción en paralelo de redes de tensores mediante comunidades <i>Alfred M. Pastor, Jose M. Badia, Maribel Castillo</i>	675
Heurísticas para la computación cuántica en el problema 3-SAT <i>Jose J. Paulet, Luis F. LLana, Hernan I. de la Cruz, Mauro Mezzini, Fernando Cuartero, Fernando L. Pelayo</i>	685
Revisión sistemática de la literatura en computación cuántica para el procesamiento de imágenes <i>Santiago Merino Bajo, Lidia Sánchez-González</i>	695
Backtesting de algoritmos de computación cuántica para la optimización de carteras <i>Ginés Carrascal, Guillermo Botella, Alberto A. del Barrio</i>	705
Hungarian Qubit Assingment para el Mapeo Optimizado de Circuitos Cuánticos en Arquitecturas de Múltiples Núcleos <i>Pau Escofet, Anabel Ovide, Carmen G. Almudever, Eduard Alarcón, Sergi Abadal</i>	719

Eficiencia energética

GreenDVFS: un controlador de frecuencia basado en la carga de trabajo <i>Darong Huang, Luis Costero, David Atienza</i>	729
Modelado automatizado y flexible del consumo de energía de la CPU <i>Tomé Maseda, Jonatan Enes, Roberto R. Expósito, Juan Touriño</i>	739

Energy and performance analysis of GAN models applied to weather forecasting	749
<i>Daniel Suárez, Francisco Almeida, Vicente Blanco, Gustavo Marrero, Pedro Toledo</i>	

Generación, gestión y procesamiento de datos

Estandarización de Datos de Radiación Solar: Aplicación de Principios FAIR en la Plataforma METAS-PSA	759
<i>Enrique Ugedo-Egido, Antonio Juan Rubio-Montero, Hernán Asorey, Stefan Wilbert, Luis Fernando Zarzalejo-Tirado, Rafael Mayo-García</i>	
Análisis distribuido de datos sobre la marcha para simulaciones numéricas con Ray	769
<i>Xico Fernández, Bruno Raffin, Julien Bigot, Emilio J. Padrón</i>	
Optimización del preprocesamiento de datos en imágenes WSI para aplicaciones de detección temprana de cáncer	779
<i>Antonio M. Pérez-Peña, Lourdes Duran-Lopez, Juan P. Dominguez-Morales, Alejandro Linares-Barranco, José M Marrón-Esquivel, Claudio Amaya-Rodriguez</i>	
Metodología para la generación de datos sintéticos basada en Big Data Assimilation	787
<i>Candelaria Alvarez, Jose Ibeas, Javier Balladini, Remo Suppi</i>	
Automatizando y acelerando el procesamiento de Datos LiDAR con IA	793
<i>Paula Sánchez, Irene González, Ana Cortés, Tomás Margalef, Remo Suppi, Carlos Carrillo</i>	
Evaluación de LLMs para el análisis de microtextos de redes sociales: Caso de Estudio del Mar Menor	801
<i>José Giner, José M. Cecilia</i>	

Parte 2.- VIII Jornadas de Computación Empotrada y Reconfigurable

Control de errores y sistemas en tiempo real

Protección mediante Códigos de Corrección de Errores de los pesos de una Red Neuronal implementada en Arduino	813
<i>J. Gracia-Morán, Luis J. Saiz-Adalid</i>	
Tolerancia a fallos múltiples en redes convolucionales en coma flotante de 16 bits utilizando códigos correctores de errores	823
<i>Juan Carlos Ruiz-García, David de Andrés, Luis J. Saiz-Adalid, J. Gracia-Morán</i>	
Toward Linux for safety-critical systems — The effect of PREEMPT RT on system call traces and execution times	833
<i>Markel Galarraga, Charles-Alexis Lefebvre, Jon Perez-Cerrolaza, Jose A. Pascual</i>	

IoT

Integration of blockchain in constrained IoT networks	843
<i>Jorge Señor, Jaime Señor, Gabriel Mujica, Jorge Portilla</i>	
Green Software en IoT: Estrategias para mejorar la eficiencia en plataformas Arduino	849
<i>Miguel Baños, Jorge Cancho, José Carlos Sancho, Javier Corral, Juan Antonio Rico, Felix Óscar García</i>	
Herramienta para el análisis de la orografía en redes LoRaWAN	859
<i>Pablo Doñate, Miguel Sánchez Bruno, Vicente Torres-Sanz, Julio A. Sangüesa, Piedad Garrido, Francisco J. Martínez</i>	

Procedimiento de Registro Automatizado para Dispositivos LoRaWAN	867
<i>Vicente Torres-Sanz, Julio A. Sangüesa, Felix Serna, Francisco J. Martínez, Piedad Garrido, Carlos T. Calafate</i>	

Inteligencia artificial en sistemas empuotrados

Redes Neuronales en RISC-V con aceleración vectorial	875
<i>Eugeni Casadesús, Michelle Vargas, Aleix Valdivieso, Màrius Montón</i>	
Design and Comparison of Q-learning based Transmission Power Control at the Extreme Edge of Internet of Things	881
<i>Pablo Hita, Gabriel Mujica, Jorge Portilla</i>	
Análisis de transformers para el reconocimiento de voz y evaluación de su rendimiento en sistemas de recursos limitados	887
<i>Cristina Bolaños Peño, Jesús Fernández-Bermejo Ruiz, Henry Llumiguano Solano, Javier Dorado Chaparro, Félix Jesús Villanueva Molina, Juan Carlos López López</i>	
Clasificación con retina DVS usando CNN pulsante para HW neuromórfico empuotrado	893
<i>Dagnier A. Curra-Sosa, Enrique Piñero-Fuentes, Francisco Gómez-Rodríguez, Alejandro Linares-Barranco</i>	
Aprovechamiento del audio pulsante y las redes convolucionales 1D para la clasificación de palabras	901
<i>Enrique Piñero-Fuentes, Salvador Canas Moreno, Antonio Ríos-Navarro, Alejandro Linares-Barranco</i>	
Metodología basada en protoformas lingüísticas para la detección de actividades en pacientes con diabetes tipo 2	909
<i>David Díaz Jiménez, José L. López Ruiz, Jesús González Lama, Macarena Espinilla Estévez</i>	
Exploración y análisis de datos clínicos y modelos 3D del cuerpo humano mediante el uso de técnicas de aprendizaje profundo.	917
<i>Nahuel E. Garcia-D'Urso, Jorge Azorin-Lopez, Andres Fuster-Guillo, Mario Jerez-Tallón</i>	
Estudio de la confiabilidad de una red neuronal convolucional cuantizada	925
<i>J. Gracia-Morán, Luis J. Saiz-Adalid, Juan Carlos Ruiz-García, David de Andrés</i>	

Aplicaciones de los sistemas empuotrados

Sistema móvil para la medición de PM 2.5: estudio en el área de Valencia	935
<i>Jorge Tarazona-Gala, José D. Padrón, Carlos T. Calafate</i>	
Enfoque para la optimización de sistemas de localización multiocupación basados en BLE	943
<i>José L. López Ruiz, Ángeles Verdejo Espinosa, José María Serrano Chica, Macarena Espinilla Estévez</i>	
Implementación de un modelo de memoria basado en SNN bioinspirado en el hipocampo	953
<i>Daniel Casanueva-Morato, Juan P. Domínguez-Morales, Gabriel Jimenez-Moreno</i>	
Sistema IoT para la Monitorización de Glucosa en Tiempo Real: Empoderando el Cuidado y la Prevención de Diabetes	963
<i>José L. López Ruiz, Juan F. Gaitán Guerrero, Carmen Martínez Cruz, David Díaz Jiménez, Jesús González Lama, Macarena Espinilla Estévez</i>	

