

ISSN 1390-6712

MASKAY



Vol. 7, No. 1

Revista MASKAY

Departamento de Eléctrica y Electrónica
Universidad de las Fuerzas Armadas ESPE

DIRECTOR DEL DEEE

Evelio Granizo Montalvo

EDITOR GENERAL

Diego Arcos Avilés

CONSEJO EDITORIAL

Diego Benítez (USFQ)
Enrique V. Carrera (ESPE)
Lauro Ojeda (UMICH)
Ana Guamán (ESPE)
Juan Pablo Robelly (R&S)
Vinicius Petrucci (UFBA)
José Luis Rojo (URJC)
Carlos Julio Tierra (UFRJ)
Alexis Tinoco (ITA)

INFORMACIÓN DE CONTACTO

Revista MASKAY
Departamento de Eléctrica y Electrónica
Universidad de las Fuerzas Armadas ESPE

Av. Gral. Rumiñahui (S/N)
P. O. Box 17-15-243B
Sangolquí, Pichincha, Ecuador
Telf. +593 2 3989400 ext 1867

El contenido de los artículos aquí publicados es responsabilidad exclusiva de sus correspondientes autores. Mayor información en <http://maskay.espe.edu.ec/>.

Presentación

El Departamento de Eléctrica y Electrónica de la Universidad de las Fuerzas Armadas ESPE, consciente de su gran responsabilidad con la sociedad, busca proponer e impulsar iniciativas tendientes a fomentar el desarrollo integral de la ingeniería eléctrica y electrónica en el Ecuador. Como resultado de este compromiso, nace en el año 2011 la revista Maskay. Esta revista sintetiza de una forma documentada los esfuerzos en investigación y desarrollo que desarrollan los docentes/investigadores y estudiantes de la región.

Luego de seis ediciones y fruto del esfuerzo realizado por Dr. Vinicio Carrera, Dr. Diego Arcos e Ing. Evelio Granizo, docentes del Departamento de Eléctrica y Electrónica, en el mes de enero de 2017 la revista MASKAY es acogida e indexada en **Latindex**, base de datos que alberga revistas de información científica de los países de América Latina, el Caribe, España y Portugal. Posteriormente, debido a la calidad y presentación de los contenidos de la revista, MASKAY es premiada con la indexación en **SciELO-Ecuador**, convirtiéndola en la primera revista de la Universidad de las Fuerzas Armadas ESPE y una de las primeras del Ecuador en alcanzar este objetivo.

En la actualidad, la revista MASKAY se encuentra indexada en diversas bases de datos: **SciELO-Ecuador**, **Latindex**, **LatAm-Studies** Estudios Latinoamericanos, **RootIndexing**, **MIAR** (Matriz de Información para el análisis de Revistas) y **DOAJ** (Directory of Open Access Journals), con el objetivo de difundir a nivel internacional las publicaciones realizadas en esta revista. Además, para que exista una mejor divulgación de sus contenidos, la revista MASKAY cuenta con número **DOI (Digital Object Identifier)** legítimamente registrado y validado por **Crossref**, lo que permite una fácil localización de sus contenidos en la web.

En esta séptima edición, la revista MASKAY pone a consideración de la comunidad científica seis trabajos de investigación realizados por investigadores de centros de educación superior del país.

Diego Arcos Avilés
Editor General

Contenido

Evaluación estadística de generadores de secuencias pseudoaleatorias para aplicaciones de dispersión de energía DOI: 10.24133/maskay.v7i1.334 <i>Karla Rivas, Pablo Lupera, Christian Tipantuña</i>	1
Controlador con observador de estados de orden completo para un motor de DC mediante dSPACE DOI: 10.24133/maskay.v7i1.347 <i>René Palacios Ochoa</i>	6
On the performance improvement of the optimal-sampling-inspired self-triggered control at implementation stage DOI: 10.24133/maskay.v7i1.344 <i>Carlos Rosero, Juan Pablo Benavides, Cristina Vaca</i>	15
Implementación de Controladores por Realimentación de Estados y Controlador PID aplicado a un Motor de DC con Dos Masas mediante dSPACE DOI: 10.24133/maskay.v7i1.340 <i>René Palacios Ochoa</i>	22
Desarrollo de un algoritmo en MATLAB para la optimización de la resolución de una tarjeta USRP B210 para aplicaciones SDRadar DOI: 10.24133/maskay.v7i1.338 <i>David Moreno, Julio Mejía, Hugo Moreno</i>	31
Análisis e implementación de diferencial GPS en configuración simple y doble DOI: 10.24133/maskay.v7i1.343 <i>Mónica Zabala, Franklin López, Ángel Ortega</i>	41

Comité Técnico

Yolanda Angulo	Daniel Altamirano
Darwin Alulema	Diego Arcos Avilés
Paúl Ayala	Paúl Bernal
Enrique V. Carrera	Alejandro Chacón
Lourdes De La Cruz	Rodolfo Gordillo
Danilo Granda	Ana Guamán
Marco Gualsaquí	Román Lara
Oswaldo Martínez	Luis Orozco
Salomé Pérez	Víctor Proaño
David Rivas	Carlos Romero
Fabián Sáenz	Luis Tenezaca
Paricio Vizcaíno	Miroslava Zapata

Evaluación Estadística de Generadores de Secuencias Pseudoaleatorias para Aplicaciones de Dispersión de Energía

Statistical Evaluation of Pseudorandom Sequence Generators for Energy Dispersion Applications

Karla Rivas, Pablo Lupera Morillo, Christian Tipantuña

Abstract—A set of statistical tests is proposed, and with them, three pseudorandom number generators (PRNGs) and two pseudorandom bit sequence generators (PRBSs) are evaluated, which could be applied in the process of energy dispersion. The generators analyzed conclude that none exceed all the statistical tests proposed.

Index Terms— Energy dispersion, frequency selective fading, PRNG, PRBS, statistical tests.

Resumen—Se propone un conjunto de pruebas estadísticas para evaluar tres generadores de números pseudoaleatorios y dos generadores de secuencias binarias pseudoaleatorias, que se podrían aplicar en el proceso de dispersión de energía. Se concluye que ninguno de los generadores cumple todas las pruebas estadísticas propuestas.

Palabras Claves— Dispersión de energía, desvanecimiento selectivo en frecuencia, PRNG, PRBS, pruebas estadísticas.

I. INTRODUCCIÓN

LOS generadores de secuencias pseudoaleatorias (PRNG, siglas en inglés) se usan en los sistemas de comunicaciones con diversos objetivos, entre ellos: simulación de canales de ruido, encriptación, ecualización, compresión, aleatorización de bits, etc. Por eso, constantemente se investigan métodos de generación de secuencias con características más aleatorias y eficientes para su implementación. Para la evaluación de aleatoriedad de generadores de secuencias se utilizan pruebas estadísticas.

En el caso de que la investigación presentada requiera de una revisión exhaustiva del estado del arte.

Respecto al estudio del canal inalámbrico Sklar en su trabajo de investigación afirma que el comportamiento del canal produce desvanecimientos selectivos en frecuencia que

pueden ser contrarrestados con alguna técnica de dispersión de energía [1].

En algunos sistemas de comunicaciones inalámbricos la dispersión de energía se ejecuta multiplicando la señal digital de información con una secuencia pseudoaleatoria de bits, proceso que permite obtener una señal aleatoria. De esta manera en los sistemas de radio digital DRM y TV ISDBT-b se utilizan PRNGs de grado 9 y 15 respectivamente [2], [3].

En [4] se realiza la implementación de una forma caótica de aleatorizar una señal analógica de video en tiempo real. La señal de video se transmite por un canal inalámbrico y se reporta que se recupera la señal en el receptor con una claridad razonable.

En su trabajo Marinova y Tchobanova realizaron la evaluación estadística de secuencias con pruebas de monobit, runtest y espectral de circuitos generadores de números pseudoaleatorios utilizando secuencias de 1000 bits en base a la norma 800-22 de la NITS (National Institute of Standards and Technology). En este trabajo se evaluaron generadores de secuencias binarias pseudoaleatorias (PRBS, siglas en inglés) de grado 9, 16 y 21, y se implementaron sobre FPGA [5].

En el estudio realizado por Goretti, Campo y Echanobe se utiliza para la evaluación estadística la norma FIPS-140 y 141 del NITS considerando varios generadores, entre ellos los generadores congruenciales lineales (LCG) de Fibonacci, Green y Mitchell Moore [6].

En otro estudio realizado por Katti, Kavasseri y Sai se propone un generador lineal congruencial comparativo (CLCG) que se evalúa con la normativa 800-22. El CLCG está conformado por dos LCG que determinan su salida. Según el estudio este tipo de generador supera todas las pruebas de la norma 800-22, pero requiere dos LCG como fuente de aleatoriedad [7].

Considerando que la generación de secuencias pseudoaleatorias se aplica en los sistemas de transmisión inalámbrica para ejecutar la dispersión de energía, surge la necesidad de establecer una forma estadística para evaluar dichos generadores de secuencias pseudoaleatorias con la finalidad de establecer cuál de los generadores presenta mejores características para la aplicación de técnicas de dispersión de energía.

Karla Rivas, Estudiante Escuela Politécnica Nacional, Departamento de Electrónica, Telecomunicaciones y Redes de Información, (e-mail: krivasp1504@hotmail.com).

Pablo Lupera Morillo, Escuela Politécnica Nacional, Departamento de Electrónica, Telecomunicaciones y Redes de Información, (e-mail: pablo.lupera@epn.edu.ec).

Christian Tipantuña, Escuela Politécnica Nacional, Departamento de Electrónica, Telecomunicaciones y Redes de Información, (e-mail: christian.tipantuna@epn.edu.ec).

El presente artículo se halla estructurado de la siguiente manera. En la sección II se describe el comportamiento del canal inalámbrico y la aplicación de la dispersión de energía como técnica para mitigar el efecto del desvanecimiento selectivo en frecuencia. En la sección III se proponen las pruebas estadísticas para evaluar los generadores de secuencias pseudoaleatorias. En la sección IV se muestran los resultados de la evaluación de los tres generadores LCG (Fibonacci, Green y Mitchell-Moore) y dos PRBS (grado 9 y 15) con LFSR (registro de desplazamiento con realimentación lineal). Para la evaluación estadística se aplican seis pruebas seleccionadas de la norma 800-22. En la sección V se describen las conclusiones del trabajo desarrollado y se proponen los trabajos futuros a realizar.

II. COMPORTAMIENTO DEL CANAL INALÁMBRICO Y LA APLICACIÓN DE LA DISPERSIÓN DE ENERGÍA

El comportamiento del canal inalámbrico es totalmente impredecible por los fenómenos de reflexión, refracción, difracción y dispersión de las señales que pueden producirse en la propagación de las ondas electromagnéticas. Estos fenómenos producen réplicas de las señales que se manifiestan con cambios de la amplitud, fase, frecuencia y ángulo de llegada de la señal al receptor, produciendo desvanecimientos selectivos en frecuencia de la señal. El desvanecimiento de Rayleigh es muy utilizado como modelo del canal inalámbrico [8].

A. Desvanecimiento selectivo en frecuencia

Se pueden identificar dos tipos de desvanecimientos: desvanecimiento a gran escala y desvanecimiento a pequeña escala. El desvanecimiento a gran escala produce la atenuación de la señal como consecuencia de la propagación de la onda en áreas extensas y de la interacción con cuerpos grandes en relación a la longitud de onda de la señal, tales como bosques, edificios, etc. Por otro lado, el desvanecimiento a pequeña escala se produce por el movimiento de las estaciones enlazadas y por los obstáculos en el canal, cuyas dimensiones son comparables con la longitud de onda de la señal, y que afectan a su amplitud y fase.

Una de las manifestaciones del desvanecimiento a pequeña escala se produce cuando el tiempo que tarda la componente de la multitrayectoria en llegar al receptor, T_m , es mayor en relación al tiempo de duración del símbolo, T_s , es decir bajo la condición de $T_m > T_s$ [1]. En el dominio del tiempo, el desvanecimiento selectivo en frecuencia se manifiesta principalmente como distorsión por la interferencia entre símbolos ISI. Las componentes que se reciben son derivaciones de una señal principal, por lo que guardan un alto grado de similitud. Sin embargo, el desvanecimiento se puede analizar considerando la potencia de cada componente de frecuencia de señal original de manera independiente [1].

Se establece entonces una banda de coherencia, que es el rango de frecuencias donde el canal afecta por igual a las componentes. Fuera de la banda de coherencia, las atenuaciones de las componentes espectrales de la señal sufren

degradaciones independientes y diferentes a las que están dentro del mismo. Los efectos de este tipo de desvanecimiento son “independientes del rango de frecuencias de la señal y ocurren todo el tiempo en el canal”

B. Dispersión de energía para mitigar el desvanecimiento selectivo en frecuencia

Para mitigar los efectos del desvanecimiento selectivo en frecuencia se utilizan varias técnicas como la ecualización, el ensanchamiento espectral, la multiplexación por división de frecuencia ortogonal (OFDM), y la dispersión de energía [1]. En este proyecto se estudió la dispersión de energía mediante la aleatorización de la señal.

Debido a la naturaleza de la información, las secuencias binarias que son transmitidas en los sistemas de comunicaciones pueden contener patrones repetitivos, produciendo consecuentemente una concentración de potencia en ciertas regiones del espectro, tal como se observa en la Fig. 1. Por lo tanto, en el dominio de la frecuencia, se puede observar que la energía de la señal se concentra en unas frecuencias más que en otras. A diferencia de estas secuencias de información, las secuencias de bits aleatorias tienen una distribución equitativa de la potencia en el espectro de frecuencias [9]. La dispersión de energía consiste en “aleatorizar” una señal binaria mediante la suma lógica de la misma con una secuencia aleatoria [10].

Mediante la aleatorización se distribuye la potencia de la señal en todo el espectro, reduciendo el efecto del desvanecimiento selectivo en frecuencia [9] [10]. En la Fig. 1 se muestra el efecto de la aplicación de la dispersión de energía, al combinar una secuencia binaria de audio MP3 con un generador PRBS de grado 15. En la Fig. 1 se visualiza como disminuyen ciertos picos de potencia de la señal después de la dispersión de energía, lo que reduciría el efecto de los desvanecimientos selectivos en frecuencia.

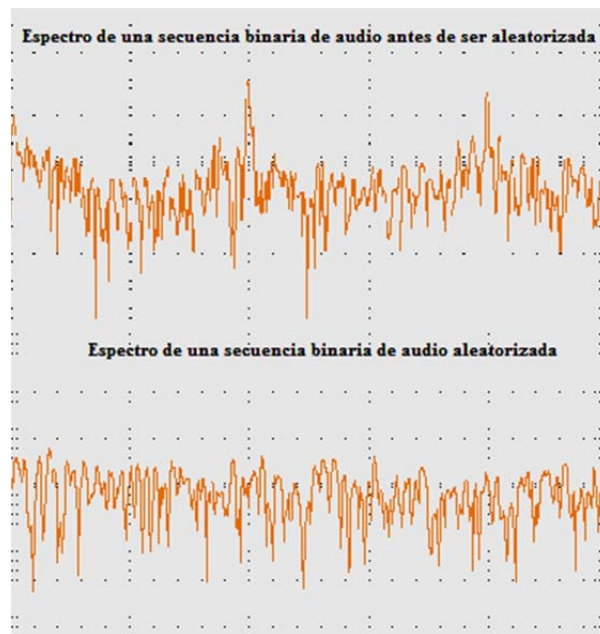


Fig. 1. Espectro de frecuencia de la secuencia binaria de audio antes y después de ser aleatorizada.

III. PROPUESTA DE PRUEBAS ESTADÍSTICAS

La evaluación del grado de aleatoriedad de los generadores se realiza en base a conjuntos de pruebas estadísticas. De los trabajos previos estudiados se establece que los conjuntos de pruebas más populares son las normas: 800-22, FIPS-140 y FIPS-141. En la evaluación de un generador se puede utilizar una parte o la totalidad de las pruebas propuestas de acuerdo a la profundidad de análisis que se requiera realizar. En [11] se enuncia que de acuerdo a la aplicación se deben establecer ciertas condiciones particulares para las pruebas estadísticas, como: longitud de la secuencia, condición mínima para superar la prueba, entre otros.

Para la evaluación de los generadores de secuencias pseudoaleatorias en aplicaciones de dispersión de energía se ha escogido un conjunto de pruebas estadísticas para evaluar el nivel de aleatoriedad de 6 generadores. El conjunto de pruebas se seleccionó de la norma 800-22 del NIST [11] y se muestran en la Tabla I con la propiedad estadística que permiten comprobar.

TABLA I
CONJUNTO DE PRUEBAS ESTADÍSTICAS

#	Prueba	Propiedad que se comprobará
1	Frecuencia Monobit	Uniformidad de valores y distribución uniforme en secuencia global
2	Frecuencias en bloques de M-bits	Uniformidad de valores y distribución uniforme en subsecuencias
3	Ráfagas (Runttest)	Aleatoriedad Global en secuencia global
4	Ráfagas largas (Longest Runttest)	Aleatoriedad Global en subsecuencias
5	Rango de matriz binaria	Independencia lineal de valores en secuencia global
6	Transformada de Fourier	Repetición de patrones y autocorrelación ¹ .

Para evaluar la aleatoriedad, en MatlabTM se programaron los generadores y se ejecutaron las pruebas estadísticas con los parámetros que se muestran en la Tabla II.

Para evaluar si un generador supera la prueba se utilizó el parámetro estadístico del p-valor. Como resultado de cada prueba realizada sobre las muestras analizadas se obtiene un conjunto de p-valores. Para interpretar los resultados considerando el conjunto de p-valores se utilizan dos métodos: el del intervalo de confianza y el del histograma.

En el primer método de interpretación de los resultados se analiza si el conjunto de p-valores se encuentra dentro del intervalo de confianza, para ello se utiliza la siguiente ecuación [11].

$$\text{intervalo confianza} = \hat{p} \pm \sqrt{\frac{\hat{p} \cdot (1 - \hat{p})}{m}} \quad (1)$$

En donde $p = 1 - \alpha$ y m representan el número de secuencias por muestra que se van a probar. En el caso de una muestra

¹ En el contexto, significa que se compara la señal con sí misma para identificar los patrones repetidos.

con 1000 secuencias el límite inferior del intervalo de confianza según la ecuación anterior sería 0.9806. Este primer método se utiliza para establecer los generadores con mejores características de aleatoriedad.

En el segundo método de interpretación se evaluaron los histogramas obtenidos de los p-valores en las pruebas con los generadores. Este segundo método de interpretación complementa al primero para establecer el generador con mejores características de aleatoriedad.