Procesamiento de nubes de puntos procedentes del interior de la pala de un aerogenerador <i>Pablo A. Fuenzalida, Aurelio Bermúdez, Rafael Casado</i>	971
Diseño e Implementación de mercados eléctricos locales en comunidades energéticas fotovoltaicas PVLEM <i>X. Moreno-Vassart, F. Javier Toledo, Victoria Herranz, Vicente Galiano</i>	981
Vehículos aéreos no tripulados	
Misiones de VANT seguras mediante ajuste orográfico dinámico <i>Jamie Wubben, Carlos T. Calafate, Enrique Hernández-Orallo, Juan-Carlos Cano, Pietro Manzoni</i>	989
Localización relativa de enjambres de UAV en entornos sin GNSS <i>Asma Boulkaid, Aurelio Bermúdez, Rafael Casado, Luis Orozco-Barbosa, Abdelkrim Haqiq</i>	999
Sistema de generación de procedimientos de vuelo con curvatura progresiva <i>Wafae Azoulay, Rafael Casado, Aurelio Bermúdez, Luis Orozco-Barbosa, Abdelkrim Haqiq</i>	1009
Arquitecturas eficientes	
FPGA accelerator for ORB feature extraction <i>Amel Chafroud, David Castells-Rufas, Jordi Carrabina, Chokri Souani</i>	1019
Proposal for PRNG System to Produce Pseudorandom Sequences <i>María de Lourdes Rivas, Juan José Raygoza, Edwin Christian Becerra, Susana Ortega-Cisneros</i>	1027
Optimización Automatizada de Hiperparámetros en Pipelines de Procesamiento de Imágenes en Sistemas Empotrados mediante Técnicas Genéticas <i>José L. Mira, Jesús Barba, Julián Caba, José A. de la Torre, Fernando Rincón, Juan Carlos López</i>	1033
Docencia y sistemas ciberfísicos	
Exploring New Approach With Open Source EDA Tools <i>Emilio Isaac Baungarten-Leon, Susana Ortega-Cisneros, Cristian Torres, Uriel Jaramillo-Toral, Juan Carlos Garcia Lopez, Juan José Raygoza</i>	1043
First steps and benchmarks using a pure Rust Operating System as Tock OS <i>Joaquín Cuéllar-Padilla, Fernando León-García, Jose M. Palomares, Iago R. Martínez-Sánchez, Joaquín Olivares</i>	1051
Cómo desarrollar sistemas embebidos a partir de modelos en máquinas de estado extendidas <i>Lluís Ribas Xirgo, Màrius Montón, A. Josep Velasco González</i>	1059
Indice de autores	1067

Misiones de VANT seguras mediante ajuste orográfico dinámico

Jamie Wubben¹, Carlos T. Calafate¹, Enrique Hernández-Orallo¹,
Juan-Carlos Cano¹, y Pietro Manzoni¹

Resumen— El campo de los Vehículos Aéreos No Tripulados (VANTs) ha progresado mucho en los últimos años. Los VANTs se utilizan ahora para muchas aplicaciones, a menudo realizando misiones de manera autónoma. Una característica comúnmente implementada en un vuelo automático es seguir una misión a una altitud estable. Sin embargo, esta altitud casi siempre se mide en el punto de despegue, por lo que no tiene en cuenta la variabilidad del perfil del terreno. Se trata de una cuestión crítica y peligrosa porque, si el nivel del terreno cambia bruscamente (por ejemplo, en regiones montañosas, o en edificios en una ciudad), puede provocar accidentes, o se puede alcanzar una altitud elevada no deseada (o ilegal). Nuestro objetivo en este trabajo es ofrecer una solución que logre mantener una altitud constante sobre el nivel del suelo. Para ello, utilizaremos los Modelos Digitales de Elevación (MDE) disponibles. Estos modelos, que contienen la elevación del terreno, nos sirven de entrada a nuestro algoritmo, el cual ajustará dinámicamente la altitud del VANT para que se mantenga casi constante respecto al suelo. Los resultados han demostrado que el método propuesto es efectivo, al tiempo que se introduce apenas un ligero incremento del tiempo de vuelo y del consumo de batería, ambos de manera proporcional a la irregularidad del terreno.

I. INTRODUCCIÓN

En la última década, el interés por el uso de vehículos aéreos no tripulados (VANTs) no ha dejado de crecer. Hoy en día, los VANTs, coloquialmente llamados drones, son una herramienta habitual para numerosas aplicaciones, como tareas de inspección, cartografía, seguridad, agricultura, cine, carreras, y muchas más [1].

En muchas aplicaciones, el VANT vuela automáticamente de un punto de ruta, registrado mediante el Sistema de Posicionamiento Global (GPS), a otro punto. A lo largo de su ruta, que a menudo se define utilizando puntos de referencia de la misión, el VANT tiene que mantener su altitud constante. Se trata de una función integrada habitual en muchos VANTs de gama alta, de la que se encarga el controlador de vuelo, es decir, un pequeño ordenador de a bordo con múltiples sensores que regula la velocidad de las hélices para cumplir sus objetivos de vuelo.

El controlador de vuelo de código abierto más utilizado se llama Pixhawk [2], que a menudo utiliza el firmware Ardupilot [3]. Mediante el uso de Ardupilot (o cualquier otro firmware controlador de vuelo de esa manera), podemos mantener una altitud constante. Sin embargo, esta altitud constante suele estar referenciada al punto de despegue del VANT (es decir, desde el origen). Como se muestra en la Figura 1,

esto es problemático si el terreno no es plano. Esto se debe a que, si el terreno se eleva, existe la posibilidad de que el VANT se estrelle contra el suelo. Por otro lado, si el terreno descende, el dron volará a más altura; esto puede disminuir el rendimiento de los sensores del VANT (por ejemplo, la alimentación de la cámara), o hacer que supere la restricción legal relativa a la altitud máxima de vuelo (por ejemplo, en Europa es de 120 metros [4]).

Por lo tanto, muchas aplicaciones requieren mantener una altitud constante sobre el nivel del suelo (AGL: Above Ground Level en inglés). En este trabajo, nos centramos en proporcionar esta funcionalidad porque actualmente no está implementada en los controladores de vuelo disponibles comercialmente. En concreto, nuestra solución se basa en Modelos Digitales de Elevación (MDE), es decir, archivos con datos topográficos. Mediante el uso de estos modelos de elevación del terreno, fácilmente disponibles, podemos manipular el dron de tal manera que se intente mantener a una altitud constante sobre el nivel del suelo. Debemos señalar que ya existen aplicaciones de terceros que tienen en cuenta la topografía en la planificación de misiones (por ejemplo, FlyLitchi [5]). Sin embargo, muchas aplicaciones requieren que los drones puedan tomar decisiones ad-hoc y desviarse del plan de vuelo original (por ejemplo, para evitar colisiones, o debido a un cambio en el itinerario). Por ello, en este trabajo optamos por un enfoque más holístico, de modo que concedemos a las aplicaciones actuales y futuras esta libertad, además de simplificar las tareas del operador.

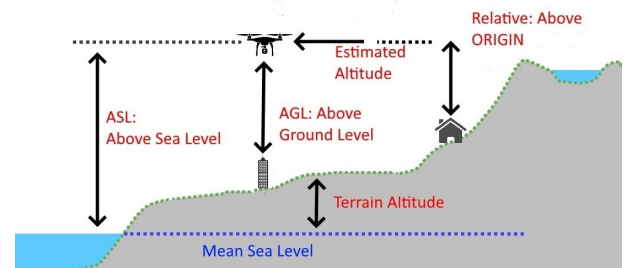


Fig. 1: Los diferentes tipos de altitud.

Este trabajo de investigación está relacionado con dos áreas principales: los drones y la topografía. Estas áreas tienen una estrecha relación, ya que son numerosas las aplicaciones en las que el uso de drones ha permitido mejorar la topografía y la precisión de los modelos de elevación, como puede verse en [6].

En cuanto a la investigación dedicada a estudiar esta relación, es decir, a utilizar la topografía para

¹Computer engineering department (DISCA). Universitat Politècnica de Valencia (UPV) e-mail: {jwubben, calafate, ehernandez, jucano, pmanzoni}@disca.upv.es

ayudar a los vuelos de drones, es aún escasa. El principal uso de la topografía en un vuelo autónomo es trazar rutas y evitar colisiones. Por el momento, su principal campo de aplicación consiste en vuelos pre-programados con altitudes fijas y constantes, pero sin tener en cuenta los obstáculos. Este enfoque simplificado es aplicable en muchos casos en los que el terreno es mayoritariamente llano, y se utiliza ampliamente en aplicaciones como la agricultura de precisión [7].

Sin embargo, los vuelos totalmente autónomos, en los que las aeronaves están dotadas de capacidad de decisión y selección de rutas, no están lejos de convertirse en una realidad. Ya podemos encontrar trabajos que hablan de su potencial y viabilidad como medio de transporte [8]. Otros trabajos [9] defienden su utilidad como solución de transporte de sangre para transfusión en situaciones de emergencia, dada la rapidez y eficacia de estas aeronaves.

Además, los drones como medio de transporte para personas son ya una realidad, y la topografía es el elemento clave para conseguirlo. En [10] se describió uno de los primeros trabajos de investigación que incorporaba la topografía para la evitación autónoma de colisiones. Esta investigación partía de un marco teórico del IEEE [10], y fue validada por las Fuerzas Aéreas de EE.UU. en un F-16 [11]. [12].

En [13], Muliadi et al. proponen una red neuronal artificial que puede utilizarse como controlador de altitud de VANTs. Comparan su método con el sistema de control PID tradicional. El sistema de control PID tradicional implica expresiones complejas, que son difíciles de resolver analíticamente. Su enfoque supera este problema y, como demuestran (en simulación), con el uso de su red neuronal se puede lograr un mejor rendimiento.

La investigación que define la base teórica del proyecto puede encontrarse en [14]. En base a ella, nuestro trabajo estudia y establece, mediante formulación matemática, la viabilidad de utilizar modelos de elevación como fuente de datos para el vuelo autónomo de drones.

El resto de este trabajo se organiza de la siguiente manera: en Sección II explicamos el formato de los DEMs, dónde pueden obtenerse, y qué representan. En Sección III, explicamos en detalle la solución propuesta. A continuación, evaluamos nuestra propuesta en la Sección IV, donde simulamos un amplio conjunto de casos de uso, incluida una complicada región montañosa. Por último, en la Sección V, resumimos nuestros hallazgos, y sugerimos futuros trabajos prometedores.

II. MODELOS DIGITALES DE ELEVACIÓN (MDE)

Un Modelo Digital de Elevación (MDE) consiste en una base de datos que representa la altura sobre el nivel del mar de un lugar determinado. Un MDE se caracteriza por su versatilidad en términos de precisión/compresión. A pesar de que existen diferentes formatos, el más común divide una superficie terrestre en cuadrados de tamaño regular. Para cada cuadrado, se determina el valor de altitud más alto, el cual se

guarda en una gran matriz. En la cabecera del archivo MDE se indican las coordenadas GPS del origen, así como el tamaño de cada cuadrado (también llamado resolución). Dado que es fácil recuperar el número de fila y columna al leer el archivo MDE, se puede calcular fácilmente la localización GPS de cada casilla (es decir, sumando el desplazamiento desde el origen). De este modo, se puede representar una superficie relativamente grande en un archivo pequeño. El tamaño del fichero MDE dependerá de la resolución, pero, para dar una idea general, utilizando una resolución (alta) de 4 m^2 para cada cuadrado, un área de 500 km^2 puede representarse en un fichero de solo 250 MB. Teniendo en cuenta que estos requisitos de almacenamiento no son un problema hoy en día, y que los VANTs solo pueden volar unos pocos kilómetros de todos modos, el tamaño no representa un problema.

Los archivos MDE están fácilmente disponibles en Internet. Para nuestra investigación, hemos utilizado los MDE proporcionados por el gobierno español [15]. Como se puede observar, en este sitio (y en general), existen dos tipos de MDE (ver Figura 2): tenemos el Modelo Digital del Terreno (MDT), que representa el nivel del suelo, y el Modelo Digital de Superficie (MDS), que también incluye edificios, árboles, etc. Es obvio que los MDS representan el mundo real con mayor precisión (sobre todo en las ciudades), pero también se quedan obsoletos mucho más rápido. Dado que el formato de los archivos MDT y MDS es muy similar, nuestra solución funciona tanto con archivos MDT como MDS. No obstante, para nuestros experimentos hemos optado por trabajar únicamente con archivos MDT.

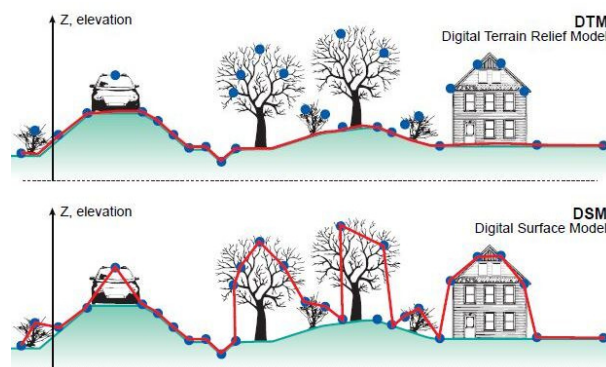


Fig. 2: La diferencia entre MDT y MDS.

Como ya se ha dicho, para crear un MDE es necesario medir la elevación del terreno (con respecto al nivel del mar). Existen muchos métodos para medir la elevación del terreno, pero aquí solo destacaremos algunos de ellos:

1. **Mediciones topográficas convencionales:** Para estos levantamientos se utilizan equipos convencionales como taquímetros ópticos o láser, así como instrumentos de nivelación. Con este método se puede obtener un MDE muy detallado. Sin embargo, es un proceso largo y costoso, por lo que solo se utiliza para zonas pequeñas.
2. **GPS cinemático:** En este método, se monta un receptor GPS en un vehículo que se desplaza

sobre el terreno. Con el uso de los datos de altitud del GPS se puede cubrir un área relativamente grande.

3. **Imágenes ópticas por satélite:** Con el uso de imágenes de satélite, se puede hacer una estimación de la altitud del nivel del terreno. La ventaja de este método es que cubre un área enorme rápidamente; sin embargo, la resolución es mucho menor que la de los métodos mencionados anteriormente.