TABLA II
PARÁMETROS PARA LAS PRUEBAS ESTADÍSTICAS REALIZADAS SOBRE LAS SECUENCIAS

Parámetro	Valor
Bits por muestra	Para cumplir con el requisito de la prueba de Rango de Matriz Binaria se generan secuencias de 5000 bits por cada muestra en los modelos.
Secuencias por muestra	Mínimo 1000 para ambos métodos de interpretación.
Cantidad de muestras probadas	10 muestras por cada generador.
Valor crítico α	0,01 para la batería de pruebas y métodos de interpretación.

IV. RESULTADOS DE EVALUACIÓN DE LOS GENERADORES

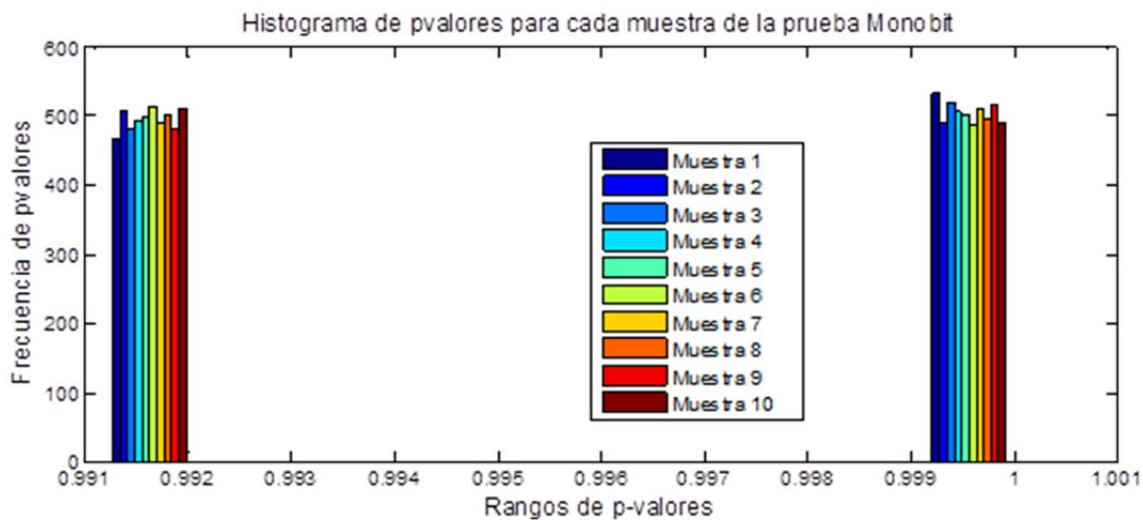
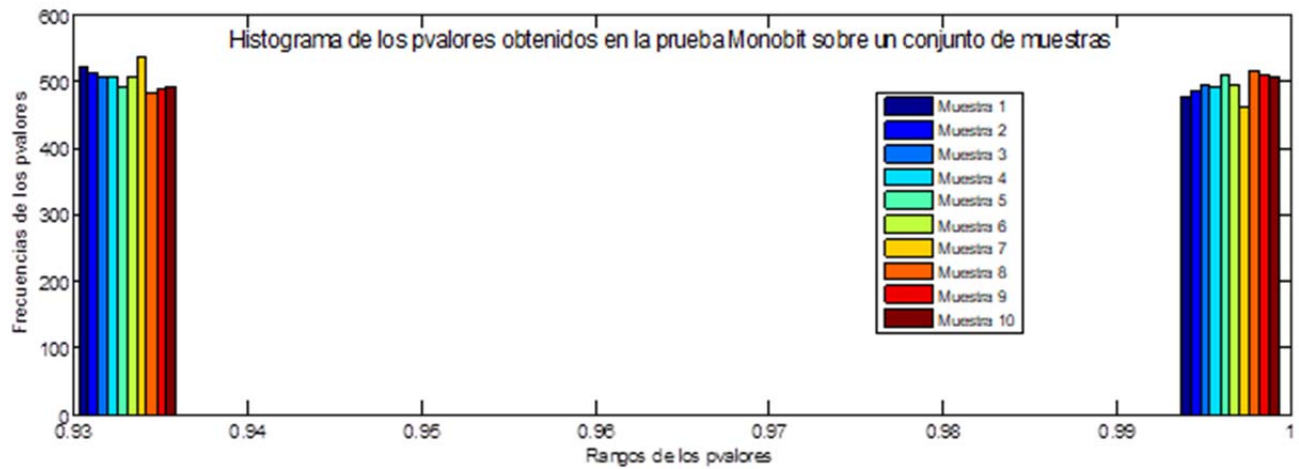
Mediante el método de interpretación del intervalo de confianza, se obtuvieron los resultados de la Tabla III. En el análisis se considera que se supera la prueba si el valor estadístico es mayor a 0,9806. Analizando los resultados, se tiene que los PRNG de Fibonacci, Green y Mitchell-Moore superaron solo la prueba de frecuencia en bloques de M-bits. Por otro lado, el PRBS de grado 9 supera las tres primeras pruebas, siendo el modelo que más pruebas supera, seguido por el PRBS de grado 15 que supera la primera y tercera prueba.

TABLA III
RESULTADOS DE LAS PRUEBAS ESTADÍSTICAS

# de Prueba	1	2	3	4	5	6	
Generador de secuencias pseudoaleatorias	Fibonacci	0.955	0.990	0.959	0	0	0.070
	Green	0.962	0.992	0.965	0	0	0
	Mitchell Moore	0.961	0.992	0.959	0	0	0
	PRBS grado 9	1	1	1	0	0.77	0
	PRBS grado 15	0.993	0.843	0.998	0	0.40	0

El segundo método de interpretación se utiliza para evaluar la característica de aleatoriedad de los PRBS de grado 9 y grado 15. Este método se divide en dos partes, en la primera se evalúan los generadores en todo el período de los mismos, y la segunda es la evaluación de la secuencia generada con la "semilla" que establecen las normas de radiodifusión.

De los resultados de las pruebas estadísticas sobre el período completo de los PRBS de grado 9 y 15, se observan los mismos resultados para los dos generadores, obteniéndose



que los dos superan las tres primeras pruebas con p-valores de 1 y que las tres últimas pruebas no son superadas con p-valores de 0. Por los resultados obtenidos se analizan los histogramas de los p-valores.

En las Fig. 2 y 3 se muestran los histogramas de los p-valores obtenidos de la prueba monobit para el PRBS de grado 9 y 15 respectivamente. La prueba monobit se toma como referencia para comparar el comportamiento de los dos generadores analizados. De los histogramas se observa que el PRBS de grado 15 concentra los p-valores en un rango más pequeño y más cercano a 1 que el PRBS de grado 9, interpretándose que el primero muestra mejores características de aleatoriedad.

Los resultados de la evaluación de la influencia de la “semilla” sobre los PRBS de grado 9 y 15 se presentan en la Tabla IV. En este caso se observan mejores efectos de aleatoriedad en el modelo del PRBS de grado 15 con la semilla de la norma ISDB-Tb. La quinta prueba de matriz de rango binario no se realizó sobre el PRBS de grado 9 debido a los requerimientos de longitud de secuencia

TABLA IV
EVALUACIÓN DE LOS PRBS 9 Y 15 CON LA SEMILLA DE LAS NORMAS DE
RADIODIFUSIÓN DRM E ISDB-TB RESPECTIVAMENTE

# de Prueba	1	2	3	4	5	6
PRBS grado 9	0.929	0.280	0.929	0	NA*	0
PRBS grado 15	0.991	0.351	0.991	0	0	0

*NA - no aplica

V. CONCLUSIONES

Para la evaluación estadística de secuencias pseudoaleatorias en dispersores de energía se propone la aplicación de 6 pruebas planteadas en la norma 800-22. Para los parámetros establecidos en las pruebas estadísticas se observó que los PRBS de grado 9 y 15 tienen mejores características de aleatoriedad que el resto de generadores propuestos, porque superaron la mayor cantidad de pruebas estadísticas planteadas. Sin embargo, estos generadores no poseen las siguientes propiedades: distribución uniforme de unos y ceros en subsecuencias (Prueba de ráfagas en

subsecuencias), independencia lineal entre subsecuencias (Rango de matriz binaria), y distribución aleatoria entre los valores de las potencias en el espectro de frecuencia (Prueba espectral). Este incumplimiento de ciertas propiedades se debe principalmente a que los generadores mantienen patrones repetitivos, ya que utilizan una fórmula de recurrencia. En las condiciones establecidas en el presente análisis se comprobó que la utilización de las semillas de las normas DRM e ISDB-Tb no mejora los resultados de las pruebas estadísticas.

Resta por ejecutar un estudio del efecto del dispersor de energía en un canal inalámbrico mediante simulaciones y pruebas en ambiente real para comprobar el efecto del incumplimiento de ciertas propiedades de aleatoriedad.

REFERENCIAS

- [1] B. Sklar, "Rayleigh Fading Channels in Mobile Digital Communication Systems. Part I: Characterization.", *IEEE Communication Magazine*, vol. 1, n° 0163-6804, pp. 90-100, 1997.
- [2] ETSI, "Digital Radio Mondiale (DRM). System Specification" Enero 2014. [En línea]. Disponible: <http://www.drm.org/wp-content/uploads/2014/01/DRM-System-Specification-ETSI-ES-201-980-V4.1.1-2014-01.pdf>. [Último acceso: 21 Octubre 2014].
- [3] Asociación Brasileña de Normas Técnicas, "NBR 15601", 11 Noviembre 2007. [En línea]. Disponible: http://www.upjet.org.ar/archivos_noticias/356-1.pdf. [Último acceso: 21 Octubre 2014].
- [4] Ned, Corron; Billy Reed; Blakely Jonathan; Krishna Myneni; Shawn Pethel; "Chaotic scrambling for Wireless analog video"; *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, Issue 9, pp. 2504-2513, 2010.
- [5] Galia, Marinova; Zdravka, Tchobanova; "Simulation, Measurement and Test Environment for Pseudo Random Number Generator Circuits", *Research on Electric and Electronic Measurement for the Economic Upturn*, vol. 20, pp. 15-17, 2014.
- [6] Goretti, Campo, Echanobe; "Circuitos digitales basados en FPGA para generación de números aleatorios", [En línea]. Disponible: http://gtts.ehu.es/dEyE/Actualizable/Anual/Curso05-06/VI_Jornadas_IE/trabajos_dirigidos/Goreti_Sevillano.pdf [Último acceso: 27 Octubre 2014].
- [7] Raj, Katti; Rajesh, Kavasseri; Vyasa, Sai, "Pseudorandom bit generation using coupled congruential generators", *Transactions on Circuits and Systems II*, vol. 57, n° 3, pp. 203-207, 2010.
- [8] Xinjia, Chen; Guoxiang, Gu; Kemin, Zhou; "Measurement complexity of Rayleigh Fading Channels", *IEEE Transactions on Vehicular Technology*, vol. 58, issue: 7, pp. 3776-3781, 2009.
- [9] S. Morris y A. Smith-Chaigneau, *Interactive TV Standars. A guide to MHP, OCAP and JavaTV.*, Burlington, USA: Focal Press, 2013, p. 28.
- [10] H.-J. Zepernick y A. Finger, *Pseudo random signal processing. Theory and applications.*, Primera ed., Chischester, Inglaterra: John Wiley & Sons, Ltd., 2005.
- [11] A. Rukhin, J. Soto, J. Nechvatal, M. Smid y E. Barker, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications.", Abril 2010. [En línea]. Disponible: <http://csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf>. [Último acceso: 9 Octubre 2014].

Controlador con Observador de Estados de Orden Completo para un motor de DC mediante dSPACE

Controller with Full Order State Observer for a DC Motor utilizing dSPACE

Alexander Palacios Ochoa

Abstract— This paper presents the design, simulation, and implementation of two controllers with state observers for a DC motor with two masses. The first controller is a state feedback controller, and the second is a proportional integral controller that includes state feedback. The design and simulation are realized in Matlab, and the implementation is done using the controller module dSPACE DS1104. The dSPACE module is programmed through Simulink blocks using the C code generator Real-Time Workshop. The electromechanical system has five state variables, from which the motor angular position is controlled using full-order state observers. The performance of the two controllers implemented considering over-impulse, steady-state error, and establishment time of the unit step response is compared. The integral proportional controller has the advantage of having a more minor steady-state error, while the controller without integral proportional control has a smaller establishment time.

Index Terms—State variable feedback controller SVF, proportional - integral controller PI, state observer, DC motor.

Resumen— En este artículo se presenta el diseño, simulación e implementación de dos controladores con observadores de estado para un motor de DC con dos masas. El primer controlador es un controlador con realimentación de estados y el segundo es un controlador proporcional integral con realimentación de estados. El diseño y la simulación se la realiza en Matlab y la implementación se la realiza utilizando el módulo controlador dSPACE DS1104. El módulo dSPACE se programa mediante los bloques de Simulink utilizando el generador de código C, Real-Time Workshop de Matlab. El sistema electromecánico tiene cinco variables de estado, de las cuales se controla la posición angular del motor utilizando observadores de estado de orden completo. Se realiza una comparación del desempeño de los dos controladores implementados considerando sobre impulso, error en estado estable y tiempo de establecimiento de la respuesta a la función escalón. El controlador proporcional integral tiene la ventaja de tener menor error en estado estable, mientras que el controlador sin control proporcional integral tiene un tiempo de establecimiento menor.

Palabras Claves— Controlador con realimentación de estados SVF, control proporcional integral PI, observador de estados, motor de DC.

I. INTRODUCCIÓN

LOS observadores de estado se utilizan cuando no es posible medir los estados de las variables del sistema, por ejemplo porque no se dispone de los sensores, porque es de difícil acceso o por el costo alto del sensor. Como solución se realiza un modelo matemático que se comporta de manera similar al sistema físico, en este modelo matemático se dispone de las medidas de todas las variables del sistema. Sin embargo las variables reales no son exactamente iguales al sistema modelado por lo que se requiere un factor de corrección de estas variables, antes de ser utilizadas por el controlador de realimentación de estados. El modelo de la planta del motor debe ser lo más cercano posible al sistema real para tener un mejor desempeño del controlador y del observador de estados.

Existen varios trabajos relacionados al control de un motor de DC con variables de estado y observadores, por ejemplo: en [1] presenta un observador de estados con control de velocidad de un motor de DC implementado en dSPACE, en [2] presenta el diseño de un estimador de estados para una máquina de DC, en [3] presenta la estimación de la posición del rotor de un motor de DC mediante observadores de estado, en [4] presenta un controlador para un motor de DC implementado mediante dSPACE y Simulink.

En este trabajo se implementa dos controladores, el primero es un controlador con variables de estado simple sin realimentación de la salida, es decir sin comparar la salida actual con la referencia del sistema, no incluye control proporcional, ni integral, simplemente se tiene realimentación de los estados del sistema; y el segundo controlador dispone de un control proporcional integral, con lo cual se realiza una comparación de la salida actual del sistema con la referencia. Al error de esta comparación se le aplica el control PI y esta señal se convierte en la referencia a la cual se le resta la realimentación de estados del sistema [5].

Para cada uno de los dos controladores SVF y SVF-PI se incluye observadores de estado de orden completo. La planta dispone de cinco variables de estado: posición angular en el lado del motor, posición angular en el lado de la carga, velocidad angular en el lado del motor, velocidad angular en el

lado de la carga y torque en el motor. Las variables de estado del sistema se presentan en la Tabla I.

TABLA I
NOMENCLATURA DE LAS VARIABLES DE ESTADO [6]

Abreviatura	Significado
φ_M	Posición angular en el lado del motor.
φ_L	Posición angular en el lado de la carga.
ω_M	Velocidad angular en el lado del motor
ω_L	Velocidad angular en el lado de la carga
M_M	Torque actual del motor

El sistema electromecánico consiste en un motor de DC con dos masas, la primera masa está acoplada al eje del motor y la segunda masa está conectada a la primera mediante una banda, la cual transfiere el torque del motor. Los datos de posición y velocidad medidos en la primera masa se denominan lado del motor y la posición y velocidad angular medidas en la segunda masa se denominan lado de la carga.

La simulación de los controladores se la realiza en Simulink y mediante el generador de código C, Real Time Workshop de Matlab se programa el módulo dSPACE ds1104. El objetivo de este trabajo es comparar el desempeño del controlador SVF con observador de estados y el controlador SVF-PI con observador de estados en cuanto a sobre impulso, error en estado estable y tiempo de establecimiento al aplicar una función escalón de referencia. La variable a controlar es la posición angular en el lado del motor.

Los controladores son simulados en Matlab y luego cargados en un módulo controlador dSPACE. A través de la interfaz en tiempo real RTI de Matlab se puede configurar las entradas y salidas del controlador gráficamente en Simulink. Los controladores dSPACE principalmente se utilizan para realizar pruebas en prototipos.

Para diseñar los controladores se ha realizado previamente la identificación del sistema, el cual contiene la parte mecánica y eléctrica. El sistema mecánico consta del eje del motor con dos masas, una en el lado del motor y la otra en el lado de la carga; las dos masas están conectadas entre sí mediante una banda que transfiere el torque del motor desde la una masa a la otra. Se identificó las inercias de cada masa, el coeficiente de amortiguamiento viscoso y la elasticidad de la banda. El sistema eléctrico corresponde al motor y al controlador del motor. Una vez realizada la identificación de la planta se la representa en variables de estado, y sobre este modelo matemático se basan los diseños de los controladores.

II. MODELO DEL MOTOR DE DC CON DOS MASAS

El sistema electromecánico es modelado en dos partes: 1) La parte eléctrica que contiene el driver del motor y el motor de corriente continua y 2) La parte mecánica correspondiente a las dos masas y a la banda del motor

El voltaje en los terminales de armadura del motor está dado por (1) [7].

$$V_T = i_a \cdot R_a + L_a \cdot \frac{d}{dt} i_a + E_b \quad (1)$$

Donde V_T es el voltaje en terminales de armadura, i_a es la

corriente de armadura, E_b es el voltaje inducido por la fuerza contra-electromotriz y L_a es la inductancia de la armadura.