III. SOLUCIÓN PROPUESTA

El objetivo principal de nuestro trabajo es desarrollar un marco que permita a un VANT mantener una altitud constante sobre el nivel del suelo. Esto es importante porque, si no se tiene en cuenta, el VANT podría estrellarse contra el suelo si el nivel de este sube. Por otro lado, si el nivel del suelo disminuye, la altitud relativa del VANT aumentará. Esta mayor altitud relativa puede influir en el rendimiento de los sensores del VANT (por ejemplo, la detección de objetos desde una cámara), o el VANT puede superar la altitud legalmente permitida (120 metros en Europa [4] y en la mayoría de los países del mundo).

Para abordar y resolver este problema, proponemos una solución que hace que un VANT ascienda o descienda automáticamente mientras sigue una misión para adaptarse al perfil del terreno. Antes de iniciar el vuelo, se cargan en el VANT el archivo de la misión (que consta de waypoints GPS) y el archivo MDE. Durante el vuelo, el VANT utilizará el archivo de la misión para determinar su dirección, y el archivo MDE para ajustar su altitud. Es importante señalar que el cambio de altitud se calcula en tiempo real, y no antes del inicio de la misión. Este enfoque nos permite: (i) emplear menos tiempo en el cálculo al inicio, (ii) realizar cambios en la misión sin preocuparnos por el perfil del terreno, y (iii) hacer que nuestro método sea compatible con otros enfoques (futuros), que utilizan sensores en lugar del archivo MDE para determinar la altitud relativa al suelo.

A. Modelo

En este trabajo proponemos un sistema de control que controle la velocidad del VANT en el eje z (o eje arriba/abajo). Este sistema de control tendrá como objetivo minimizar el error entre la altitud de vuelo deseada y la actual, manteniéndola constante respecto al terreno. La elevación del terreno, procedente del fichero MDE, se utiliza como señal de referencia. La altitud relativa del dron la determina el controlador de vuelo, que se basa en datos GNSS y barométricos. El error, que se intenta minimizar, viene determinado por la diferencia entre la altitud de referencia y la altitud relativa del dron:

$$e(t) = \text{altitud_de_referencia}(t) - \text{altitud_relativa}(t) \quad (1)$$

La tarea de nuestro controlador es calcular una salida que minimice este error. En nuestro sistema, la salida del controlador es la velocidad vertical [m/s].

En teoría, esos valores pueden oscilar entre $-\infty$ y $+\infty$; en la práctica, sin embargo, los motores eléctricos tienen una velocidad máxima que limitará el empuje. Por supuesto, diferentes motores tienen diferentes velocidades máximas. Sin embargo, a no ser que nuestro campo sean las carreras de drones, la mayoría de los operadores prefieren una velocidad vertical (relativamente) lenta, principalmente por razones de seguridad y control. Por lo tanto, hemos limitado la salida de nuestro control al rango ± 5 m/s.

Un controlador PID general puede describirse mediante la siguiente ecuación:

$$u(t) = K_p \cdot e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (2)$$

Así pues, la señal de salida $u(t)$ viene dada por la suma de tres términos, todos ellos dependientes de la señal de error: 1) el término proporcional (P), que corregirá el error actual; 2) el término integral (I), que crecerá con el tiempo y, por tanto, puede garantizar que el error en estado estacionario se haga cero; y 3) el término derivativo (D), que permite predecir el error futuro y, por tanto, aumenta la estabilidad del sistema. Como en nuestra aplicación el nivel del suelo (es decir, la señal de referencia) cambiará con el tiempo, no tenemos que preocuparnos por un error de estado estacionario. Por lo tanto, eliminamos el término integral (estableciendo $K_i = 0$), lo que nos permitió simplificar la ecuación 2 a un controlador PD como sigue:

$$u(t) = K_p \cdot e(t) + K_d \frac{de(t)}{dt} \quad (3)$$

La ecuación anterior es continua; sin embargo, un sistema digital nunca puede ser continuo, y por lo tanto nos vemos obligados a discretizar la ecuación. Para nuestro dron (y todos los drones que utilizan ArduCopter [3]), el vector de velocidad vertical se actualiza cada 200 ms. Por lo tanto, la ecuación del controlador puede ser discretizada considerando este período de muestreo (T_s). Además, cambiamos la denominación de la señal de salida genérica u por el nombre más apropiado v_{ver} , para aclarar que utilizamos un controlador PD para controlar la velocidad vertical.

$$v_{ver}[t] = K_p \cdot e[t] + K_d \frac{e[t] - e[t-1]}{T_s}, \quad T_s = 200ms \quad (4)$$

En este punto, hemos diseñado el controlador vertical que permite al VANT mantener la altitud. Sin embargo, también necesitamos que el VANT se mueva hacia sus objetivos, es decir, que siga la misión planificada. El objetivo actual (o waypoint) se da en coordenadas GPS. Para mover el VANT hacia el objetivo, basta con definir un vector de dirección $\vec{u}$ que parta del dron y termine en el objetivo. Tras la normalización, este vector puede escalarse de forma que el VANT se mueva a la velocidad prevista (o máxima) (en nuestro caso, $max.v_{hor}=15$ m/s, ya que es la configuración por defecto en el controlador

de vuelo). Así, tenemos la siguiente expresión para obtener el vector de velocidad horizontal $\vec{v}_{hor}$

$$\begin{aligned}\vec{u} &= \vec{target} - \vec{VANT} \\ \hat{u} &:= \frac{\vec{u}}{\|\vec{u}\|} \\ \vec{v}_{hor} &= max_v_{hor} \cdot \hat{u}\end{aligned}\quad (5)$$

Dados los parámetros adecuados, esto debería bastar para desplazar el VANT desde su posición actual hasta su objetivo, ajustando al mismo tiempo su altitud; el objetivo es que la altitud relativa con respecto al suelo se mantenga lo más constante posible. Sin embargo, durante nuestras pruebas simuladas, hemos observado dos problemas: en primer lugar, cuando el cambio de altura del terreno es significativo, el VANT necesita más tiempo para ascender/descender, ya que de lo contrario podría estrellarse. En la práctica, esto significa que, si la velocidad en el eje z se aproxima a su límite max_v_{ver} (5 m/s en nuestra propuesta), es necesario detener el movimiento hacia el objetivo para evitar una probable colisión. En segundo lugar, en la ecuación 1, denotamos la señal de error como la diferencia entre la altitud de referencia y la altitud real, en la misma instancia temporal. Durante el vuelo real, como mostraremos más adelante, se puede realizar una mejora mirando hacia delante, hacia donde estará el VANT, y tomar esa altitud como referencia. Estos pequeños ajustes se han implementado en nuestra propuesta, como se muestra en Figura 3.

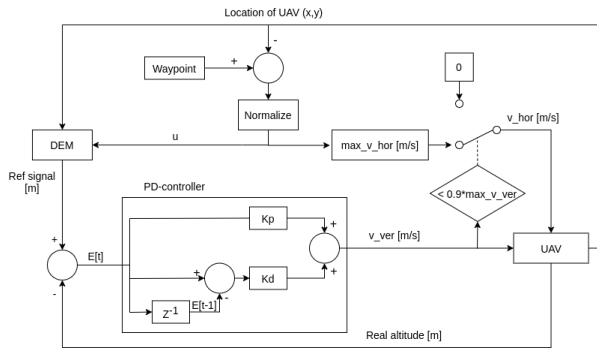


Fig. 3: Diagrama de bloques del método propuesto.

El controlador de vuelo del VANT (bloque de la derecha) nos proporcionará la altitud actual (absoluta) del dron. La diferencia entre la señal de referencia y la altitud actual denota la señal de error. Esta señal de error se introduce en el controlador PD, que a su vez calcula la velocidad vertical necesaria para minimizar la señal de error. La señal de velocidad vertical también controla si la velocidad horizontal se ajusta a 0 m/s o no. En caso de que la velocidad vertical sea alta (más del 90 % de la máxima permitida), el interruptor (a la derecha del diagrama de bloques) conmutará y pondrá la velocidad horizontal a cero. Esto dará al VANT el tiempo necesario para ajustarse a las diferencias de altitud antes de moverse hacia el waypoint de nuevo. Hay que tener en cuenta que se pretende continuar con la misión planificada lo más rápidamente posible, reduciendo la velocidad horizontal solo cuando sea estrictamente necesario.

El controlador de vuelo también nos proporciona la localización del VANT. Esta ubicación, junto con la ubicación del waypoint, se utiliza para calcular el vector de dirección $\vec{u}$. El vector de velocidad horizontal se obtiene escalando el vector de dirección $\vec{u}$ con la velocidad horizontal adecuada. Por último, el bloque MDE calcula la señal de referencia. Este cálculo se basa en el archivo MDE, que contiene el nivel de altitud absoluto para una ubicación determinada. Utilizamos la ubicación actual del VANT, así como su vector de dirección $\vec{u}$, para calcular su posición futura. La altitud sobre el nivel del suelo de esta posición futura se obtiene del archivo MDE, y la señal de referencia se calcula añadiendo el desplazamiento deseado sobre el nivel del suelo.

B. Parámetros

En nuestro modelo hay cinco parámetros: la ganancia proporcional K_p , la ganancia derivativa K_d , la distancia de anticipación [m], la velocidad horizontal máxima (max_v_{hor}) y la velocidad vertical máxima (max_v_{ver}). Por lo tanto, tenemos que encontrar los valores óptimos para estos parámetros, ya que los valores elegidos determinarán la rapidez con el que el VANT responde a los cambios, así como la estabilidad del sistema. Hemos decidido fijar los dos parámetros de velocidad máxima en 15 m/s para la velocidad horizontal, y 5 m/s para la velocidad vertical. Como ya se ha dicho, estos valores son un buen compromiso entre seguridad y velocidad, y se utilizan a menudo en aplicaciones de drones. En esta sección explicaremos cómo se han ajustado los valores de los otros tres parámetros. El objetivo de este ajuste es alcanzar nuestros objetivos específicos, que incluyen lograr un equilibrio adecuado entre el error vertical (incluido el rebasamiento), y el tiempo de misión. Es importante señalar que, dependiendo de las características del VANT, estos valores pueden variar. Por lo tanto, será necesario ajustar estos parámetros para otros tipos de VANT. Existen métodos de autoajuste, pero requieren la obtención de un modelo matemático del VANT, del que actualmente no disponemos. Por lo tanto, estimaremos los valores de nuestros parámetros basándonos en algunas reglas comúnmente conocidas. En esta sección, explicaremos el proceso que hemos seguido para que los resultados puedan reproducirse.

En primer lugar, hay que señalar que la determinación de los valores óptimos de los parámetros a partir de la simulación de una misión completa es poco práctica. Esto se debe a varias razones:

1. Simular una misión completa lleva mucho tiempo;
2. Es difícil determinar el verdadero efecto de un cambio en los parámetros;
3. Uno puede fijarse en resolver un problema específico (es decir, la misión), mientras que, en cambio, la solución debe ser aplicable a diferentes misiones (es decir, parámetros optimizados para una sola misión en lugar de una solución general);
4. Cuando se considera toda la misión, el movi-

miento horizontal del VANT también desempeña un papel;

Por lo tanto, obtenemos los valores de los parámetros utilizando una técnica clásica de la teoría de control, es decir, observando la respuesta al escalón del sistema. La respuesta al escalón es la respuesta (es decir, la salida) del sistema cuando se aplica un escalón (es decir, un cambio repentino en la señal de referencia). Dicha respuesta al escalón se representa en un gráfico (ver Figura 4), y es habitual normalizar la salida. Usando este gráfico, podemos determinar dos valores importantes: (i) el tiempo de subida, y (ii) el rebasamiento. El tiempo de subida nos explica lo rápido que el sistema convergerá a la nueva señal de referencia. El rebasamiento se refiere a la salida que supera la nueva señal de referencia deseada. En nuestra aplicación, ambas métricas son importantes. No queremos un sobreimpulso (o subimpulso) elevado, porque esto haría que el VANT volara demasiado alto (lo que podría infringir la legislación), o demasiado bajo (lo que podría provocar un accidente). Sin embargo, la respuesta del VANT también debe ser lo suficientemente rápida (es decir, un tiempo de subida corto) para aquellos casos en los que el perfil del terreno cambia constantemente, y el VANT debe ser capaz de seguir ese cambio.

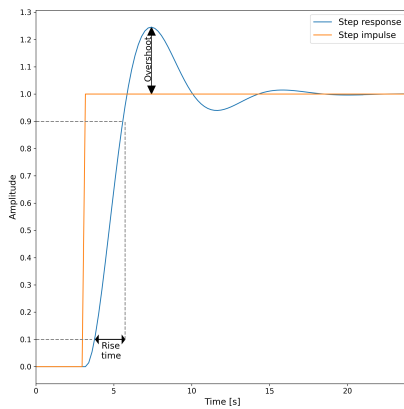
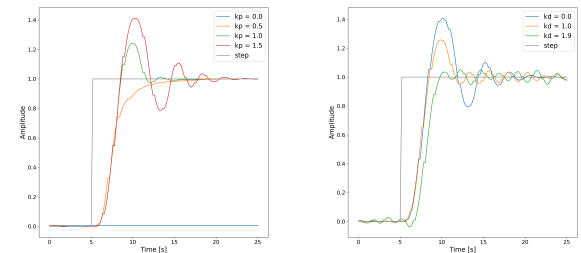


Fig. 4: Dos métricas importantes para la respuesta al escalón.

Ahora que hemos definido (aproximadamente) nuestra respuesta al escalón (es decir, bajo rebasamiento, pero también corto tiempo de subida), vamos a proceder variando los parámetros para observar cómo influyen en la respuesta al escalón. Comenzamos con el parámetro K_p , y fijamos K_d (Ecuación 4) a su mínimo (la distancia de anticipación no se utiliza en la respuesta al escalón, ya que estamos cambiando directamente la señal de referencia). En la 5a mostramos la respuesta al escalón para varios valores del parámetro K_p . En esta figura, podemos observar cómo el parámetro K_p influye en la respuesta al escalón. Empezamos poniendo K_p a cero. Obviamente, en este caso, el VANT no cambiará su altitud, y simplemente permanece volando según su señal de referencia original. A continuación, aumentamos el valor de K_p y K_d con pasos de 0,1 (no mostramos todas las posibilida-