Utilizando la herramienta de identificación del sistema de Matlab se encuentra la función de transferencia que incluye el motor y el driver del motor dada por:

$$G(s) = \frac{M_M}{M_{ref}} = \frac{k_E}{T_E \cdot s + 1} \cdot e^{-T_d \cdot s} \quad (2)$$

Para la parte mecánica del sistema se presenta de (3) a (7) [6]. En la Tabla II se presenta la nomenclatura utilizada para describir los parámetros de estas expresiones.

$$\ddot{\omega}_M = -\frac{r_M^2 \cdot c_B}{J_M} \cdot \varphi_M - \frac{d_M}{J_M} \cdot \omega_M + \frac{i \cdot r_M^2 \cdot c_B}{J_M} \cdot \varphi_L + \frac{M_M}{J_M} \quad (3)$$

$$\ddot{\omega}_L = \frac{r_L^2 \cdot c_B}{J_L \cdot i} \cdot \varphi_M - \frac{r_L^2 \cdot c_B}{J_L} \cdot \varphi_L - \frac{d_L}{J_L} \cdot \omega_L - \frac{M_L}{J_L} \quad (4)$$

$$\dot{M}_M = \frac{M_{ref}}{T_E} - \frac{M_M}{T_E} \quad (5)$$

$$\dot{\varphi}_M = \omega_M \quad (6)$$

$$\dot{\varphi}_L = \omega_L \quad (7)$$

Las entradas son: M_L torque en la carga y M_{ref} torque de referencia de la planta.

TABLA II
PARÁMETROS DEL SISTEMA [6]

Abreviatura	Significado
c_B	Coficiente de elasticidad de la banda
d_M	Coficiente de amortiguamiento en el lado del motor
d_L	Coficiente de amortiguamiento en el lado de la carga
i	Relación de radios de las dos masas
J_M	Momento de inercia en el lado del motor
J_L	Momento de inercia en el lado de la carga
T_E	Constante eléctrica de tiempo
r_M	Radio de la masa en el lado del motor
r_L	Radio de la masa en el lado de la carga

La ecuación de estado del sistema se presenta en (8) y la salida del sistema está dada por (9) como sigue:

$$\dot{X}(t) = A \cdot X(t) + B \cdot u(t) \quad (8)$$

$$Y(t) = C \cdot X(t) + D \cdot u(t) \quad (9)$$

En la expresión de salida de este sistema no existe un camino directo de la variable de entrada $u(t)$ hacia la salida, por lo que $D = 0$. Las matrices A, B y C [6] son:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\frac{r_M^2 \cdot c_B}{J_M} & -\frac{d_M}{J_M} & \frac{r_M^2 \cdot c_B \cdot i}{J_M} & 0 & \frac{1}{J_M} \\ 0 & 0 & 0 & 1 & 0 \\ \frac{r_L^2 \cdot c_B}{J_M \cdot i} & 0 & -\frac{r_L^2 \cdot c_B}{J_L} & -\frac{d_L}{J_L} & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{T_E} \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -\frac{1}{J_L} \\ \frac{1}{T_E} & 0 \end{bmatrix}, \quad C = [1 \ 0 \ 0 \ 0 \ 0].$$

La entrada del sistema $u(t)$ es: $u(t) = \begin{bmatrix} M_{ref} \\ M_L \end{bmatrix}$.

Las variables de estado del sistema electromecánico descritas en la tabla I son agrupadas en el vector $X(t)$:

$$X(t) = \begin{bmatrix} \varphi_M \\ \varpi_M \\ \varphi_L \\ \varpi_L \\ M_M \end{bmatrix}.$$

La variable de salida es la posición angular en el lado del motor como se indica en el vector C.

III. CONTROLADOR CON OBSERVADOR DE ESTADOS

Para el diseño del controlador con observador de estados, se requiere tener el modelo de la planta, el cual provee de todas las medidas de las variables de estado, incluso la variable que sí se la puede medir, en este caso la posición angular del motor. El control con realimentación de estados utiliza las variables estimadas en lugar de las variables reales. La Fig. 1 presenta el esquema de un controlador con observadores de estado. Para el diseño del observador de estados se compara la variable de salida del sistema físico con la variable de salida estimada de la planta; el resultado de esta comparación es multiplicado por un factor correctivo y luego se suma a cada integrador del observador de estados.

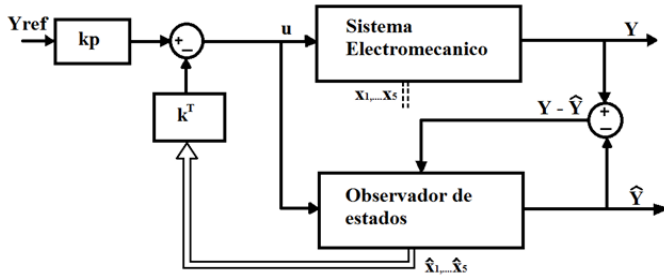


Fig. 1. Controlador por realimentación de estados con Observador de estados.

En las siguientes ecuaciones se utiliza \hat{X} para las variables de estado estimadas y X para las variables de estado reales. La Fig. 2 presenta el observador de estados detallado, en el cual se observa que la diferencia entre la salida real del sistema y la salida estimada, se multiplica por el factor correctivo \hat{k} y luego ingresa a la planta antes de cada integrador.

A. Descripción del sistema

De (8) y (9) se tiene la descripción del sistema en variables de estado, donde $D = 0$. Este sistema tiene una única entrada y una única salida, sistema SISO. Estas ecuaciones se reescriben como mediante (10) y (11), respectivamente.

$$\frac{d}{dt}x = A \cdot x + b \cdot u \quad (10)$$

$$y = c^T \cdot x \quad (11)$$

La entrada $u(t)$ es común al sistema real y al observador de estados, la cual está dada mediante:

$$u = K_p \cdot y_{ref} - k^T \cdot \hat{x} \quad (12)$$

Las variables de estado observadas son utilizadas para diseñar el controlador por realimentación de estados SVF. La ecuación 12 presenta la entrada del sistema $u(t)$ en función de las ganancias de realimentación y de las variables estimadas. Al reemplazar (12) en (10) se tiene:

$$\frac{d}{dt}x = A \cdot x - b \cdot k^T \cdot \hat{x} + b \cdot K_p \cdot y_{ref} \quad (13)$$

B. Descripción del Observador de estados

La ecuación de estados del observador y la ecuación de salida del observador están dadas por:

$$\frac{d}{dt}\hat{x} = \hat{A} \cdot \hat{x} + \hat{b} \cdot u + \hat{k} \cdot (y - \hat{y}) \quad (14)$$

$$\hat{y} = \hat{c}^T \cdot \hat{x} \quad (15)$$

Al reemplazar (12) en (14) se tiene:

$$\frac{d}{dt}\hat{x} = \hat{A} \cdot \hat{x} + \hat{b} \cdot (K_p \cdot y_{ref} - k^T \cdot \hat{x}) + \hat{k} \cdot (y - \hat{y}) \quad (16)$$

Al reemplazar (11) y (15) en (16), y al considerar que los parámetros del observador son iguales a los parámetros del sistema $\hat{A} = A$, $\hat{b} = b$, y $\hat{c} = c$, se tiene:

$$\frac{d}{dt}\hat{x} = (A - b \cdot k^T - \hat{k} \cdot c^T) \cdot \hat{x} + \hat{k} \cdot c^T \cdot x + b \cdot K_p \cdot y_{ref} \quad (17)$$

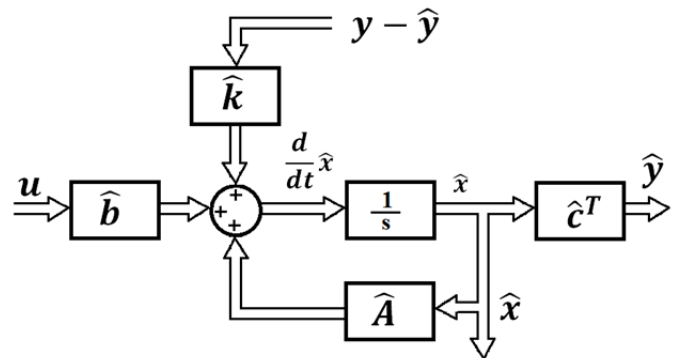


Fig. 2. Observador de estado de orden completo.

Al agrupar (13) y (17) se obtiene la ecuación de estado del controlador con observadores de estado, mientras que (11) es

la ecuación de salida del sistema.

$$\frac{d}{dt} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} = \begin{bmatrix} A & -b \cdot k^T \\ \hat{k} \cdot c^T & A - b \cdot k^T - \hat{k} \cdot c^T \end{bmatrix} \cdot \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} b \cdot K_p \\ b \cdot K_p \end{bmatrix} \cdot y_{ref} \quad (18)$$

C. Localización de los polos del sistema en lazo cerrado

Para encontrar los polos en lazo cerrado se debe describir el sistema en función de las variables de estado reales y del error del observador de estados. El error del observador de estados está dado por:

$$\Delta x = x - \hat{x}, \quad (19)$$

por lo que al aplicar (19) en (17) se obtiene:

$$\frac{d}{dt} \Delta x = (A - \hat{k} \cdot c^T) \cdot \Delta x \quad (20)$$

De (19) se tiene que $\hat{x} = x - \Delta x$, ésta expresión se reemplaza en (13) de manera que quede descrita en función de las variables de estado x y Δx .

$$\frac{d}{dt} x = (A - b \cdot k^T) \cdot x + b \cdot k^T \cdot \Delta x + b \cdot K_p \cdot y_{ref} \quad (21)$$

Al agrupar las ecuaciones (20) y (21), se tiene la descripción dinámica del sistema en lazo cerrado.

$$\frac{d}{dt} \begin{bmatrix} x \\ \Delta x \end{bmatrix} = \begin{bmatrix} A - b \cdot k^T & b \cdot k^T \\ 0 & A - \hat{k} \cdot c^T \end{bmatrix} \cdot \begin{bmatrix} x \\ \Delta x \end{bmatrix} + \begin{bmatrix} b \cdot K_p \\ 0 \end{bmatrix} \cdot y_{ref} \quad (22)$$

De la (22) se obtiene la ecuación característica del sistema en función de las ganancias de la realimentación de estados y de las ganancias de corrección debido al desbalance entre el observador de estados y el sistema real. La ecuación característica está dada por:

$$\left| s \cdot I - (A - b \cdot k^T) \right| \cdot \left| s \cdot I - (A - \hat{k} \cdot c^T) \right| = 0 \quad (23)$$

Para calcular las ganancias de realimentación de estados y las ganancias del factor de corrección, se debe verificar previamente que el sistema sea controlable y observable [8]. La condición de controlabilidad está dada por (24) y la condición de observabilidad está dada por (25), como sigue:

$$|Q_c| = \begin{vmatrix} b & Ab & A^2b & A^3b & \dots & A^{n-1}b \end{vmatrix} \neq 0 \quad (24)$$

$$|\tilde{Q}_{ob}| = \begin{vmatrix} c & A^T c & (A^T)^2 c & \dots & (A^T)^{n-1} c \end{vmatrix} \neq 0 \quad (25)$$

De (23) se tiene que los polos del sistema en lazo cerrado son los polos del controlador con realimentación de estados concatenados con los polos del observador de estados. El cálculo de los polos de realimentación de estados se realiza asumiendo que no existe el observador de estados y de igual modo, los polos del observador se calculan como si no existiese el controlador por realimentación de estados. La concatenación de los polos está dada por:

$$\begin{bmatrix} p^T & \hat{p}^T \end{bmatrix} = [p_1, \dots, p_n \quad \hat{p}_1, \dots, \hat{p}_n] \quad (26)$$

Para el cálculo de las ganancias del controlador de estados se selecciona la ubicación de los polos deseados, para lo cual se utiliza la función de transferencia cuyos coeficientes del denominador se presentan en (27), este tipo de función de transferencia tiene un estrés moderado en el actuador, es decir un sobre impulso mediano.

$$P(s) = 1 + T \cdot s + \frac{1}{2} \cdot T^2 \cdot s^2 + \frac{1}{8} \cdot T^3 \cdot s^3 + \frac{1}{64} \cdot T^4 \cdot s^4 + \frac{1}{512} \cdot T^5 \cdot s^5 \quad (27)$$

Los cálculos de las ganancias del controlador y del observador de estados se realizan separadamente mediante la fórmula de Ackermann [9] dada por:

$$K^T = t^T \cdot \phi(A) \quad (28)$$

$$\phi(A) = A^n + \gamma_{n-1} A^{n-1} + \dots + \gamma_1 A + \gamma_0 I \quad (29)$$

Los coeficientes γ_i son tales que los polos deseados sean las raíces de la ecuación característica dada por:

$$(s - p_1) \cdot (s - p_2) \cdot \dots \cdot (s - p_n) = s^n + \gamma_1 s^{n-1} + \gamma_2 s^{n-2} + \dots + \gamma_{n-1} s + \gamma_n \quad (30)$$

Para el cálculo de las ganancias del controlador por realimentación de estados, t^T está dado por (31) y para el cálculo de las ganancias del factor de corrección del observador de estados, t^T está dado por (32), como sigue:

$$t^T = [0 \quad \dots \quad 0 \quad 1] \cdot [b \quad Ab \quad A^2b \quad \dots \quad A^{n-1}b]^{-1} \quad (31)$$

$$t^T = [0 \quad \dots \quad 0 \quad 1] \cdot [c^T \quad Ac^T \quad A^2c^T \quad \dots \quad A^{n-1}c^T]^{-1} \quad (32)$$

La ganancia K_p que se indica en la Fig. 1 se calcula mediante (33), lo cual permite tener una ganancia igual a uno en estado estable.

$$K_p = \frac{1}{C^T \cdot (-A + B \cdot K^T)^{-1} \cdot B} \quad (33)$$

IV. DISEÑO DE LOS OBSERVADORES DE ESTADOS

Se diseñan dos controladores de estado: 1) Controlador con realimentación de estados sin control proporcional integral PI y 2) Controlador con realimentación de estados y control proporcional integral PI.

A. Observador de estados sin Control PI

La Fig. 3 presenta el diagrama de bloques del observador de estados sin control PI. Este control no tiene realimentación directa de la salida hacia la entrada, pero si existe realimentación de la salida hacia el observador de estados para tener las variables de estado estimadas. Las variables de estado estimadas son utilizadas para encontrar las ganancias de realimentación de estados del controlador por realimentación de estados.

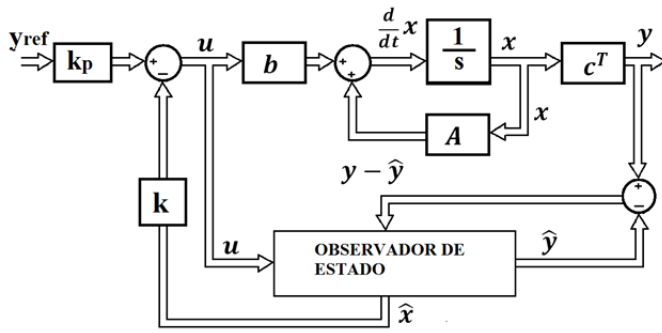


Fig. 3. Observador de estado de orden completo sin control PI.

B. Observador de estados con Control PI

El observador de estados con control PI se diseña de igual modo que el controlador sin control PI, con la diferencia de que este controlador incluye una realimentación directa de la salida, para implementar el control proporcional integral.

El control proporcional integral PI es aplicado a la diferencia entre la salida actual de la planta y la referencia del sistema, en este caso la posición angular del motor. Al igual que el observador de estados sin PI, la señal de entrada a la planta y al observador es $u(t)$. La posición angular del motor es la única variable de estado que se puede medir, las otras variables de estado son calculadas mediante el observador de estados. La Fig. 4 presenta el diagrama de bloques del observador de estados con control PI.

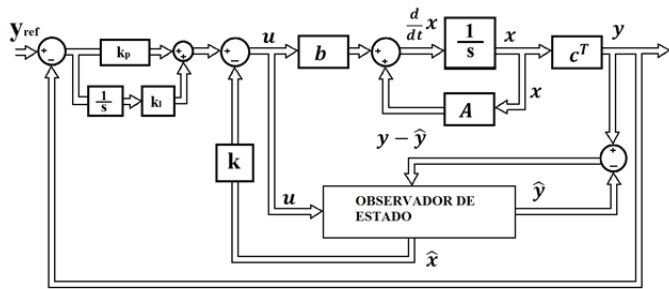


Fig. 4. Observador de estado de orden completo con control PI.

El control proporcional integral permite tener un error en estado estable próximo a cero.

El diseño del controlador de realimentación de estados, se realiza como si no existiera el estimador de estados [10]. El integrador del controlador PI añade una nueva variable de estado al sistema, por lo que el sistema en lazo cerrado contiene once polos, cinco del estimador de estados, cinco de la planta a ser controlada y uno del control PI.

El controlador PI se indica en la Fig. 5, la derivada de la variable de estado x_i del integrador está dada por la diferencia entre la salida actual del sistema y la referencia:

$$\frac{d}{dt}x_i = y_{ref} - c^T \cdot x \quad (34)$$

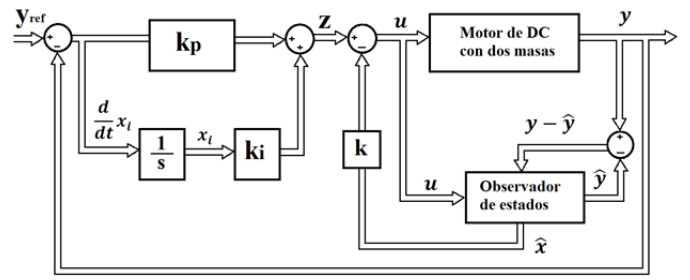


Fig. 5. Diagrama de bloques del controlador con estimadores de estado y PI.

En la Fig. 5, la salida z del PI está dada por:

$$z = \frac{K_p}{T_i} x_i + K_p \cdot (y_{ref} - c^T \cdot x) \quad (35)$$

El valor de T_i se selecciona igual a la constante de tiempo del polinomio característico de la función de transferencia dada por (27), que contiene los polos deseados.

La entrada a la planta y al observador de estados $u(t)$ es:

$$u = z - k^T \cdot x \quad (36)$$

Para el diseño del controlador por realimentación de estados se asume que no existe estimador, por lo que en (36) se utiliza los valores reales de las variables de estado, al considerar que se las tiene disponibles.

$$k_e = [K_p \cdot c^T + k^T - k_i] \quad (37)$$

$$u = K_p \cdot y_{ref} - k_e \cdot \begin{bmatrix} x \\ x_i \end{bmatrix} \quad (38)$$

De (10) y (34) se encuentra la nueva ecuación de estados del sistema. La ecuación de estados del sistema en lazo cerrado del controlador con realimentación de estados y control PI y la ecuación de salida están dadas por:

$$\frac{d}{dt} \begin{bmatrix} x \\ x_i \end{bmatrix} = \begin{bmatrix} A & 0 \\ c^T & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ x_i \end{bmatrix} + \begin{bmatrix} b \\ 0 \end{bmatrix} \cdot u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \cdot y_{ref} \quad (39)$$

$$y = \begin{bmatrix} c^T & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ x_i \end{bmatrix} \quad (40)$$

Las ecuaciones (38), (39) y (40) describen el sistema con controlador de realimentación de estados y controlador proporcional integral PI en el lazo.