des por legibilidad). Cuando aumentamos el valor de K_p , podemos ver que el tiempo de subida disminuye. Con un valor de 0,5 el VANT alcanza la nueva altitud sin sobrepasarla. Sin embargo, el tiempo de subida sigue siendo demasiado largo. Si aumentamos el valor de K_p , podemos reducir aún más el tiempo de subida. En la 5a parece que, en términos de tiempo de subida, no hay diferencia entre $K_p = 1,0$ y $K_p = 1,5$. Sin embargo, siempre es importante tener en cuenta tanto K_p como K_d . Como se ha indicado anteriormente, cambiamos ambos parámetros con un tamaño de paso de 0,1 (hasta 2,0) y, tras 400 experimentos, nuestros resultados muestran que con $K_p = 1,5$ y $K_d = 1,9$ se obtienen los mejores resultados. En la 5b mostramos la influencia del parámetro K_d . Como cabría esperar en base a la teoría de control, si aumentamos el valor del parámetro K_d , el rebasamiento se reducirá. Sin embargo, un valor demasiado alto dará lugar a un sistema inestable que tiende a oscilar. Como podemos ver, el sobreimpulso disminuye con un valor más alto de K_d . Aumentar el valor de K_d más allá de 1,9 (la mejor opción) solo introduce más oscilaciones no deseadas. Como se muestra en la Figura 6, para la respuesta del sistema con los parámetros óptimos (con respecto al sobreimpulso y el tiempo de misión), se observa alguna oscilación para nuestro enfoque de control. Sin embargo, como en nuestra aplicación el nivel del terreno cambia constantemente, esta oscilación no prevalecerá. Además, la influencia de la distancia de anticipación solo puede observarse claramente en el contexto de una misión real (períodos de tiempo más largos). Por lo tanto, estudiaremos su influencia en la siguiente sección.



(a) Influencia del parámetro K_p ($K_d=0$). (b) Influencia del parámetro K_d ($K_p = 1.5$).

Fig. 5: Análisis de la influencia de los parámetros K_p y K_d en la respuesta escalonada del VANT.

IV. EVALUACIÓN

Después de modelar nuestra solución propuesta y obtener los valores óptimos de los parámetros, en esta sección describimos las pruebas realizadas bajo diferentes condiciones. Para experimentar con seguridad, nos basamos en nuestro simulador/emulador realista VANT llamado ArduSim [16]. Experimentamos con dos escenarios diferentes: una zona rural con algunas colinas (Cumbre de Calicanto, Valencia, España), y una región montañosa (Mulhacén, Granada, España). Cada escenario se simuló varias veces, y con distintos valores para el parámetro distancia de anticipación.

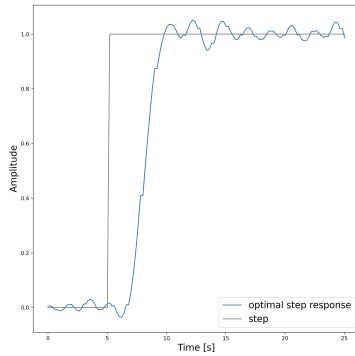


Fig. 6: Respuesta escalonada con parámetros óptimos $K_p = 1,5$; $K_d = 1,9$.

Durante nuestros experimentos, medimos la altitud del terreno y la altitud sobre el nivel del suelo de la VANT. Comenzamos presentando el simulador ArduSim, y a continuación repasamos los dos escenarios.

A. ArduSim

ArduSim es un simulador/emulador de vuelo multi-VANT de código abierto. El código está disponible en línea [17] bajo la Licencia Apache 2.0. El simulador está implementado en Java, y fue presentado formalmente en [16]. A continuación, destacamos brevemente las características clave de ArduSim.

- **Simulador y GCS:** ArduSim se puede utilizar en dos modos de funcionamiento. Puede funcionar como un simulador, y como una estación de control en tierra (GCS) para vehículos aéreos no tripulados reales. El protocolo (en este caso, el modelo que hemos creado) está desacoplado del modo operativo. Esto garantiza un rápido despliegue en VANT reales, ya que el protocolo no necesita ningún cambio una vez que funciona en el entorno simulado.
- **Pruebas automatizadas:** Como simulador, ArduSim puede ejecutarse con o sin interfaz gráfica de usuario (GUI). El modo con GUI se utiliza normalmente cuando se desarrolla el protocolo. Una vez que el protocolo está desarrollado y es necesario realizar muchos experimentos, se puede utilizar el modo sin GUI, y las pruebas se pueden ejecutar automáticamente.
- **Comunicación VANT-a-VANT:** ArduSim utiliza el estándar 802.11a para comunicarse, tanto entre los propios vehículos aéreos no tripulados, como entre los vehículos aéreos no tripulados y la estación de tierra. Cuando ArduSim se utiliza como simulador, la comunicación se realiza mediante enlaces virtuales. Cuando los protocolos se han probado a fondo, pueden desplegarse en VANT reales. En este caso, ArduSim enviará los mensajes a través de difusiones del Protocolo de Datagramas de Usuario (UDP).
- **Escalabilidad:** ArduSim fue diseñado para ser un simulador de vuelo multi-VANT. Por lo tanto, se puso mucho esfuerzo en la escalabilidad.

Esto se traduce en un simulador que es capaz de ejecutar hasta 256 VANTs en tiempo real en un PC de gama alta (Intel Core i7-7700, 32 GB de RAM).

- **API:** Dentro de la base de código de ArduSim, una interfaz de programación de aplicaciones (API), ofrece a los desarrolladores muchas funciones comunes (despegar, moverse a una ubicación GPS, aterrizar, etc.) para controlar el VANT.
- **Registro de datos:** ArduSim registra ampliamente los datos en varios formatos después de un vuelo, para que sea fácil de desarrollar y depurar.

Nuestra herramienta ArduSim actúa como simulador en tiempo real, y como elemento de software embarcado. Cuando se despliega en VANTs reales, se espera que se ejecute en un sistema integrado desplegado en el propio VANT (por ejemplo, Raspberry Pi), el cual se comunica con el controlador de vuelo mediante una conexión en serie. El protocolo de comunicaciones utilizado para dicha conexión es MAVLink [18], un protocolo de mensajería ligero para comunicarse con drones que sigue un patrón de diseño híbrido publicador-suscriptor punto a punto. De esta forma, ofrecemos un software universal que puede ser adoptado por otros usuarios sin necesidad de realizar ningún cambio en el propio firmware de Ardupilot.

B. Escenario A: Zona rural

Nuestro primer escenario (Cumbres Calicanto) está situado en una zona rural cercana a Valencia, España (ver Figura 7). Hemos elegido esta región por su relieve moderado (ver Figura 7). La cima de la colina tiene unos 210 metros de altura; por un lado, la pendiente de la colina es suave, en el centro hay algunos picos y valles abruptos, y en el otro lado de la colina la pendiente es mucho más pronunciada.



Fig. 7: Escenario A: Zona rural. Cumbres Calicanto vista aérea.

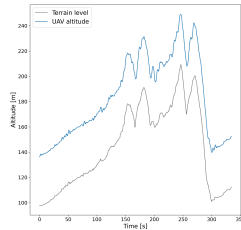
Como ya se ha mencionado, la influencia de la distancia de anticipación solo puede medirse durante una misión completa. En general, mirar más hacia adelante proporcionará al VANT más tiempo para anticiparse a los cambios de altitud. Sin embargo, mirar demasiado hacia adelante introducirá un comportamiento no deseado si el terreno tiene picos estrechos o depresiones. Durante nuestros experimentos, variamos el valor de la distancia de anticipación entre 2 y 20 metros. El fichero MDE que utilizamos tiene una resolución de 2 m (cuadrado de 2 veces $2 \times$ metros). Por lo tanto, aumentamos la distancia en pasos de dos metros cada vez. Durante la misión, medimos el error cada 200 ms. Tras el experimento, calculamos

el error medio y la desviación estándar (ver Tabla I). Basándonos en estas métricas, podemos determinar que, en este escenario, una distancia de 8 metros es la más efectiva para reducir el error.