El controlador PI incrementa un polo al sistema, el cual se selecciona con el valor necesario para cancelar el cero del controlador PI: $p_6 = -1/T_i$.

De (37) se calcula, la ganancia k_i , la ganancia k_p , y las ganancias de realimentación de estado k^T aplicando la fórmula de Ackermann para los seis polos seleccionados:

$$k_i = -k_{e6} \quad (41)$$

$$K_p = k_i \cdot T_i \quad (42)$$

$$k^T = [k_e(1) \quad \dots \quad k_e(5)]^T - K_p \cdot c^T \quad (43)$$

En Matlab se ha realizado un programa con las expresiones anteriores. La fórmula de Ackermann entrega un vector de ganancias extendidas, dado por (37). El negativo del sexto elemento del vector de ganancias corresponde a la ganancia del integrador según se indica en (41). La ganancia k_p del control PI se calcula con (42) y el vector de ganancias de realimentación de estados está dado por (44).

V. METODOLOGÍA

El diseño de los observadores de estado se lo realiza utilizando las ecuaciones descritas. Mediante Matlab se encuentra las ganancias de realimentación de estados y las ganancias correctivas del observador de estado. Se realiza la simulación en Simulink y luego se carga los bloques del controlador de estados y del observador en el módulo dSPACE ds1104. Se mide la posición angular del motor cuando se aplica una función escalón de referencia. Se mide los parámetros de la respuesta dinámica del sistema a la función escalón: tiempo de establecimiento, sobre impulso, y error en estado estable.

Las mediciones se realizan en el sistema con controlador de realimentación de estados con y sin observador de estados. Se realiza una comparación del efecto de los observadores de estado sobre la respuesta dinámica del sistema, y también se realiza una comparación de los dos controladores con observadores de estado.

VI. PRUEBAS REALIZADAS Y ANÁLISIS DE RESULTADOS

Para realizar las pruebas de los controladores se aplica una función escalón como referencia de la posición angular del motor y se mide la respuesta dinámica de la posición angular actual en el lado del motor. Se realiza cuatro controladores, dos controladores sin observador de estados y dos controladores con observador de estados. Los controladores implementados se diseñan en Matlab y se aplican sobre el motor físico por medio del módulo dSPACE, para lo cual se utiliza una interfaz con el motor como se indica en la Fig. 6, Fig. 8, Fig. 10 y Fig. 12. El bloque de interfaz con el motor contiene como entrada el torque aplicado al motor, y como salidas: la posición angular en el lado del motor, la posición angular en el lado de la carga, la velocidad angular en el lado del motor, la velocidad angular en el lado de la carga y el torque del motor.

A. Controlador con realimentación de estados SVF sin controlador PI

La Fig. 6 presenta el diagrama de bloques implementado en Matlab del controlador con realimentación de estados sin control proporcional integral PI. En este caso, no se dispone del observador de estados, por lo tanto las variables de estado son las variables reales del motor físico. Se realiza este controlador con el objetivo de comparar los valores obtenidos cuando este controlador tenga un estimador de estados.

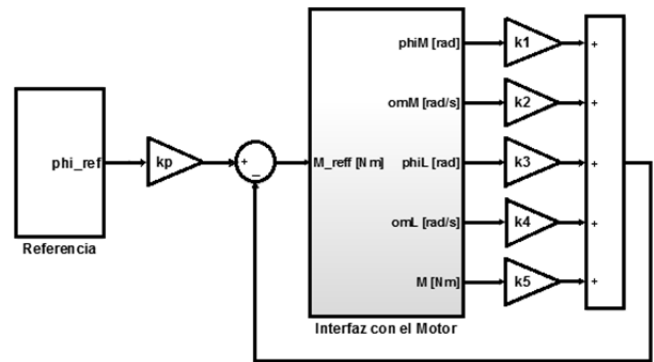


Fig. 6. Diagrama de bloques del controlador SVF.

La Fig. 7 presenta la respuesta a la función escalón aplicada al motor físico y al modelo de la planta en Simulink.

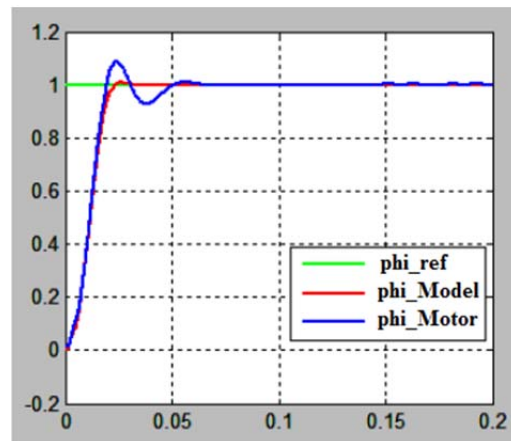


Fig. 7. Respuesta a la función paso del controlador SVF [6].

La Tabla III presenta los resultados de la respuesta a la función escalón para un controlador SVF simple aplicado al motor de DC.

TABLA III
CONTROLADOR SVF SIMPLE [6]

Parámetro	
Tiempo de establecimiento en segundos	0.063
Error en estado estable	0.028%
Sobre impulso	8.73%

B. Observador de estados aplicado al SVF simple

La Fig. 8 presenta el diagrama de bloques implementado en Matlab del controlador con realimentación de estados y observador de estados sin control proporcional integral PI. El observador de estados, proporciona una estimación de estados de las cinco variables de estado. La única variable medida en el motor físico es la posición angular en el lado del motor. El control por realimentación de estados se realiza utilizando las variables de estado estimadas.

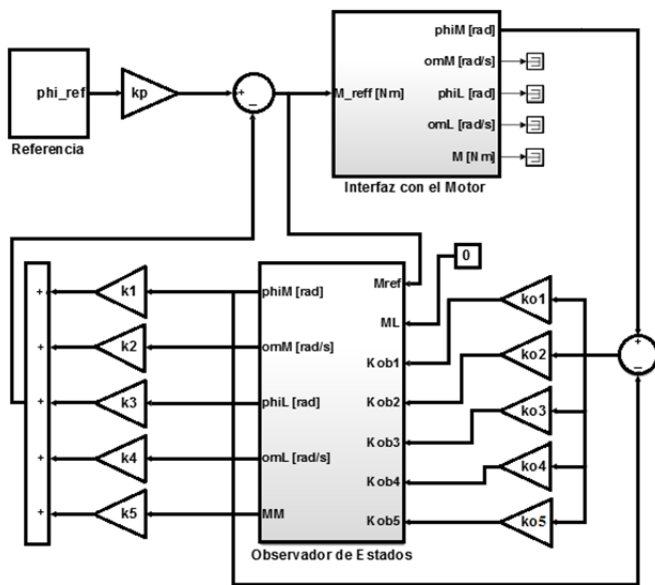


Fig. 8. Diagrama de bloques del controlador SVF con estimador de estados.

La Tabla IV presenta los resultados de la respuesta a la función escalón para un controlador SVF simple con observadores de estado de orden completo aplicado al motor físico de DC.

TABLA IV
CONTROLADOR SVF SIMPLE CON OBSERVADOR DE ESTADOS [6]

Parámetro	
Tiempo de establecimiento en segundos	0.063
Error en estado estable	0.041%
Sobre impulso	10.3%

La Fig. 9 presenta la respuesta a la función escalón aplicada al motor físico y al modelo de la planta en Simulink.

Al aplicar el observador de estados al control SVF, tratando de mantener el mismo tiempo de establecimiento se tiene un pequeño incremento en el error en estado estable y en el sobre impulso. Esto se observa tanto en el motor físico como en el modelo matemático del sistema.

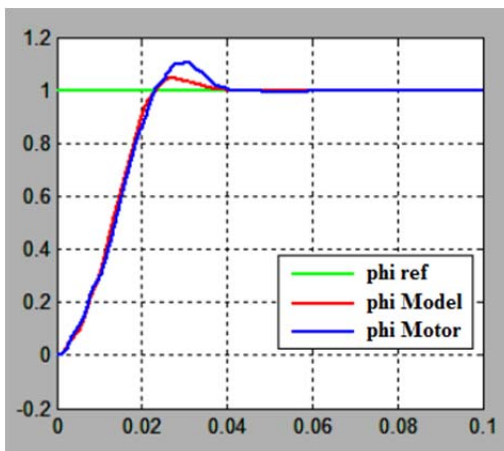


Fig. 9. Respuesta a la función paso del estimador de estados [6].

C. Controlador con realimentación de estados SVF con controlador PI

La Fig. 10 presenta el diagrama de bloques implementado en Matlab del controlador con realimentación de estados con control proporcional integral PI. En este caso, no se dispone de observador de estados, por lo que las variables de estado son las variables reales del motor físico. Se realiza este controlador con el objetivo de comparar los valores obtenidos cuando a este controlador se le aplique un estimador de estados.

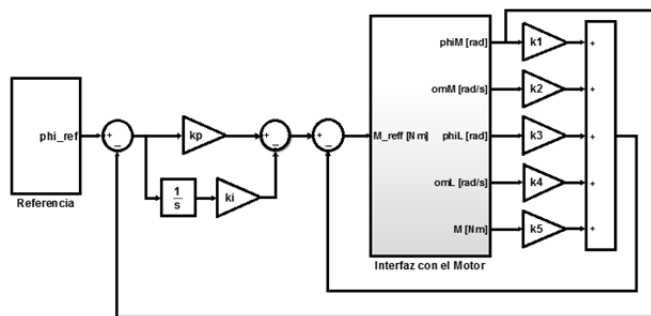


Fig. 10. Diagrama de bloques del controlador SVF-PI.

La Fig. 11 presenta la respuesta a la función escalón aplicada al motor físico y al modelo de la planta en Simulink. El controlador se aplica en forma paralela a la planta del sistema y al motor físico para visualizar ambas respuestas.

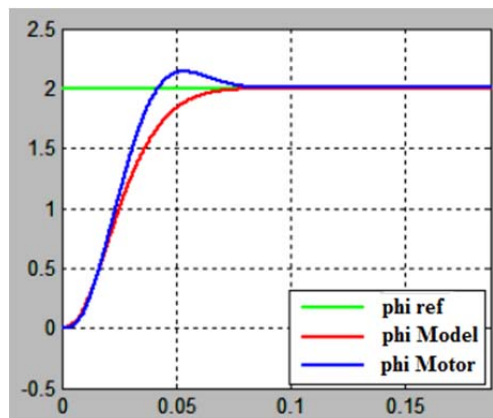


Fig. 11. Respuesta a la función paso del controlador SVF-PI [6].

La Tabla V presenta los resultados de la respuesta a la función escalón para un controlador con control PI incluido, aplicado al motor físico de DC.

TABLA V
CONTROLADOR SVF CON CONTROL PI [6]

Parámetro	
Tiempo de establecimiento en segundos	0.08
Error en estado estable	0.003%
Sobre impulso	6.97%

D. Observador de estados con controlador PI

La Fig. 12 presenta el diagrama de bloques implementado en Matlab del controlador con realimentación de estados con

control PI integrado, y observador de estados. El observador de estados proporciona una estimación de estados de las cinco variables de estado, al igual que en el caso anterior, la única variable medida en el motor físico es la posición angular en el lado del motor, como se indica en la Fig. 12 en el bloque de interfaz con el motor. El observador tiene cinco variables de estado estimadas que son utilizadas para el control por realimentación de estados.

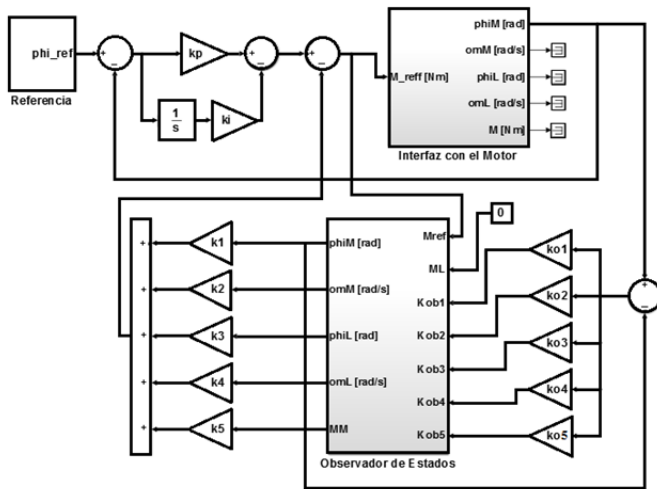


Fig. 12. Diagrama de bloques del Observador de estados con control PI.

La Fig. 13 presenta la respuesta a la función escalón aplicada al motor físico y al modelo de la planta en Simulink.

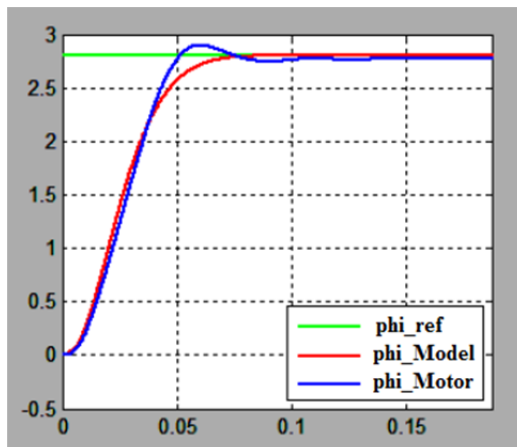


Fig. 13. Respuesta a la función paso del Observador de estados con PI [6].

La Tabla VI presenta los resultados de la respuesta a la función escalón para un controlador con control PI incluido aplicado al motor físico.

TABLA VI
OBSERVADOR DE ESTADOS CON CONTROL PI [6]

Parámetro	
Tiempo de establecimiento en segundos	0.13
Error en estado estable	0.01%
Sobre impulso	3.48%

En este caso, se ha obtenido una reducción del sobre

impulso debido al incremento en el tiempo de establecimiento. Como se presenta en la Tabla VI se ha incrementado el error en estado estable, respecto al controlador SVF sin observador de estados.

El controlador PI hace que el sistema tenga una respuesta dinámica más lenta respecto al controlador SVF simple debido a la presencia del integrador.

E. Comparación de Resultados

La Tabla VII presenta un resumen de los parámetros de respuesta dinámica de los dos controladores con y sin estimador de estados. Los controladores con control PI integrado respecto a los controladores sin PI, tienen tiempo de establecimiento más alto, error en estado estable menor y sobre impulso menor.

TABLA VII
CONTROLADOR SVF y SVF PI [6]

Parámetro	SVF	SVF PI	SVF con Observador	SVF-PI con Observador
Tiempo de establecimiento en segundos	0.063	0.08	0.06	0.13
Error en estado estable	0.028%	0.003%	0.041%	0.01%
Sobre impulso	8.73%	6.97%	10.3%	3.48%

El estimador de estados tiene un efecto negativo en el tiempo de establecimiento debido a que se incluye nuevos polos en el sistema en lazo cerrado, sin embargo el efecto no es significativo debido a que en el diseño se ha seleccionado la ubicación de los polos en un valor diez veces más lejos en el semiplano izquierdo del plano s , que los polos del sistema en lazo cerrado.

VII. CONCLUSIONES

Mediante el Real Time Workshop de Matlab se puede programar el módulo dSPACE, los controladores simulados en Simulink Matlab se cargan directamente en el hardware. Esto ahorra gran cantidad de tiempo en la implementación física de los controladores desarrollados, al no tener que programar en lenguaje C los controladores. Los observadores de estado son de gran utilidad cuando no se dispone de todas las variables de estado, y es posible identificar el modelo matemático de la planta. En este trabajo se ha realizado estimadores de estado de orden completo, sin embargo en el caso de tener un cierto número de variables de estado conocidas, se puede realizar un estimador de estados de orden reducido. El control con variables de estado diseñado en este trabajo es un control con una sola entrada y una sola salida. Para controladores con múltiples entradas y salidas, también se puede implementar alguna técnica basada en control con variables de estado como es el caso del regulador lineal cuadrático LQR. Para este sistema electromecánico se dispone de un modelo matemático apropiado, por lo tanto el diseño del observador de estados se ha realizado correctamente con un desempeño bastante similar al controlador sin estimador de estados, en el cual todas las variables de estado son medidas. El estimador de estados no afecta en forma significativa la respuesta dinámica del

sistema, ya que los polos del estimador de estados tienen constantes de tiempo de valores diez veces menores que las de los polos del controlador SVF.

RECONOCIMIENTOS

A la Universidad de Ciencias Aplicadas Esslingen de Alemania, por la prestación tecnológica para realizar este proyecto.

REFERENCIAS

- [1] U. Manwong, S. Boonpiyathud, y S. Tunyasirut, "Implementation of a dSPACE DSP-based state feedback with state observer using Matlab/Simulink for a speed control of DC motor system," in *2008 International Conference on Control, Automation and Systems*, Seoul, South Korea, Oct. 2008, pp. 2433-2436.
- [2] R. Salas-Cabrera, J. D. León-Morales, J. C. Mayo-Maldonado, J. C. Rosas-Caro, E. N. Salas-Cabrera, y C. A. Reyna-López, "Observer Design for DC Electric Machines," in *2009 Second International Conference on Computer and Electrical Engineering*, Dubai, United Arab Emirates, Dec. 2009, pp. 90-94.
- [3] G. Fabri, C. Olivieri, y M. Tursini, "Observer-based sensorless control of a five-phase brushless DC motor," in *The XIX International Conference on Electrical Machines - ICEM 2010*, Rome, Italy, Sep. 2010, pp. 1-6.
- [4] K. Meah, S. Hietpas, y S. Ula, "Rapid Control Prototyping of a Permanent Magnet DC Motor Drive System using dSPACE and Mathworks Simulink," in *APEC 07 - Twenty-Second Annual IEEE Applied Power Electronics Conference and Exposition*, Anaheim, CA, USA, Mar. 2007, pp. 856-861.
- [5] P. Hippe y J. Deutscher, *Design of Observer-based Compensators: From the Time to the Frequency Domain*. London, UK: Springer London, 2009.
- [6] A. Palacios, "Modeling, Identification and Controller Design of an Electrical Servo System with a Two-Mass Characteristic," M.S. thesis, University of Applied Science Esslingen, Germany, 2010.
- [7] H. Lutz y W. Wendt, *Taschenbuch der Regelungstechnik: mit MATLAB und Simulink*. Europa Lehrmittel Verlag, 2007.
- [8] K. Ogata, *Modern Control Engineering*. Prentice Hall, 2010.
- [9] R. L. W. II y D. A. Lawrence, *Linear State-Space Control Systems*. John Wiley & Sons, 2007.
- [10] H. P. Geering, *Regelungstechnik: Mathematische Grundlagen, Entwurfsmethoden, Beispiele*. Springer-Verlag, 2013.

On the Performance Improvement of the Optimal-Sampling-inspired Self-Triggered Control at Implementation Stage

Mejoramiento del Desempeño del Control Auto-Disparado Inspirado en Muestreo Óptimo en la Fase de Implementación

Carlos Xavier Rosero, *Member, IEEE*, Cristina Vaca and Juan Benavides

Abstract—The self-triggered control includes a sampling strategy that focuses on decreasing the use of computational resources (processor and network) while preserving the same control performance as the one obtained via a controller with periodic sampling. Within this framework it has been developed recently a self-triggered control technique inspired by a sampling pattern whose optimal density minimizes the control cost, this approach is called “optimal-sampling inspired self-triggered control”. However, the strategies used to implement it on microprocessor-controlled systems working under perturbation are still unclear; this paper addresses some techniques to organize and improve the implementation on actual controllers. The proposed solution comprises both the formulation of two algorithms to organize the implementation and the insertion of a closed-loop observer to deal with the perturbation that normally appears on real plants. Regarding the former, certain computationally expensive processes involved in the implementation of this control technique are treated through their replacement by lightweight polynomials fitted at design stage. Simulations and practical experiments confirm the solution is effective and there could be an open research topic concerning observation in optimal-sampling self-triggered control strategies.

Index Terms—Event-driven control, aperiodic sampling, real-time embedded control systems, aperiodic observer.

Resumen—El control auto-disparado incluye una atractiva estrategia de muestreo que se enfoca en disminuir el uso de recursos computacionales (procesador y red) mientras se preserva el mismo rendimiento de control que el obtenido a través de un controlador con muestreo periódico. Dentro de este marco, se ha desarrollado recientemente una técnica de control auto-disparado inspirada en un patrón de muestreo cuya densidad óptima minimiza el costo de control, se llama “control auto-disparado inspirado en muestreo óptimo”. Sin embargo, las estrategias utilizadas para implementarlo en sistemas controlados por microprocesadores que funcionan bajo perturbación aún no son claras; este documento aborda algunas técnicas para organizar y mejorar la implementación sobre controladores reales. La solución propuesta comprende la formulación de dos algoritmos para organizar la implementación y también la inserción de un observador de lazo cerrado para lidiar con las perturbaciones que normalmente aparecen en las plantas reales. En cuanto a los algoritmos, ciertos procesos computacionalmente costosos implicados en su implementación son tratados mediante

la sustitución por polinomios ligeros ajustados en la fase de diseño. Tanto simulaciones como experimentos confirman que la solución es efectiva y que podría haber un tema de investigación abierto relacionado con la observación en las estrategias de control auto-disparado con muestreo óptimo.

Palabras Claves—Control manejado por eventos, muestreo aperiódico, sistemas de control empujados de tiempo real, observador aperiódico.

I. INTRODUCTION

NOWADAYS controllers are implemented on digital systems consisting of microprocessors and communication networks. Among some of the alternatives that have efficient resource consumption in a nonperiodic fashion are the self-triggered control techniques (STC), initially proposed by [1]–[5]. They solve the fundamental problem of determining both optimal sampling and efficient processing/communication strategies. Each time the control task is triggered, both the time the next sampling will be performed (sampling rule) and the control action which should be maintained until this event happens, are estimated.

Several approaches aimed at solving the problem of determining optimal sampling rules in STC have been addressed recently. An optimal sampling pattern proposed in [6] inspired the approach in [7], which is analyzed in the present study. This technique describes a sampling rule that generates approximated control actions by solving the continuous-time LQR problem [8] at each sample time. The performance guarantee is based on a number of samples over a time interval with a given sampling constraint. The sampling time is calculated by the derivative of a continuous-time LQR problem and the rule produces smaller sampling times while the control action has more variation.

Though the optimal-sampling in [6] and [7] has standard cost lower than the one obtained by periodic sampling techniques, and even than other optimal-sampling approaches i.e. [9], [10], it has many weaknesses. Since the research is still new there are many open topics, among which two stand out: (a) clarifying and organizing the implementation on real microprocessor systems, and (b) adapting the approach to cases with disturbances.

C.X. Rosero and J. Benavides are with the Applied Sciences Faculty, Universidad Técnica del Norte, Av. 17 de Julio 5-21, 100105 Ibarra, Ecuador (e-mail: cxrosoero@utn.edu.ec).

C. Vaca is with the Health Sciences Faculty, Universidad Técnica del Norte, Av. 17 de Julio 5-21, 100105 Ibarra, Ecuador (e-mail: cvaca@utn.edu.ec).

To solve problem (a), in [7] both a simulated and an experimental set-ups are described. However, a deeper explanation of the paradigm that a designer of control systems should use to put this approach on a microprocessor-based system is not shown.

With regard to problem (b) the approach in [6] could be restated by inserting robustness to uncertainty in the approach by developing new theory, or on the other hand by using observation techniques. A settlement applying observation in presence of unknown disturbances but on a different STC strategy to that used herein, is presented in [11], [12].

To overcome problems (a) and (b), the contribution of this paper is twofold. First, two algorithms are formulated to organize and synthesize the implementation of the approach in [7]. Second, a time-varying closed-loop observer is applied on the approach in [7] in order to make it less sensitive to noise.

The rest of the paper is organized as follows. Section II summarizes the theory on optimal-sampling-inspired self-triggered control (OSISTC). Section III presents the insertion of state observation into the self-triggered control and also the strategies to describe the implementation. Section IV shows the simulations and experiments on a selected plant. At the end, Section V performs the analysis of results and Section VI concludes the article.

II. REVISITING THE OPTIMAL-SAMPLING-INSPIRED SELF-TRIGGERED CONTROL

This section summarizes the theory on OSISTC extracted from the original works in [6] and [7], and included for better understanding of the subject of study.

A. Continuous-time dynamics

Consider the linear time-invariant system (LTI) represented in continuous-time by

$$\begin{cases} \dot{x}(t) = A_c x(t) + B_c u(t), & \text{given } x(0) = x_0 \\ y(t) = C x(t) \end{cases} \quad (1)$$

where $x(t) \in \mathbb{R}^n$ is the state and $u(t) \in \mathbb{R}^m$ is the continuous control input signal. $A_c \in \mathbb{R}^{n \times n}$ and $B_c \in \mathbb{R}^{n \times m}$ describe the dynamics of the system, and $C \in \mathbb{R}^{q \times n}$ is the weight matrix used to read the state; x_0 is the initial value of the state.

B. Sampling

The control input $u(t)$ in (1) is piecewise constant, meaning that it remains with the same value between two consecutive *sampling instants*, thus

$$u(t) = u(k) \quad \forall t \in [t_{k-1}, t_k), \quad (2)$$

where the control input $u(k)$ is updated at discrete times $k \in \mathbb{N}$ and the sampling instants are represented by $t_k \in \mathbb{R}$. Consecutive sampling instants are separated by *sampling intervals* τ_k , and the relationship between instants and intervals is

$$\tau_k = t_{k+1} - t_k, \quad \text{where } t_k = \sum_{i=0}^{k-1} \tau_i \quad \text{for } k \geq 1. \quad (3)$$

C. Discrete-time dynamics

In periodic sampling, a constant sampling interval τ is considered. The continuous-time dynamics from (1) is discretized using methods taken from [8] by

$$A_d = e^{A_c \tau}, \quad B_d = \int_0^\tau e^{A_c(\tau-t)} dt B_c, \quad (4)$$

resulting in the discrete-time LTI system

$$\begin{cases} x(k+1) = A_d x(k) + B_d u(k), \\ y(k) = C x(k) \end{cases}, \quad \text{given } x(0) = x_0 \quad (5)$$

where the state $x(k)$ is sampled at t_k .

The location of the system poles (or eigenvalues of the dynamics matrices A_c, A_d) is fundamental to determine/change the stability of the system [8]. Poles in continuous-time p_c become poles in discrete-time p_d through

$$p_d = e^{p_c \tau}. \quad (6)$$

State-feedback control by means of pole placement requires to assign the desired closed-loop poles by hand. Nevertheless, the LQR technique allows to place the poles automatically and optimally. LQR is used by OSISTC at each t_k considering τ_k as the sampling time.

D. Linear quadratic regulator

The LQR optimal control problem allows to find an optimal input signal that minimizes the continuous-time and discrete-time infinite-horizon cost functions in (7) and (8) respectively.

$$J_c = \int_0^\infty (x(t)^T Q_c x(t) + 2x(t)^T S_c u(t) + u(t)^T R_c u(t)) dt, \quad (7)$$

$$J_d = \sum_0^\infty (x(k)^T Q_d x(k) + 2x(k)^T S_d u(k) + u(k)^T R_d u(k)) \quad (8)$$

Regarding dimensionality in (7) and (8), the weight matrices $Q_c, Q_d \succeq 0 \in \mathbb{R}^{n \times n}$ are positive semi-definite, $R_c, R_d \succ 0 \in \mathbb{R}^{m \times m}$ are positive definite, and $S_c, S_d \in \mathbb{R}^{n \times m}$. Refer to [13] to know about the transformation of the weight matrices from their continuous forms Q_c, R_c, S_c to their discrete versions Q_d, R_d, S_d .

E. Optimal sampling-inspired self-triggered control

The approach in [7] involves designing both a sampling rule as a piecewise control input, such that the LQR cost is minimized. Additionally, the periodicity of execution of the controller is relaxed so that the consumption of resources is diminishing. Then, the *sampling rule* is

$$\tau_k = \tau_{max} \frac{1}{\frac{\tau_{max}}{\eta} |K_c(A_c + B_c K_c)x(k)|^\alpha + 1} \quad (9)$$

where an upper bound on the sampling intervals is given by τ_{max} ; similarly η modifies the degree of density of the sampling sequence (smaller η yields denser sampling instants and vice versa). By minimizing the continuous-time cost function (7) an optimal continuous-time feedback gain K_c is found once. According to [6] and [7] there exist optimal settings for the exponent $\alpha \geq 0$ which influences the density

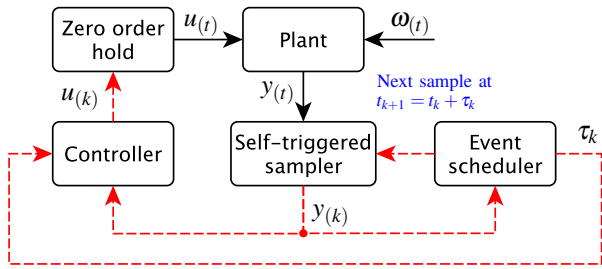


Fig. 1. Original architecture of the self-triggered feedback control. Solid lines denote continuous-time signals and dashed lines denote signals updated only at each sampling time.

of the samples set; with $\alpha = 0$ the sampling becomes regular (periodic).

Additionally, from [7] the *piecewise optimal control signal* expressed in linear state feedback form is

$$u(k) = -K_d(\tau_k)x(k), \quad (10)$$

where $K_d(\tau_k)$ is calculated at each controller execution τ_k . Its value is obtained by solving the discrete-time LQR problem (8) considering a fixed sampling period τ_k .

III. ON THE IMPLEMENTATION OF OSISTC

The model of the proposed approach as well as the guidelines for its implementation are explained in this section. This corresponds to the main contribution of the work.

A. Original OSISTC architecture

Figure 1 is used to ensure better understanding of the original OSISTC scheme. In this configuration the output of the plant $y(t)$ is sampled by the *self-triggered sampler* at each τ_k ; the measured state $y(k)$ is used by both the *event scheduler* and the *controller*. The event scheduler is responsible for calculating when the next sampling time t_{k+1} will be executed by means of (9). The controller computes the control action using both (2) and (10). The control input $u(k)$ is kept constant along the entire sampling interval τ_k in a zero-order hold manner.

In the same Figure 1, the bounded exogenous disturbances $\omega(t)$ are not treated in any way, causing noisy states and affecting the system performance. With respect to both the event scheduler and the controller, they base their procedures on the measured state $y(k)$ (or on the error $e(k)$ when there is a reference). Thus, the insertion of noise into the states leads to the emergence of uncertainty in both the linear piece-wise control $u(k)$ and the sampling interval τ_k .

B. Discrete-time observer

An observer constitutes a computer copy of the observed dynamic system (5) whose predicted states $\hat{x}(k)$ converge to the real states $x(k)$ by reducing the observer's output error $\tilde{e}(k)$. The

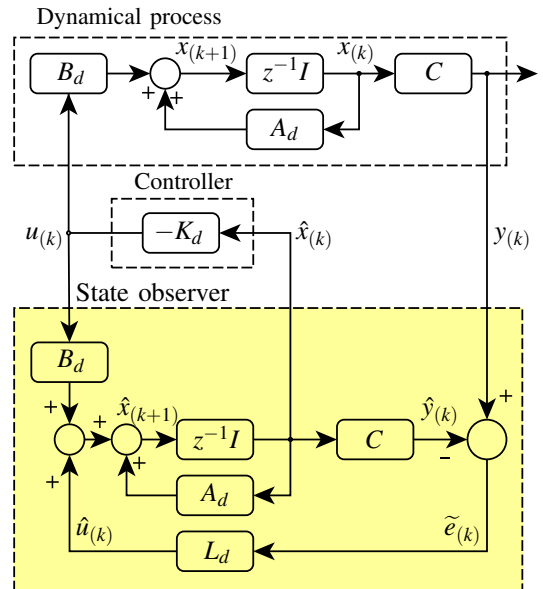


Fig. 2. Discrete-time Luenberger state observer

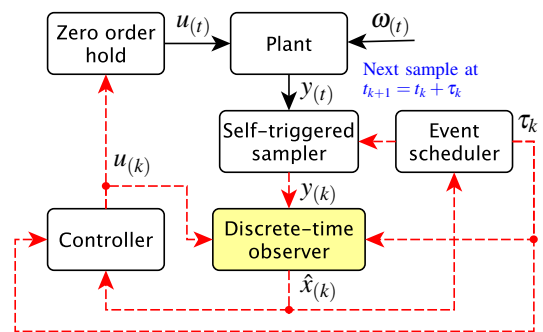


Fig. 3. Proposed architecture of the self-triggered feedback control with observation. Solid lines denote continuous-time signals and dashed lines denote signals updated only at each sampling time.

discrete-time Luenberger observer proposed in [14] and shown in Fig. 2 is a state estimator which works properly in presence of unknown disturbances; see [8] for better understanding. Then, the system in (5) is reformulated as

$$\begin{cases} \hat{x}(k+1) = A_d \hat{x}(k) + B_d u(k) + L_d (y(k) - \hat{y}(k)) \\ y(k) = Cx(k) \end{cases}, \quad (11)$$

where $\hat{x}(k+1) \in \mathbb{R}^n$ is the state estimate and $\hat{y}(k) \in \mathbb{R}^q$ is the output estimate. $L_d \in \mathbb{R}^{n \times q}$ is the observer gain matrix.

In (11), if the pair (A_d, C) is completely observable, the dual system (A_d', B_d', C', D') is completely reachable. Then, an observer gain matrix L_d for the dual system can be designed and the eigenvalues (poles) of $(A_d' - C'L_d)$ can be arbitrarily placed [14]. Consider that the eigenvalues of a matrix $(A_d' - C'L_d)$ are equal to the eigenvalues of its transpose $(A_d - L_d'C)$.

C. Proposed OSISTC architecture based on observer

Figure 3 shows the proposed self-triggered architecture in which the use of a *discrete-time observer* stands out to deal with noise $\omega(t)$. Assuming that the pair $(A_d(\tau_k), C)$ is observable along the set of all possible sampling intervals τ_k , the eigenvalues of $(A_d(\tau_k) - L_d'(\tau_k)C)$ can be placed arbitrarily

[14]. Notice that the dynamics now depends on τ_k because $A_{d(\tau_k)}$ is a time varying matrix. The discrete poles in (6) are also dependent on the sampling interval τ_k , as in

$$p_{d(\tau_k)} = e^{p_c \tau_k}. \quad (12)$$

In this context the observer needs to solve a new pole placement at each execution, since the discrete dynamics matrices and the discrete poles are dependent on the sampling interval τ_k . This implies that the observer has a different gain matrix $L_{d(\tau_k)}$ at each execution. Then, considering the changing dynamics, the system in (11) becomes

$$\begin{cases} \hat{x}_{(k+1)} = A_{d(\tau_k)} \hat{x}_{(k)} + B_{d(\tau_k)} u_{(k)} + L_{d(\tau_k)} (y_{(k)} - \hat{y}_{(k)}) \\ y_{(k)} = C x_{(k)} \end{cases}, \quad (13)$$

where $A_{d(\tau_k)}$ and $B_{d(\tau_k)}$ are discretized matrices for a sampling interval τ_k , $u_{(k)}$ is the linear piecewise control action calculated by (10), and $L_{d(\tau_k)} \in \mathbb{R}^{n \times q}$ is the gain matrix of the sampling-dependent observer.

D. Problems considered

There are several drawbacks in assembling both the OSISTC controller and the time-varying observer on a real-time control system.

The first issue has to do with calculation of the *controller gain matrix* $K_{d(\tau_k)}$ in (10) by solving the problem in (8) through recursive computation of the discrete algebraic Ricatti equation (DARE) until convergence [8]. The second issue is the pole placement solved by Ackermann's formula [15] in order to obtain the *observer gain matrix* $L_{d(\tau_k)}$.

Both processes are computationally expensive and must be performed at each controller execution. If the execution time of the control task is too close to the minimum sampling interval, undesirable effects such as jitter could appear [16]. Particularly in OSISTC, the worst case scenario comes out when the rate of change of the control action is maximal, causing a highest density in the emergence of samples (minimum τ_k).

E. Set of sampling intervals T

The *set of sampling intervals* $T \in \mathbb{R}^{1 \times s}$ within a closed interval $[\tau_{min}; \tau_{max}]$ is

$$T = \{\tau_{min}, \tau_{min} + \tau_g, \tau_{min} + 2\tau_g, \dots, \tau_{max}\}, \quad (14)$$

where τ_g is the *sampling granularity* defined as the least increase-unit for the sampling intervals. Each element of the set can be addressed in this way

$$\tau_h = T_{[h]} \quad \forall h \in \mathbb{N} : 1 \leq h \leq s \quad (15)$$

being s the length of T .