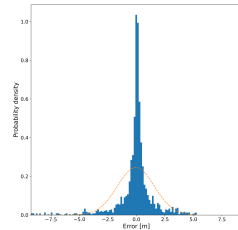
Tabla I: Influencia de la distancia de anticipación para el escenario A.

Distancia de anticipación [m]	μ [m]	σ [m]
2	0.1834	2.5176
4	-0.1464	2.3930
6	-0.0721	1.7832
8	-0.0493	1.6260
10	-0.0724	1.6399
12	0.0203	1.5128
14	0.0049	1.7969
16	0.0449	1.9540
18	0.1118	2.1984
20	0.1678	2.5435

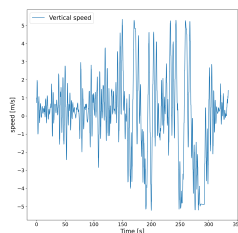
Los resultados de toda la misión se muestran en la Figura 8, mostrando que el VANT se adapta correctamente al perfil del terreno. Como se muestra en la 8b, el error se distribuye normalmente con $\mu = -0,0493$ y $\sigma = 1,6260$. Dicho de otro modo, el 99,7 % de las veces el error fue inferior a 5 metros.



(a) Altitud del VANT con respecto a la altitud del terreno.



(b) Distribución del error.



(c) Velocidad vertical m/s.

Fig. 8: Resultados para el escenario A.

Durante nuestro experimento, también tomamos varias métricas. Una de ellas fue la velocidad vertical del VANT, que mostramos en la 8c. Es interesante observar cómo cambia la velocidad vertical del VANT en función de la altitud del terreno (mostrada en la 8a). Como podemos ver, al principio (antes de 150 segundos) el nivel del terreno cambia lentamente, y la velocidad vertical del VANT está limitada entre -3 m/s, y 3 m/s. A partir de los 150 segundos, el nivel del terreno cambia mucho más rápido. Esto también se refleja en la velocidad vertical del VANT, que a menudo alcanza su máximo de ± 5 m/s. Además, comprobamos que el cambio de los niveles de altitud no

tiene un impacto significativo en términos de tiempo o energía adicionales. Para comprobarlo, medimos varias métricas del vuelo utilizando nuestro enfoque de ajuste de altitud, y también sin utilizar este mecanismo. Como se muestra en la Tabla II, la misión (representado en Figura 7) tiene una longitud de 3238 metros. Debido a la desviación, el VANT tuvo que ajustar su altitud en 583 metros en total. Este cambio de altitud causó un ligero retraso de la misión de alrededor del 3 %, y un aumento en el uso de energía del 3,49 %. Estos beneficios son muy pequeños, sobre todo si se tiene en cuenta que la alternativa (no cambiar de altitud) provocaría un accidente.

Tabla II: Vuelo rural: influencia del ajuste de la altitud sobre el tiempo de vuelo y la energía.

	Sin ajuste	Con ajuste	Diferencia
Dist. horizontal [m]	3238	3238	0 %
Dist. vertical [m]	0	583	-
Tiempo de vuelo [s]	324	334	+3.01 %
Gasto energético [kWh]	127	132	+3.49 %

C. Escenario B: zona de montaña

Tras estos prometedores resultados en una zona rural, probamos nuestro modelo en un entorno más desafiante, es decir, una región montañosa. Hemos optado por realizar esta simulación en la zona de Mulhacén, Granada, España (véase Figura 9). En estas montañas, los niveles del suelo cambian más rápido y con más frecuencia, lo que hace más difícil para el VANT mantener una altitud constante sobre el suelo.



Fig. 9: Escenario B: Zona de montaña. Mulhacén a vista de pájaro

Para esta región, también experimentamos con la distancia de anticipación. Como se muestra en la Tabla III, el margen de error es mayor debido a la mayor complejidad del escenario. En una región montañosa, la distancia de anticipación tiene que ser más

corta, como era de esperar. En este caso, podemos obtener los mejores resultados con una distancia de anticipación de 4 metros.

Tabla III: Influencia de la distancia de anticipación para el escenario B.

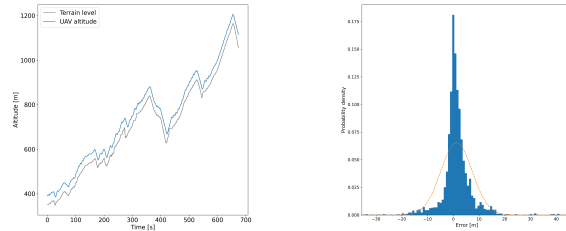
Distancia de anticipación [m]	μ [m]	σ [m]
0	-1.5206	8.6427
2	-0.1372	7.4118
4	1.2548	6.1141
6	2.9995	6.5989
8	4.4752	7.7420
10	5.9446	9.3839

Los resultados de este escenario se muestran en la Figura 10. De la 10a podemos observar que el terreno es más difícil, y con el uso de nuestro enfoque se puede evitar un accidente. Sin embargo, como también se muestra en la 10b, la distribución del error es más amplia que en el caso anterior. También para este vuelo hicimos un análisis de varias métricas. Como se muestra en la 10c, la velocidad vertical del VANT es a menudo máxima en un intento de mantener una altitud constante en este terreno rápidamente cambiante. Teniendo esto en cuenta, podríamos obtener mejores resultados (es decir, un error medio y una desviación estándar menores) si no restringimos la velocidad vertical máxima a 5 m/s. Sin embargo, no aumentamos esta velocidad máxima por varias razones. En primer lugar, por razones obvias de seguridad. Debemos recordar que hay un retraso de 200 ms antes de que podamos actualizar la velocidad, y luego hay un retraso adicional (debido a la inercia) antes de que el VANT alcance realmente esta nueva velocidad. Si aumentáramos la velocidad vertical máxima, lo más probable es que se produjera un comportamiento oscilatorio no deseado. Además, queremos que nuestro enfoque sea aplicable a muchos tipos de drones, y no todos tienen potencia suficiente para ascender a más de 5 m/s. Por último, aumentar la velocidad vertical máxima también supondría un mayor consumo de energía.

Tabla IV: Vuelo de montaña: influencia del ajuste de la altitud sobre el tiempo de vuelo y la energía.

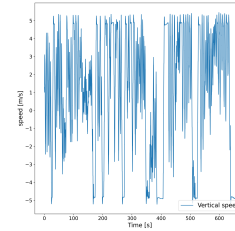
	sin ajuste	con ajuste	Diferencia
Horizontal distance [m]	6193	6193	0 %
Vertical distance [m]	0	2100	-
Flight time [s]	507	655	+28.99 %
Energy consumed [kWh]	200	261	+30.59 %

Como era de esperar, la influencia del ajuste de la altitud en el tiempo de vuelo y el consumo de energía es mucho mayor en este escenario exigente en



(a) Altitud del VANT con respecto a la altitud del terreno.

(b) La distribución del error.



(c) Velocidad vertical m/s.

Fig. 10: Results of scenario B.

comparación con el escenario anterior (rural). Como se muestra en la Tabla IV, la misión representada en la Figura 9 tiene una distancia horizontal de 6193 metros. La principal diferencia entre este escenario y el anterior es que, en el escenario actual, la distancia vertical recorrida es mucho mayor. Para regular la altitud sobre el suelo, el VANT tuvo que ascender y descender un total de 2100 metros a lo largo de la misión. Esto hizo que el tiempo de vuelo aumentara en un 29 % y, en consecuencia, el consumo de energía también se incrementó en un 30 %.