The minimum and maximum sampling times, τ_{min} and τ_{max} , as well as τ_g are chosen following the conditions detailed in [7]

$$\begin{cases} \frac{\beta^\alpha}{\eta} \leq \frac{1}{\tau_{min}} - \frac{1}{\tau_{max}}, & \beta := \sup_{x \in X} |K_c(A_c + B_c K_c)x| \\ \tau_{RTOS} \leq \tau_g, \end{cases} \quad (16)$$

where X is the entire state space taken from the physical constraint of the plant, and τ_{RTOS} is the sampling granularity of the real-time operating system (RTOS) in which the technique will be implemented.

F. Strategy to calculate the controller gain matrix $K_{d(\tau_k)}$

The gain $K_{d(\tau_h)}$ is calculated by brute force for each h^{th} element of the set T in (14) by the discrete-time LQR problem (8). Therefore, we obtain a total of s controller gain matrices $K_{d(\tau_h)} \in \mathbb{R}^{m \times n}$ that have the form

$$K_{d(\tau_h)} = dlqr(A_{d(\tau_h)}, B_{d(\tau_h)}, Q_{d(\tau_h)}, R_{d(\tau_h)})$$

$$K_{d(\tau_h)} = \begin{bmatrix} kd_{11}^{(\tau_h)} & \dots & kd_{1n}^{(\tau_h)} \\ \vdots & \ddots & \vdots \\ kd_{m1}^{(\tau_h)} & \dots & kd_{mn}^{(\tau_h)} \end{bmatrix}. \quad (17)$$

Regrouping the elements of all gain matrices according to their position yields a group S_{K_d} that is $m \cdot n$ training sets long, where m and n are the dimensions of inputs and states respectively, then

$$S_{K_d} = \{[kd_{11}^{(\tau_1)}, \dots, kd_{11}^{(\tau_s)}], \dots, [kd_{1n}^{(\tau_1)}, \dots, kd_{1n}^{(\tau_s)}], \dots, [kd_{m1}^{(\tau_1)}, \dots, kd_{m1}^{(\tau_s)}], \dots, [kd_{mn}^{(\tau_1)}, \dots, kd_{mn}^{(\tau_s)}]\}. \quad (18)$$

Each training set in (18) is defined in $\mathbb{R}^{1 \times s}$ and used to perform a *polynomial curve fitting* in order to find the coefficients θ of the d -degree polynomials $K_{ij}(\tau_k)$. Therefore, we have a total of $m \cdot n$ polynomials each one following the form

$$K_{ij}(\tau_k) = \theta_1^{(ij)} \tau_k^d + \theta_2^{(ij)} \tau_k^{d-1} + \dots + \theta_d^{(ij)} \tau_k + \theta_{d+1}^{(ij)}, \quad (19)$$

where superscript (ij) indicates the belonging of coefficients $\theta^{(ij)}$ to polynomial $K_{ij}(\tau_k)$; i -row and j -column show the position of polynomials into the gain matrix. Note the change of τ_k instead of τ_h since the former is the current sampling interval calculated online through equation (9) on a real controller. Thus, (17) to (19) become

$$K_{d(\tau_k)} = \begin{bmatrix} K_{11}(\tau_k) & \dots & K_{1n}(\tau_k) \\ \vdots & \ddots & \vdots \\ K_{m1}(\tau_k) & \dots & K_{mn}(\tau_k) \end{bmatrix}, \quad (20)$$

where

$$K_{11}(\tau_k) = \theta_1^{(11)} \tau_k^d + \theta_2^{(11)} \tau_k^{d-1} + \dots + \theta_d^{(11)} \tau_k + \theta_{d+1}^{(11)}$$

$$K_{1n}(\tau_k) = \theta_1^{(1n)} \tau_k^d + \theta_2^{(1n)} \tau_k^{d-1} + \dots + \theta_d^{(1n)} \tau_k + \theta_{d+1}^{(1n)}$$

$$K_{m1}(\tau_k) = \theta_1^{(m1)} \tau_k^d + \theta_2^{(m1)} \tau_k^{d-1} + \dots + \theta_d^{(m1)} \tau_k + \theta_{d+1}^{(m1)}$$

$$K_{mn}(\tau_k) = \theta_1^{(mn)} \tau_k^d + \theta_2^{(mn)} \tau_k^{d-1} + \dots + \theta_d^{(mn)} \tau_k + \theta_{d+1}^{(mn)}.$$

G. Strategy to calculate the observer gain matrix $L_{d(\tau_k)}$

It is a process similar to that described in subsection III-F. All possible observer gain matrices $L_{d(\tau_h)}$ are evaluated offline as functions of sampling intervals τ_h .

The error dynamics of the observer is given by the poles of $(A_{d(\tau_h)} - L_{d(\tau_h)} C)$. A rule of thumb considers to place the observer poles five to ten times farther to the left of s -plane than the dominant poles of the system.

By computing $A_{d(\tau_h)}$ through (4), assigning statically the continuous-time poles and discretizing them by (6) in order to have the vector $P_{d(\tau_h)}$, and finally considering C which remains

constant, we obtain a total of s observer gain matrices $L_d(\tau_h) \in \mathbb{R}^{n \times q}$ by the poles placement method in [15] with the form

$$L_d(\tau_h) = \text{ackermann}(A_d(\tau_h)^T, C^T, P_d(\tau_h)^T)$$

$$L_d(\tau_h) = \begin{bmatrix} ld_{11}^{(\tau_h)} & \cdots & ld_{1q}^{(\tau_h)} \\ \vdots & \ddots & \vdots \\ ld_{n1}^{(\tau_h)} & \cdots & ld_{nq}^{(\tau_h)} \end{bmatrix}. \quad (21)$$

Using the same regrouping criterion as in (18) a group S_{L_d} , $n \cdot q$ training sets long, is obtained

$$S_{L_d} = \{[ld_{11}^{(\tau_1)}, \dots, ld_{11}^{(\tau_s)}], \dots, [kd_{1q}^{(\tau_1)}, \dots, kd_{1q}^{(\tau_s)}], \dots, [kd_{n1}^{(\tau_1)}, \dots, kd_{n1}^{(\tau_s)}], \dots, [kd_{nq}^{(\tau_1)}, \dots, kd_{nq}^{(\tau_s)}]\}. \quad (22)$$

Subsequently a total of $n \cdot q$ polynomials are calculated with the form

$$L_{ij}(\tau_k) = \theta_1^{(ij)} \tau_k^d + \theta_2^{(ij)} \tau_k^{d-1} + \dots + \theta_d^{(ij)} \tau_k + \theta_{d+1}^{(ij)}, \quad (23)$$

such as in (19). Finally it is obtained

$$L_d(\tau_k) = \begin{bmatrix} L_{11}(\tau_k) & \cdots & l_{1q}(\tau_k) \\ \vdots & \ddots & \vdots \\ L_{n1}(\tau_k) & \cdots & l_{nq}(\tau_k) \end{bmatrix}, \quad (24)$$

where

$$L_{11}(\tau_k) = \theta_1^{(11)} \tau_k^d + \theta_2^{(11)} \tau_k^{d-1} + \dots + \theta_d^{(11)} \tau_k + \theta_{d+1}^{(11)}$$

$$L_{1q}(\tau_k) = \theta_1^{(1q)} \tau_k^d + \theta_2^{(1q)} \tau_k^{d-1} + \dots + \theta_d^{(1q)} \tau_k + \theta_{d+1}^{(1q)}$$

$$L_{n1}(\tau_k) = \theta_1^{(n1)} \tau_k^d + \theta_2^{(n1)} \tau_k^{d-1} + \dots + \theta_d^{(n1)} \tau_k + \theta_{d+1}^{(n1)}$$

$$L_{nq}(\tau_k) = \theta_1^{(nq)} \tau_k^d + \theta_2^{(nq)} \tau_k^{d-1} + \dots + \theta_d^{(nq)} \tau_k + \theta_{d+1}^{(nq)}.$$

Then, on each execution of the actual controller, after calculating the next sampling interval τ_k via (9), each element of the observer gain matrix is computed through a different polynomial in the matrix $L_d(\tau_k)$.

H. Implementation guidelines

Through Algorithm 1 what was said in subsections III-F and III-G is summarized; this program can be performed offline by any numerical computing programming language. Algorithm 2 shows how to implement OSISTC on any processor with reduced performance features.

IV. RESULTS

An experiment on a real plant is presented in order to illustrate the theory introduced in the previous section.

A. Plant

The experimental plant with form (1) is the same electronic double integrator circuit as the one used in [7], so advise with that document for further information. The state space representation is

$$A_c = \begin{bmatrix} 0 & -23.81 \\ 0 & 0 \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 \\ -23.81 \end{bmatrix}, \quad C = [1 \quad 0]. \quad (25)$$

Data: $A_c, B_c, C, Q_c, R_c, \tau_{min}, \tau_{max}, \tau_g, \alpha, \beta, \eta, P_c$

Result: $K_c, K_d(\tau_k), L_d(\tau_k)$

$K_c = f(A_c, B_c, Q_c, R_c)$ by continuous LQR (7);

$T = f(\tau_{min}, \tau_{max}, \tau_g)$ by (14) and (16);

for $\tau_h \in T$ **do**

 Compute $A_d(\tau_h), B_d(\tau_h), Q_d(\tau_h), R_d(\tau_h)$ by (4);

$K_d(\tau_h) = f(A_d(\tau_h), B_d(\tau_h), Q_d(\tau_h), R_d(\tau_h))$ by discrete LQR (17);

 Compute $P_d(\tau_h)$ by (6);

$L_d(\tau_h) = f(A_d(\tau_h)^T, C^T, P_d(\tau_h))$ by Ackermann (21);

end

Create S_{K_d} based on all $K_d(\tau_h)$ by (18);

Create S_{L_d} based on all $L_d(\tau_h)$ by (22);

for $i \leq m$ **and** $j \leq n$ **do**

$K_{ij}(\tau_k) = f(S_{K_d}(i,j))$ by polynomial curve fitting;

end

for $i \leq n$ **and** $j \leq q$ **do**

$L_{ij}(\tau_k) = f(S_{L_d}(i,j))$ by polynomial curve fitting;

end

Create $K_d(\tau_k)$ by ordering all $K_{ij}(\tau_k)$ as in (20);

Create $L_d(\tau_k)$ by ordering all $L_{ij}(\tau_k)$ as in (24);

Algorithm 1: Offline design

Data: $A_c, B_c, C, K_c, \tau_{max}, \tau_g, \alpha, \beta, \eta, K_d(\tau_k), L_d(\tau_k)$

Result: $\tau_k, u^{(k)}$

Initialization of hardware, RTOS and variables;

$x^{(k)} := \text{read_input}()$;

$\tau_k := f(\tau_{max}, K_c, \alpha, \beta, \eta, \hat{x}^{(k)})$ by (9);

Set RTOS to trigger next time after τ_k ;

Calculate $K_d(\tau_k)$ by (20);

$u^{(k)} := f(K_d(\tau_k), \hat{x}^{(k)})$ by (10);

Set actuator with $u^{(k)}$;

Calculate $L_d(\tau_k)$ by (24);

Compute $A_d(\tau_k), B_d(\tau_k)$ by (4);

$\hat{x}^{(k)} := f(A_d(\tau_k), B_d(\tau_k), K_d(\tau_k), L_d(\tau_k), \hat{x}^{(k)}, x^{(k)})$ by (13)

Algorithm 2: Online implementation

In Table I most important configurations used to design both controller and observer are detailed. These values have been based on recommendations from the literature in [7]. Note that the poles of the observer have been chosen to be fast enough so that they do not slow down the dynamics of the plant

TABLE I
EXPERIMENT SETTINGS

Symbol	Description	Value
Q_c	States weighting matrix	$0.0025 \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
R_c	Control weighting matrix	0.1
τ_{min}	Minimum sampling time	20ms
τ_{max}	Maximum sampling time	60ms
τ_g	Sampling granularity	1ms
α	Density of the samples set	0.667
β	-	9.4
η	Density degree of the sampling seq.	0.11
P_c	Observer continuous-time poles	$[-50 + 2j, -50 - 2j]$

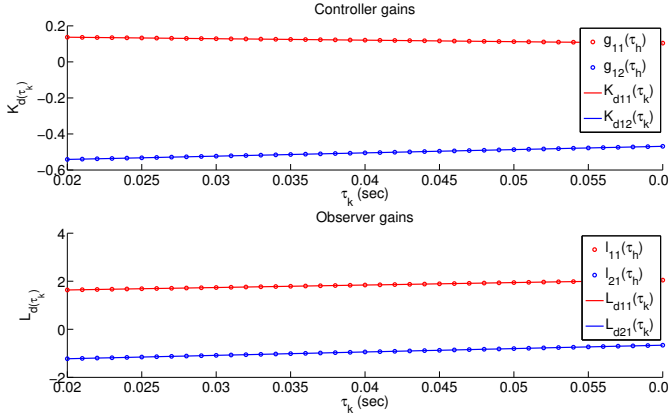


Fig. 4. Gains polynomial fitting: controller (top), and observer (bottom)

B. Controller and observer

Algorithm 1 has been followed step by step to perform the offline design. In Fig. 4 the gains of both controller and observer evaluated for the set of sampling intervals, are shown by circles. Likewise, fitted curves (continuous lines) roughly describe the behaviour of these gains. Additional numeric results are summarized next:

- Continuous-time feedback gain

$$K_c = [0.1581 \quad -0.5841].$$

- Controller gain matrix which consists of two polynomial as in

$$K_d(\tau_k) = [-0.8203\tau_k + 0.1533 \quad 1.819\tau_k - 0.5778].$$

- Observer gain matrix formed by two polynomials, as in

$$L_d(\tau_k) = [10.2521\tau_k + 1.431 \quad 14.094\tau_k - 1.5081]^T.$$

C. Implementation on a processor

The development platform comprises the digital signal controller (DSC) dsPIC33FJ256MC710A from Microchip which internally runs the Erika real-time kernel. To learn more about this environment, it is recommended to see the original work in [18] and its references, and the same implementation in [7].

The self-triggered controller uses rule (9) to calculate when it will activate itself next time; this value is used to set the RTOS to trigger the next sampling instant. Other functions of the controller are to read the states of the plant $x(k)$ through the DSC analog/digital converter, to estimate the states $\hat{x}(k)$ through the observer, and to compute the control action $u(k)$ which is applied directly to the plant via pulse width modulation (PWM).

Algorithm 2 has been used to perform the implementation that works on the microcontroller. To calculate the controller gain matrix, two first-degree polynomials that are functions of τ_k are represented as $K_{11}(\tau_k)$ and $K_{12}(\tau_k)$, grouped into $K_d(\tau_k)$. This is done instead of minimizing DARE. Finally, the observer gain matrix is replaced by a pair of first-degree polynomials $L_{11}(\tau_k)$ and $L_{21}(\tau_k)$, framed within $L_d(\tau_k)$. This is done instead of using a pole placement method i.e. Ackermann.

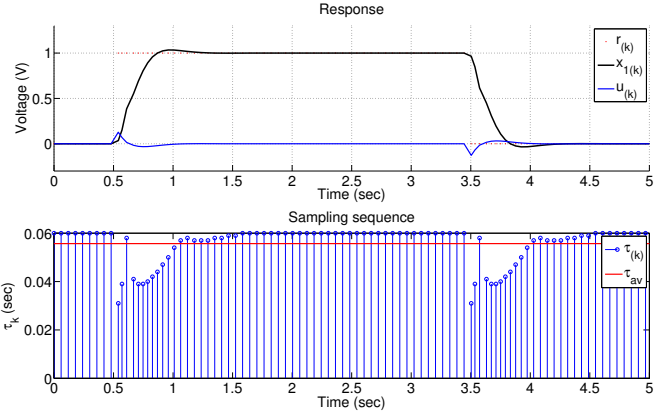


Fig. 5. Behavior of OSISTC in simulation

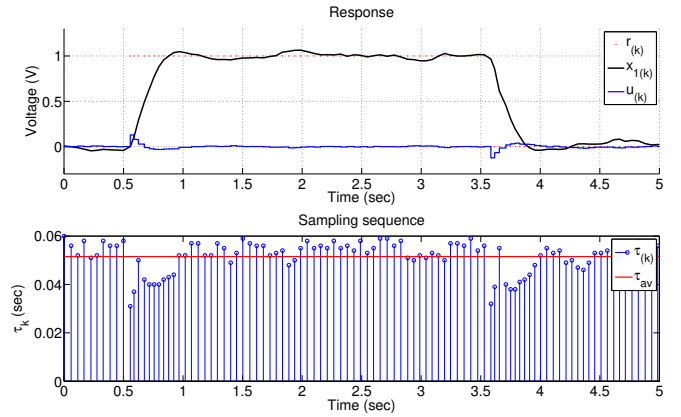


Fig. 6. Implementation of OSISTC with no observer [7]

V. DISCUSSION

Figures 5 to 7 show the states evolution and the sampling pattern both in simulation and actual implementations when OSISTC is subjected to follow a reference. The establishment times, overshoots, and steady-state errors are almost similar for all cases.

The sampling intervals in the simulation (Fig. 5) lie within the range [31; 60]ms, in the real system without observer (Fig. 6) are within [32; 59]ms, and in the real system with observer (Fig. 7) within [31; 60]ms. The red lines in the sampling sequence graphs correspond to the average sampling, explained later through equation (26).

The observer in Fig. 7 provides noise-free states that stabilize the triggering of sampling periods τ_k at the same time. The implementation without observer in Fig. 6 tends to shake in steady state since its states have noise, which causes the oscillation of the triggering of sampling periods.

The average sampling metric τ_{av} in [7] establishes

$$\tau_{av} = \frac{1}{N} \sum_{k=0}^{N-1} \tau_k, \quad (26)$$

where N is the number of samples within the experiment/simulation time; larger values of τ_{av} indicate less resource utilization. In the simulation $\tau_{avS} = 55.7$ ms, in the implementation without observer $\tau_{avNO} = 51.3$ ms, and in the

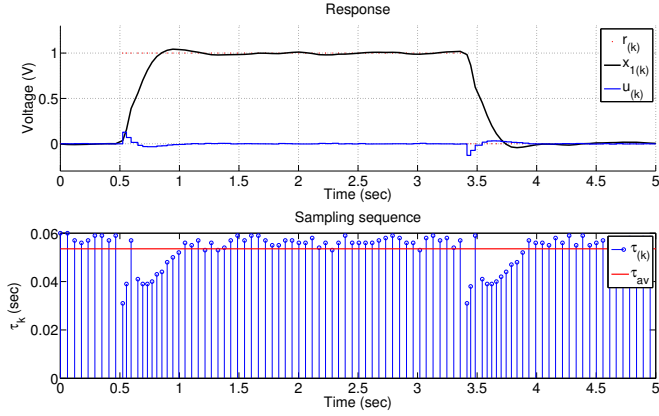


Fig. 7. Implementation of OSISTC with observer

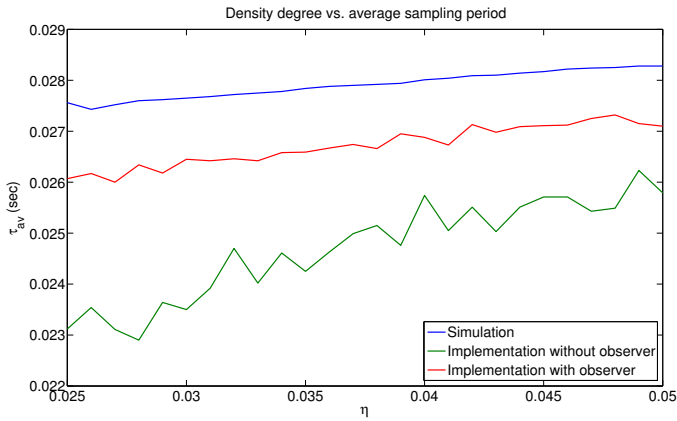


Fig. 8. Comparison of the average sampling period among simulations and implementations with and without observer, with $\tau_{min} = 3\text{ms}$, $\tau_{max} = 30\text{ms}$ and $\eta \in \{0.025, 0.026, \dots, 0.050\}$

implementation with observer $\tau_{avO} = 54.1\text{ms}$. The average sampling τ_{avNO} is less than τ_{avO} , which means that the implementation with observer has better performance than the implementation presented in [7] which has no observer, since it uses less processing resources.

Figure 8 shows how the sampling average periods behave when the density degree is changed as long as the guarantee in (16) is maintained. The behavior $\tau_{avO} > \tau_{avNO}$ is recurrent, which allows corroborating the results obtained above.

VI. CONCLUSIONS

Some techniques applied at the implementation stage to improve the performance of the method in [7] were presented. A polynomial fitted offline to calculate the discrete-time controller gains, was used to replace the online discrete-time LQR problem. A time-varying closed-loop observer has been implemented by polynomial fitting techniques while avoiding the online use of the Ackermann pole placement method.

Simulations and experiments have been confirmed the solution is effective and there could be an open research topic regarding observation techniques in OSISTC. There are interesting performance measures in the literature which could become future work for this study; metrics from [7] and [10]

would allow further evaluation on a real system. A comparison between the implementation with and without observer can be made to determine the true contribution of the latter.

ACKNOWLEDGMENT

This work has been partially supported by the University Center for Scientific and Technological Research (CUICYT) of Universidad Técnica del Norte.

REFERENCES

- [1] M. Velasco, J.M. Fuertes, and P. Martí, "The Self Triggered Task Model for Real-Time Control Systems," in *Proc. RTSS*, Cancun, Mexico, Dec., 2003, pp. 67-70.
- [2] A. Anta and P. Tabuada, "To Sample or Not to Sample: Self-Triggered Control for Nonlinear Systems," *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2030-2042, Sept. 2010.
- [3] M. Mazo Jr., A. Anta and P. Tabuada, "An ISS Self-Triggered Implementation of Linear Controllers," in *Automatica*, vol. 46, no. 8, pp. 1310-1314, Aug. 2010.
- [4] J. Almeida, C. Silvestre and A.M. Pascoal, "Self-Triggered Output Feedback Control of Linear Plants," in *Proc. ACC*, San Francisco, CA, USA, June-July, 2011, pp. 2831-2836.
- [5] A. Molin and S. Hirche, "On the Optimality of Certainty Equivalence for Event-Triggered Control Systems," in *IEEE Trans. Autom. Control*, vol. 58, no. 2, pp. 470-474, Feb. 2013.
- [6] E. Bini and G.M. Buttazzo, "The Optimal Sampling Pattern For Linear Control Systems," in *IEEE Trans. Autom. Control*, vol. 59, no. 1, pp. 78-90, Jan. 2014.
- [7] M. Velasco, P. Martí and E. Bini, "Optimal-Sampling-inspired Self-Triggered Control," in *Int. Conf. EBCSCP*, Krakow, Poland, June, 2015, pp. 1-8.
- [8] K.J. Åström and B. Wittenmark, *Computer-Controlled Systems: Theory and Design*, 3rd ed., Upper Saddle River, NJ, USA: Prentice Hall, 1997.
- [9] T. Gommans, D. Antunes, T. Donkers, P. Tabuada, and M. Heemels, "Self-Triggered Linear Quadratic Control", in *Automatica*, vol. 50, no. 4, pp. 1279-1287, Apr. 2014.
- [10] C. Rosero, C. Vaca, L. Tobar and F. Rosero, "Performance of Self-Triggered Control Approaches," in *Enfoque UTE*, vol. 8, no. 2, pp. 107-120, Mar. 2017.
- [11] J. Almeida, C. Silvestre and A.M. Pascoal, "Observer Based Self-Triggered Control of Linear Plants with Unknown Disturbances," in *Proc. ACC*, Montreal, Canada, June, 2012, pp. 5688-5693.
- [12] X. Wang and M.D. Lemmon, "Self-Triggering Under State-Independent Disturbances", in *IEEE Trans. Autom. Control*, vol. 55, no. 6, pp. 1494-1500, June 2010.
- [13] W.F. Arnold and A.J. Laub, "Generalized Eigenproblem Algorithms and Software for Algebraic Riccati Equations," in *Proc. IEEE*, vol. 72, no. 12, pp. 1746-1754, Dec. 1984.
- [14] D. Luenberger, "An Introduction to Observers," in *IEEE Trans. Autom. Control*, vol. 16, no. 6, pp. 596-602, Dec. 1971.
- [15] J. Ackermann, "On the Synthesis of Linear Control Systems with Specified Characteristics," in *Automatica*, vol. 13, no. 1, pp. 89-94, Jan. 1977.
- [16] F. Paez, R. Cayssials, J. Urriza, E. Ferro and J. Orozco, "Frequency Domain Analysis of a RTOS in Control Applications," in *Cong. CASE*, Buenos Aires, Argentina, Aug., 2016, pp. 21-26.
- [17] S.A. Dyer and X. He, "Least-squares fitting of data by polynomials," in *IEEE Instrum. Meas. Mag.*, vol. 4, no. 4, pp. 46-51, Dec. 2001.
- [18] C. Lozoya, P. Martí, M. Velasco, J. Fuertes and E. Martín, "Resource and Performance Trade-offs in Real-Time Embedded Control Systems," in *J. Real-Time Systems*, vol. 49, no. 3, pp. 267-307, May 2013.

Implementación de Controladores por Realimentación de Estados y Controlador PID aplicado a un Motor de DC con Dos Masas mediante dSPACE

State-space feedback controllers and PID controller implementation for a DC motor with two masses using dSPACE

Rene Alexander Palacios Ochoa

Abstract— This article presents the design and implementation of two state feedback controllers, and a proportional integral derivative controller PID is applied to a DC motor with a coupled mass on its axis and a second mass connected by a band. The variable to be controlled is the angular position of the DC motor. This article summarizes the design, implementation, and functional tests of three controllers, which have been implemented and tested on the same hardware but with different requirements for the angular position amplitude of the DC motor. The controllers' operation tests have been made on the physical motor with two masses and on the mathematical model of the electromechanical system, which allows a comparison of the controller running on the hardware and the simulated system. Each controller has been designed to obtain the shortest settling time for the step function response.

Index Terms—State variable feedback controller, DC motor, Proportional integral derivative controller.

Resumen—Este artículo presenta el diseño e implementación de dos controladores con realimentación de estados y un controlador proporcional integral derivativo PID aplicados sobre un motor de DC que tiene acoplado una masa a su eje y una segunda masa conectada mediante una banda. La variable a controlar es la posición angular del motor de DC. Este artículo resume el diseño, implementación y pruebas de funcionamiento de tres controladores implementados y probados en el mismo hardware, pero con requerimientos diferentes para la amplitud de la posición angular del motor de DC. Las pruebas de funcionamiento de los controladores han sido realizadas sobre el motor físico con dos masas y sobre el modelo matemático del sistema electromecánico, lo que permite tener una comparación del controlador al operar sobre el hardware y sobre el sistema simulado. Cada controlador ha sido diseñado para obtener el menor tiempo de establecimiento de la respuesta a la función escalón.

Palabras Claves—Controlador con realimentación de estados, motor de DC, Control proporcional integral derivativo.

I. INTRODUCCIÓN

LOS sistemas de control se aplican en varios campos de la Ingeniería como por ejemplo en ingeniería automotriz, aeroespacial, eléctrica, mecánica, hidráulica, etc. En este trabajo se aplica controladores a un sistema electromecánico que consiste en un motor de DC con dos masas, la primera masa está conectada al eje del motor y la segunda masa recibe el torque del motor a través de una banda.

Los sistemas de control en variables de estado caracterizan la configuración dinámica del sistema mediante la entrega de datos del estado actual de cada variable del sistema, el control mediante variables de estado se realiza a través de la realimentación de las variables de estado, por lo cual es necesario que estas variables sean medibles y disponibles para la realimentación de sus estados, de modo que aporten información del comportamiento dinámico del sistema.

En [1] presenta un controlador de velocidad para un motor de DC, el objetivo de [1] es presentar un procedimiento para realizar un control de velocidad mediante la plataforma dSPACE y Matlab; realiza una comparación entre los resultados simulados y los resultados medidos sobre el hardware. Del mismo modo que en [1] en este trabajo se utiliza una plataforma dSPACE y se realiza una comparación entre las pruebas realizadas sobre el sistema simulado y sobre el hardware. En este caso se implementa controladores para la posición angular del motor.

En [2] presenta un controlador de velocidad para un motor de DC, con realimentación de las variables de estados y observador de estados, el controlador incorpora un integrador. En el presente trabajo dos de los controladores aplicados sobre el motor de DC son implementados mediante técnicas de realimentación de estados, y uno de ellos incorpora un control proporcional integral, pero a diferencia de [2], se aplica control de posición en lugar de velocidad, y no se incluye observadores.

En [3] presenta un método de desarrollo rápido para

¹René Alexander Palacios Ochoa, Departamento de Formación Básica, Escuela Politécnica Nacional, Quito, Ecuador, alexander01pal@gmail.com

controlador un motor de DC sin escobillas, en este artículo presenta principalmente el beneficio en cuanto a reducción de tiempo de desarrollo al utilizar la plataforma dSPACE y su comunicación con Matlab. Al igual que en [3] en este trabajo se utiliza la herramienta de software en tiempo real ControlDesk y un módulo dSPACE para la verificación y pruebas de los controladores diseñados.

En [4] presenta un controlador proporcional integral derivativo PID adaptativo para un motor de DC, el controlador es implementado en un módulo dSPACE. A diferencia de [4], en este trabajo se ha implementado un controlador PID simple, calibrado por dos métodos experimentales indicados en las siguientes secciones.

En este artículo se resume la implementación de tres controladores aplicados a un sistema electromecánico. Los controladores fueron diseñados con diferentes requerimientos para la posición angular del motor, por lo cual la función escalón de referencia tiene diferente amplitud en cada experimento. En este trabajo se presenta los resultados obtenidos de tres controladores implementados en un mismo hardware como experimentos independientes. Sin embargo se realiza una comparación entre los controladores, pero al no tener la misma amplitud de entrada no se concluye cual tiene el mejor rendimiento, pues los controladores no operan bajo las mismas condiciones.

Se realiza el diseño e implementación de tres controladores, un PID y dos controladores en variables de estado, el primero es un controlador con realimentación de variables de estado "state variable feedback" SVF y el segundo un controlador SVF que incluye un control proporcional integral PI.

II. MODELAMIENTO DE LA PLANTA

Para diseñar los controladores se ha realizado previamente la identificación del sistema a través del uso de la herramienta System Identification Toolbox de Matlab, el cual permite construir el modelo matemático del sistema a través de las medidas de los datos de entrada y salida. El sistema contiene la parte mecánica y eléctrica. El sistema mecánico consta del eje del motor con dos masas, una en el lado del motor y la otra en el lado de la carga; las dos masas están conectadas entre sí mediante una banda que transfiere el torque del motor desde una masa a la otra. Se identificaron las inercias de cada masa, el coeficiente de amortiguamiento viscoso y la elasticidad de la banda. El sistema eléctrico corresponde al motor y al controlador del motor. Una vez realizada la identificación de la planta se la representa en variables de estado, y sobre este modelo matemático se basa los diseños de los controladores.

La Fig. 1 presenta los elementos de la parte mecánica del sistema, los cuales son las inercias de las dos masas, el coeficiente de rozamiento de cada inercia y el coeficiente de elasticidad de la banda que conecta las dos masas, en este modelo no se considera el rozamiento de la banda debido a que es mucho menor que el de las masas.

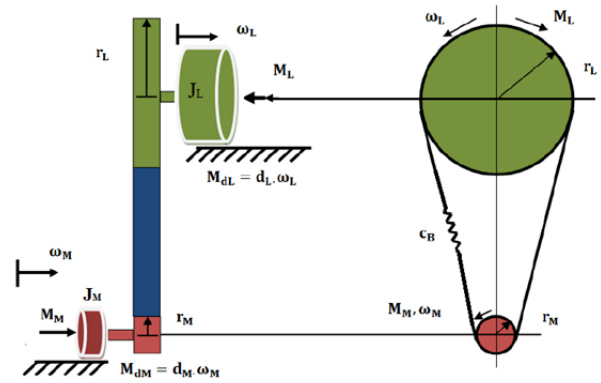


Fig. 1. Representación de las dos masas conectadas por una banda [5].

En la Tabla I, se presenta las abreviaturas de los parámetros del sistema que se utilizan en la modelación del sistema.

TABLA I
ABREVIATURAS DE LOS PARÁMETROS DEL SISTEMA [5]

Abreviatura	Significado
φ_M	Posición angular en el lado del motor.
φ_L	Posición angular en el lado de la carga.
ω_M	Velocidad angular en el lado del motor
ω_L	Velocidad angular en el lado de la carga
M_M	Torque actual del motor
M_L	Torque en la carga
M_{ref}	Torque de referencia
c_B	Coefficiente de elasticidad de la banda
d_M	Coefficiente de amortiguamiento en el lado del motor
d_L	Coefficiente de amortiguamiento en el lado de la carga
i	Relación de radios de las dos masas
J_M	Momento de inercia en el lado del motor
J_L	Momento de inercia en el lado de la carga
T_E	Constante eléctrica de tiempo
r_M	Radio de la masa en el lado del motor
r_L	Radio de la masa en el lado de la carga

La aceleración angular en el lado del motor está dada por (1) y en el lado de la carga por (2). Estas dos ecuaciones son deducidas de la suma de torques y pérdidas mecánicas [6] en el lado del motor y en lado de la carga.

$$\ddot{\omega}_M = -\frac{r_M^2 \cdot c_B}{J_M} \cdot \varphi_M - \frac{d_M}{J_M} \cdot \omega_M + \frac{i \cdot r_M^2 \cdot c_B}{J_M} \cdot \varphi_L + \frac{M_M}{J_M} \quad (1)$$

$$\ddot{\omega}_L = \frac{r_L^2 \cdot c_B}{J_L \cdot i} \cdot \varphi_M - \frac{r_L^2 \cdot c_B}{J_L} \cdot \varphi_L - \frac{d_L}{J_L} \cdot \omega_L - \frac{M_L}{J_L} \quad (2)$$

La relación de radios de las masas entre el lado de la carga y el lado del motor es:

$$i = \frac{r_L}{r_M} = 4 \quad (3)$$

La función de transferencia de la parte eléctrica está dada por:

$$G(s) = \frac{M_M}{M_{ref}} = \frac{k_E}{T_E \cdot s + 1} \quad (4)$$

De (4) se obtiene la ecuación que describe la relación entre

el torque actual y el torque de referencia:

$$\dot{M}_M = \frac{M_{ref}}{T_E} - \frac{M_M}{T_E} \quad (5)$$

Las velocidades angulares están dadas por:

$$\dot{\phi}_M = \omega_M \quad (6)$$

$$\dot{\phi}_L = \omega_L \quad (7)$$

Mediante (1), (2) y de (5) a (7) se tiene la descripción del sistema en variables de estado. En (8) se presenta la ecuación de estado y en (9) la ecuación de salida [7].

$$\dot{X}(t) = A \cdot X(t) + B \cdot u(t) \quad (8)$$

$$Y(t) = C \cdot X(t) + D \cdot u(t) \quad (9)$$

Donde:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\frac{r_M^2 \cdot c_B}{J_M} & -\frac{d_M}{J_M} & \frac{r_M^2 \cdot c_B \cdot i}{J_M} & 0 & \frac{1}{J_M} \\ 0 & 0 & 0 & 1 & 0 \\ \frac{r_L^2 \cdot c_B}{J_M \cdot i} & 0 & -\frac{r_L^2 \cdot c_B}{J_L} & -\frac{d_L}{J_L} & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{T_E} \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -\frac{1}{J_L} \\ \frac{1}{T_E} & 0 \end{bmatrix}, C = [1 \ 0 \ 0 \ 0 \ 0], D = [0 \ 0],$$

$$u(t) = \begin{bmatrix} M_{ref} \\ M_L \end{bmatrix}, X(t) = \begin{bmatrix} \phi_M \\ \omega_M \\ \phi_L \\ \omega_L \\ M_M \end{bmatrix}, \dot{X}(t) = \begin{bmatrix} \dot{\phi}_M \\ \dot{\omega}_M \\ \dot{\phi}_L \\ \dot{\omega}_L \\ \dot{M}_M \end{bmatrix}.$$

Las entradas del sistema son el torque de referencia y el torque en la carga y la salida es la posición angular en el lado del motor. La Fig. 2 presenta el sistema en diagrama de bloques que utiliza la descripción del sistema en el dominio de la frecuencia mediante el uso de la transformada de Laplace, lo cual se ha realizado para tener la visualización del sistema físico. Se utiliza torque para la entrada del sistema, el torque del motor es proporcional a la corriente de armadura del motor [8], en este caso por un factor de 0.32 [N/A].