V. CONCLUSIONES Y TRABAJO FUTURO

La investigación en el campo de los vehículos aéreos no tripulados (VANT) ha madurado en los últimos años; como resultado, muchas industrias utilizan ahora VANT en sus actividades cotidianas. La mayoría de las veces, el VANT sigue siendo controlado manualmente por un piloto. Sin embargo, cada vez se desarrollan más aplicaciones (semi)autónomas. Para una aplicación autónoma, es importante que un VANT sea capaz de mantener una altitud constante con respecto al suelo, incluso cuando la altitud del terreno varía. De este modo se evita que el VANT se estrelle contra el suelo o supere las limitaciones legales de altitud.

En este trabajo, proporcionamos una solución que intenta mantener una altitud constante con respecto al suelo. Nuestro enfoque se basa en un controlador PD, que utiliza los datos de altitud procedentes de un archivo MDE como señal de referencia. Dado que los drones difieren en tamaño, peso y muchos otros factores, los valores de los parámetros del controlador PD probablemente difieran para cada VANT. Por ello explicamos detalladamente nuestro método de ajuste para que pueda reproducirse fácilmente.

Una vez ajustado el controlador PD, experimenta-

mos con dos escenarios diferentes: una zona rural y una zona montañosa. Los resultados del primer escenario (rural) muestran que el VANT es capaz de mantener una altitud relativa constante sobre el suelo. En ese experimento, el error (es decir, la diferencia entre la altitud deseada y la altitud real) fue inferior a 5 metros, el 99,7 % del tiempo, y el aumento del tiempo de vuelo y de la energía consumida debido a esos ajustes fue del 3,01 % y del 3,49 %, respectivamente, un rendimiento que consideramos suficiente para la mayoría de las aplicaciones. En el escenario más desafiante, es decir, la zona montañosa, el error aumentó ligeramente, y el aumento en el tiempo de vuelo y la energía consumida debido a los ajustes del perfil del terreno (2100 metros en total para la movilidad vertical) fueron ahora más significativos, alcanzando el 28,99 % y el 30,59 % de aumento, respectivamente.

Aunque los resultados son bastante buenos, es necesario seguir investigando para reducir el error en entornos difíciles. Para ello, en el futuro queremos ampliar nuestra solución investigando métodos de ajuste automático. Además, para evitar cambios rápidos en la altitud de vuelo, estudiaremos la posibilidad de suavizar las altitudes entre celdas adyacentes en el MDE. Por último, nos gustaría investigar un enfoque ligeramente diferente, en el que cambiamos la altitud de vuelo programada y dejamos que el controlador de vuelo ajuste la altitud.

AGRADECIMIENTOS

El presente trabajo ha sido realizado en el marco del proyecto PID2021-122580NB-I00, financiado por MCIN/AEI/10.13039/501100011033 y "Fondo Europeo de Desarrollo Regional (FEDER)".

REFERENCIAS

- [1] Ferran Giones and Alexander Brem, "From toys to tools: The co-evolution of technological and entrepreneurial developments in the drone industry," *Business Horizons*, vol. 60, no. 6, pp. 875–884, 2017.
- [2] Pixhawk, "Pixhawk: The hardware standard for open-source autopilots," <https://pixhawk.org/>, 2021, Accessed: 2021-10-26.
- [3] ArduPilot Dev Team, "ArduPilot: Copter documentation," <https://ardupilot.org/copter/>, 2021, Accessed: 2021-10-26.
- [4] EASA, "Easy access rules for unmanned aircraft systems," 2021, Accessed: 2021-10-26.
- [5] VC Technology Ltd, "Litchi," <https://flylitchi.com/>, 2022, Accessed: 2022-02-25.
- [6] Lim Tien Sze, Wen Guey Cheaw, Zahari Awang Ahmad, Choo Ai Ling, Koo Voon Chet, Habibah Lateh, and Luhur Bayuaji, "High resolution dem generation using small drone for interferometry sar," in *2015 International Conference on Space Science and Communication (IconSpace)*, 2015, pp. 366–369.
- [7] UM Rao Mogili and B B V L Deepak, "Review on application of drone systems in precision agriculture," *Procedia Computer Science*, vol. 133, pp. 502–509, 2018, International Conference on Robotics and Smart Manufacturing (RoSMa2018).
- [8] Jean-Philippe Aurambout, Konstantinos Gkoumas, and Biagio Ciuffo, "Last mile delivery by drones: an estimation of viable market potential and access to citizens across european cities," *European Transport Research Review*, vol. 11, no. 1, pp. 1–21, 2019.
- [9] Timothy Amukele, Paul M. Ness, Aaron A.R. Tobian, Joan Boyd, and Jeff Street, "Drone transportation of blood products," *Transfusion*, vol. 57, no. 3, pp. 582–588, 2017.
- [10] Donald E. Swihart, Arthur F. Barfield, Edward M. Griffin, Richard C. Lehmann, Shawn C. Whitcomb, Billie Flynn, Mark A. Skoog, and Kevin E. Processor, "Automatic ground collision avoidance system design, integration, flight test," *IEEE Aerospace and Electronic Systems Magazine*, vol. 26, no. 5, pp. 4–11, 2011.
- [11] Edward M Griffin, Russell M Turner, Shawn C Whitcomb, Donald E Swihart, James M Bier, Kerianne L Hobbs, and Amy C Burns, "Automatic ground collision avoidance system design for pre-block 40 f-16 configurations," Tech. Rep., AIR FORCE RESEARCH LAB WRIGHT-PATTERSON AFB OH MATERIALS AND MANUFACTURING ..., 2012.
- [12] Bob Eller, Peter Stanfill, Russell Turner, Shawn Whitcomb, Donald Swihart, Amy Burns, and Kerianne Hobbs, *Test and Evaluation of a Modified F-16 Analog Flight Control Computer*, p. 4726, American Institute of Aeronautics and Astronautics, 2013.
- [13] Jemie Muliadi and Benyamin Kusumoputro, "Neural network control system of uav altitude dynamics and its comparison with the pid control system," *Journal of Advanced Transportation*, vol. 2018, 2018.
- [14] Volodymyr Kharchenko and Nataliia Kuzmenko, "Unmanned aerial vehicle collision avoidance using digital elevation model," *Proceedings of the National Aviation University*, vol. 1, pp. 21–25, 2013.
- [15] Organismo Autónomo Centro Nacional de Información Geográfica, "Centro de descargas," <http://centrodedescargas.cnig.es/CentroDescargas/index.jsp>, 2021, Accessed: 2021-10-26.
- [16] Francisco Fabra, Carlos T. Calafate, Juan-Carlos Cano, and Pietro Manzoni, "ArduSim: Accurate and real-time multicopter simulation," *Simulation Modelling Practice and Theory*, vol. 87, no. 1, pp. 170–190, sep 2018.
- [17] GRC, "ArduSim: a novel real-time flight simulator," <https://github.com/GRCDEV/ArduSim>, 2021, Accessed: 2021-03-23.
- [18] Lorenz Meier and QGroundControl, "MAVLink Micro Air Vehicle Communication Protocol," <https://mavlink.io/en/>, 2007, Accessed: 2019-05-11.
- [19] CDEMA, "Digital Elevation Models," <https://www.cdema.org/virtuallibrary/index.php/charim-hbook/data-management-book/3-base-data-collection/3-2-digital-elevation-models>, 2021, Accessed: 2021-10-26.