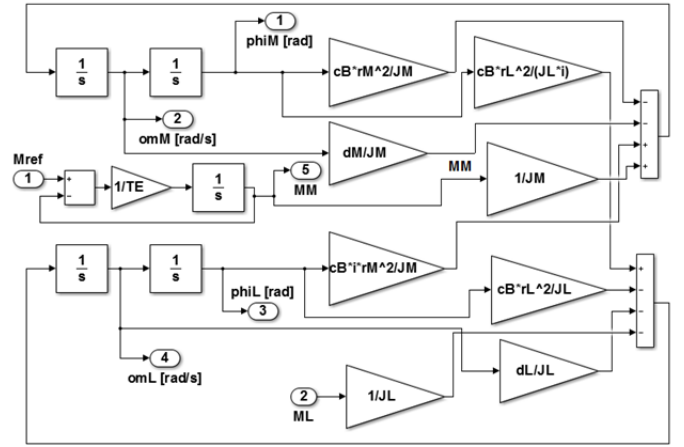


Fig. 2. Diagrama de bloques de la Planta del Motor [5].

III. CONTROLADOR POR REALIMENTACIÓN DE ESTADOS

La Fig. 3 presenta un diagrama simplificado de la realimentación en variables de estado. El modelo del sistema es representado por un solo bloque. Las ganancias son asignadas a todas las variables de estado, por lo que para implementar este controlador se tiene sensores que miden las cinco variables de estado. Para realizar el diseño del controlador por realimentación de estados, el sistema debe ser controlable y observable, lo cual se verifica mediante la condición de controlabilidad dada por (10) y de observabilidad dada por (11) [9].

$$|Q_c| = |b \quad Ab \quad A^2b \quad A^3b \quad \dots \quad A^{n-1}b| \neq 0 \quad (10)$$

$$|Q_o| = |c \quad A^T c \quad (A^T)^2 c \quad \dots \quad (A^T)^{n-1} c| \neq 0 \quad (11)$$

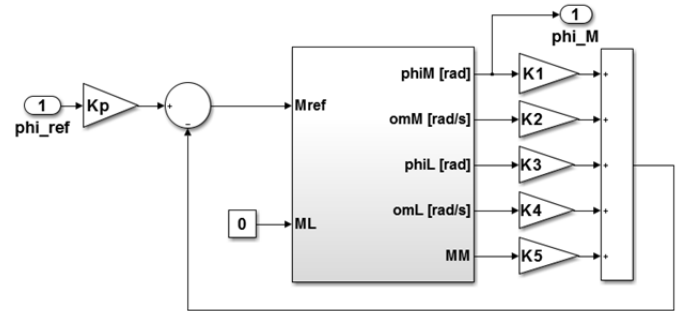


Fig. 3. Diagrama simplificado de realimentación de estados [5].

La variable de entrada es el torque de referencia dado por:

$$u(t) = M_{ref}(t) = \phi_{ref}(t) \cdot K_p - K^T \cdot X(s) \quad (12)$$

Al reemplazar (12) en (8) y (9) se tiene la función de transferencia de la planta con el controlador en variables de estado en lazo cerrado dada por (13). Con la función de transferencia se calcula los polos de la planta en lazo cerrado. Se encuentran las ganancias de la realimentación de estados mediante una comparación de los coeficientes del polinomio característico de la planta dada por (15) y la ecuación característica de la función de transferencia dada por (14).

$$G(s) = \frac{\varphi_M(s)}{\varphi_{ref}(s)} = K_p \cdot C^T \cdot (s \cdot I - A + B \cdot K^T)^{-1} \cdot B \quad (13)$$

$$\left| s \cdot I - A + B \cdot K^T \right| = 0 \quad (14)$$

$$(s - p_1) \cdot (s - p_2) \cdot (s - p_3) \cdot (s - p_4) \cdot (s - p_5) = 0 \quad (15)$$

El parámetro K_p indicado en la Fig. 3 es un factor necesario para tener una ganancia en estado estable igual a uno. El valor de K_p se lo calcula con (16).

$$K_p = \frac{1}{C^T \cdot (-A + B \cdot K^T)^{-1} \cdot B} \quad (16)$$

Una alternativa en lugar de realizar la comparación de coeficientes es el cálculo de las ganancias mediante la fórmula de Ackermann [10] dada por (17). La fórmula de Ackermann generalmente se utiliza en sistemas de mayor complejidad, mediante el uso de Matlab esta fórmula permite localizar los polos en las posiciones deseadas.

$$K^T = t^T \cdot \phi(A) \quad (17)$$

Donde: $\phi(A) = A^n + \gamma_{n-1}A^{n-1} + \dots + \gamma_1A + \gamma_0I$, y $t^T = [0 \ \dots \ 0 \ 1] \cdot [b \ Ab \ A^2b \ A^3b \ \dots \ A^{n-1}b]^{-1}$.

Los coeficientes γ_i son tales que p_1, p_2, \dots, p_n sean las raíces del polinomio.

$$(s - p_1) \cdot (s - p_2) \cdot \dots \cdot (s - p_n) = s^n + \gamma_1s^{n-1} + \gamma_2s^{n-2} + \dots + \gamma_{n-1}s + \gamma_n \quad (18)$$

La posición de los polos seleccionados es: $p = [-450+j700; -450-j700; -340; -380+j320; -380-j320]$. Con estos polos se tiene el vector ganancias: $K = [-66.0037; 0.0334; 294.4354; 0.0739; 0.9564]$ y la ganancia del factor K_p es 7.6052.

IV. CONTROLADOR POR REALIMENTACIÓN DE ESTADOS SVF CON CONTROLADOR PI INTEGRADO

La Fig. 4 presenta el diagrama simplificado del controlador SVF con PI incluido, este tipo de controlador es un sistema servo [11] que incluye un integrador y un control proporcional. Este controlador tiene un control PI que tiene como entrada la diferencia entre la posición angular del motor y un valor de referencia; la salida del control PI es comparada con la suma de todas las ganancias de realimentación de estados, cuyo resultado es la referencia de torque del sistema.

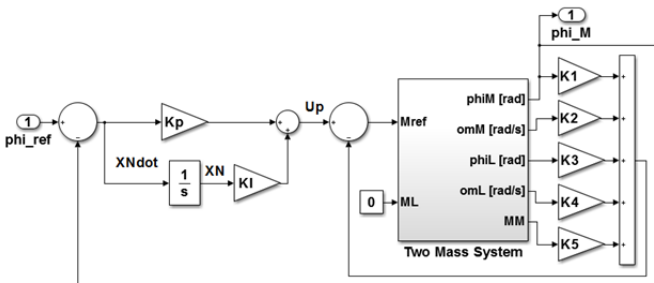


Fig. 4. Diagrama simplificado del controlador SVF con PI [5].

Este controlador añade una nueva variable de estado al sistema a causa del integrador del PI, esta variable de estado X_N está dada por (19).

$$X_N = \frac{1}{s} \cdot [\varphi_{ref}(s) - C^T \cdot X(s)] \quad (19)$$

Al transformar (19) al dominio del tiempo se obtiene:

$$\dot{X}_N = \varphi_{ref}(t) - C^T \cdot X(t) \quad (20)$$

La salida del controlador PI está dada por:

$$U_p = K_I \cdot X_N + K_p \cdot (\varphi_{ref} - \varphi_M) \quad (21)$$

$$K_I = \frac{K_p}{T_I} \quad (22)$$

Al sustituir de (19) a (21) en (8), se tiene la descripción de la dinámica del sistema en variables de estado que incluye la nueva variable X_N .

$$\begin{bmatrix} \dot{X}(t) \\ \dot{X}_N(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ C^T & 0 \end{bmatrix} \begin{bmatrix} X(t) \\ X_N(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} M_{ref}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \varphi_{ref}(t) \quad (23)$$

Se definen las matrices extendidas:

$$A_e = \begin{bmatrix} A & 0 \\ C^T & 0 \end{bmatrix}, B_e = \begin{bmatrix} B \\ 0 \end{bmatrix} \quad (24)$$

En la Fig. 4 se observa que el torque de referencia es función de las ganancias de realimentación de estados, de U_p y de las variables de estado.

$$M_{ref} = U_p - K^T \cdot X \quad (25)$$

Al reemplazar (21) en (25) se obtiene:

$$M_{ref} = K_p \cdot \varphi_{ref} - [K_p \cdot C^T + K^T - K_I] \cdot \begin{bmatrix} X \\ X_N \end{bmatrix} \quad (25.a)$$

$$K_e^T = [K_p \cdot C^T + K^T - K_I] \quad (26)$$

Donde el vector de ganancias K aparece en (26). Mediante el uso de la fórmula de Ackermann se calcula las ganancias a través de las matrices extendidas y la ubicación deseada de los seis polos, el sexto polo se forma debido a X_N . El resultado de la fórmula de Ackermann está dado por (27), el cual debe ser comparado con (26) para obtener las ganancias de realimentación de estado y el coeficiente de integración K_I .

$$K_e^T = [K_{e1} \ K_{e2} \ K_{e3} \ K_{e4} \ K_{e5} \ K_{e6}] \quad (27)$$

Con lo cual se obtiene: $K_I = -K_{e6}$, $K_p = K_I \cdot T_I$, $K^T = K_p \cdot C^T - K_e^T$. La ubicación de los polos de este sistema es: $p = [-50+j1000; -50-j1000; -90+j47; -90-j47; -98; -67.5]$.

De la fórmula de Ackermann se obtiene el vector de ganancias extendidas: $K_e^T = [-10.18 \ -0.014 \ 42.24 \ 0.08 \ -0.597 \ -9.09]$. El vector de ganancias K^T , y las ganancias K_I y K_p son: $K^T = [-10.18 \ -0.014 \ 42.24 \ 0.08 \ -0.597]$, $K_I = 9.0926$, $K_p = 0.1346$.

Al tener un polo más este sistema se reduce la velocidad del controlador, sin embargo se ha elegido la ubicación de los polos después de varias pruebas y un mejor resultado en el sobre impulso y error en estado estable de la respuesta a la función escalón.

Los resultados del controlador SVF simple y del controlador SVF con PI integrado son comparados con el controlador PID para tener un punto de referencia del comportamiento de los controladores con realimentación de estado. En la siguiente sección se presenta el diseño del controlador PID.

V. CONTROLADOR PROPORCIONAL INTEGRAL DERIVATIVO

El controlador PID se calibra directamente sobre el motor por medio del método de Ziegler-Nichols, el controlador es presentado en su forma aditiva en la Fig. 5. Para la parte derivativa del controlador PID se utiliza un DT1, el cual contiene un cero y un polo con una constante de tiempo pequeña, esto con el fin de que sea posible su implementación real. Adicional al método de Ziegler-Nichols, también se realiza un ajuste experimental del PID, al variar los parámetros de ganancia y los tiempos de integración sin exceder el 15% del valor de sobre impulso.

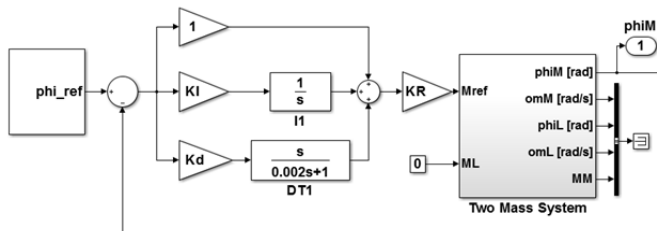


Fig. 5. Controlador PID [5].

Para ajustar el PID por medio de Ziegler-Nichols, se incrementa el valor de ganancia K_R desde un pequeño valor hasta que el sistema alcance su límite de estabilidad, en el cual se mide la ganancia crítica $K_{R-critico}$ y el periodo de oscilación $T_{critico}$. Este método permite encontrar en forma experimental las constantes de tiempo del integrador T_I y del derivador T_d al utilizar la ganancia crítica y el periodo de oscilación, según los valores indicados en la Tabla II. Se encontró los valores de ganancia crítica y periodo de oscilación crítico: $K_{R-critico} = 2$ y $T_{critico} = 0.0658$.

TABLA II

PARÁMETROS DEL PID MEDIANTE ZIEGLER-NICHOLS [5],[9]

Parámetro	Valor	Valor
Ganancia del controlador K_R	$0.45 K_{R-critico}$	0.9
Constante de tiempo del integrador T_I	$0.5 T_{critico}$	0.0329
Constante de tiempo del derivador T_d	$0.125 T_{critico}$	0.008225

VI. PROCEDIMIENTO E IMPLEMENTACIÓN

Los controladores son simulados en Matlab y luego cargados en un módulo controlador dSPACE. A través del Interfaz en tiempo real RTI se pueden configurar las entradas y salidas del controlador gráficamente en Simulink [1]. La plataforma dSPACE principalmente se utilizan para realizar

pruebas en prototipos.

En este trabajo se ha implementado dos controladores SVF y un PID y comparado su desempeño en cuanto a tiempo de establecimiento, sobre impulso y error en estado estable. En este trabajo se realizan las pruebas sobre: 1) El motor físico y 2) La planta o modelo matemático del sistema a controlar. Se ha diseñado e implementado los controladores: PID, control por realimentación de estados SVF simple y control por realimentación de estados con control proporcional integral en la realimentación SVF-PI.

VII. PRUEBAS REALIZADAS Y ANÁLISIS DE RESULTADOS

Se realizan pruebas de funcionamiento de los tres controladores: 1) Controlador PID ajustado por Ziegler-Nichols y PID ajustado experimentalmente por prueba y error para limitar el sobre impulso, 2) Controlador SVF simple y 3) Controlador SVF con control PI integrado. Para los tres controladores se aplica una función paso y se mide el tiempo de establecimiento, el error en estado estable y el sobre impulso o sobrecarga. Las tres medidas son realizadas en el lado del motor. Para cada uno de los controladores se presentan dos figuras: posición angular y torque actual, tanto para el motor como para la planta.

Posición angular: Contiene la gráfica de tres señales de la posición angular, la línea verde presenta la señal de referencia dada por la función paso, la línea roja corresponde al modelo matemático y la línea azul corresponde al motor físico.

Torque actual: Contiene la gráfica de dos señales, la línea roja es el torque actual del modelo matemático y la línea azul es el torque actual del motor físico.

Las gráficas son realizadas mediante el software ControlDesk de la plataforma dSPACE.

A. Prueba de Funcionamiento del Controlador PID

El controlador PID es ajustado 1) a través de la técnica de Ziegler-Nichols y 2) en forma experimental al variar sus parámetros mediante prueba y error. La Fig. 6 presenta el controlador PID sobre la planta y sobre el motor de DC. La función paso se aplica tanto al modelo matemático como al motor físico con el mismo nodo de conexión.

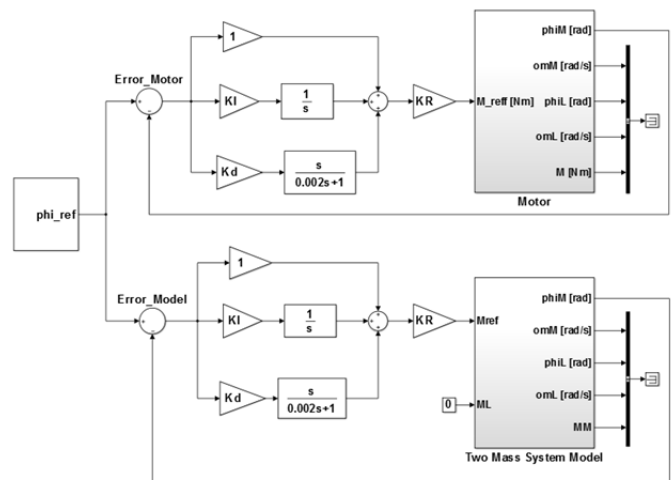


Fig. 6. Controlador PID aplicado al Motor y a la Planta [5].

El bloque superior de la Fig. 6 enmascara las señales de entrada y salida, correspondientes a las variables de estado del sistema medidas en el motor; mientras que el bloque inferior de la Fig. 6 presenta al controlador PID aplicado sobre la planta del motor con dos masas. La única variable de estado que se utilizan en el control PID es la posición angular de salida en el lado del motor.

La Fig. 7 presenta la respuesta a la función paso del PID sintonizado con el método de Ziegler-Nichols.

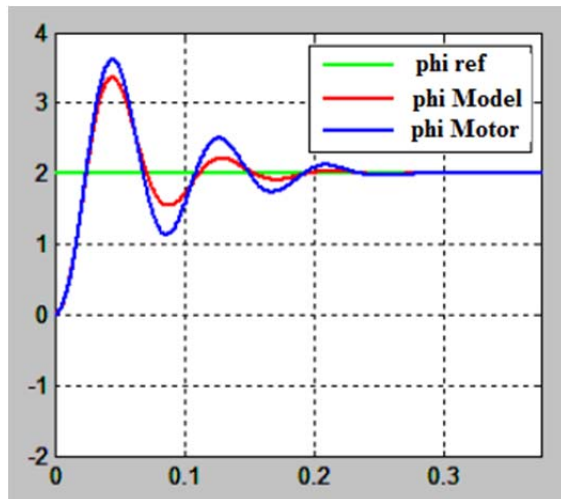


Fig. 7. Respuesta a la función paso con PID Ziegler-Nichols [5].

La Tabla III contiene los resultados de la respuesta a la función paso para el motor y para la planta cuando se utiliza el controlador PID calibrado por Ziegler-Nichols. Debido a que es un controlador PID, la parte integral del controlador mantiene el error en estado estable en un valor prácticamente igual a cero pero el sobre impulso es bastante alto en el motor y la planta. El tiempo de establecimiento es de 0.29 segundos para el motor físico y 0.33 segundos para la planta simulada.

TABLAIII
RESULTADOS DEL CONTROLADOR PID POR ZIEGLER-NICHOLS [5].

Parámetro	Motor	Planta
Tiempo de establecimiento en segundos	0.29	0.33
Error en estado estable	0.013%	0.001%
Sobre impulso	80.81%	68.55%

La Fig. 8 presenta el torque actual en el motor y en la planta. El torque en el motor tiene un valor más alto que el torque de la planta, pero no puede exceder 1.6 [N], porque el motor dispone de un limitador de corriente para protección, lo cual limita la magnitud de torque aplicado.

Se calibra el controlador PID de la Fig. 6 en forma experimental mediante la variación de sus parámetros y al considerar no exceder el sobre impulso de 15%. Los resultados de esta calibración experimental en base a prueba y error se presentan en la Tabla IV. Se obtiene un sobre impulso 13.97% para el motor, lo cual es una reducción muy alta respecto al valor de 80.81% que se obtiene con Ziegler-Nichols. Sin embargo en este caso el tiempo de establecimiento es mucho más alto con un valor de 0.75 segundos lo cual es más del doble del valor obtenido con

Ziegler-Nichols. El error en estado estable es bajo en ambos casos debido al integrador del controlador. Los datos de la Tabla IV se visualizan en la Fig. 9.

Los datos de la Tabla IV para el motor y la planta son bastante similares en cuanto a tiempo de establecimiento y sobre impulso.

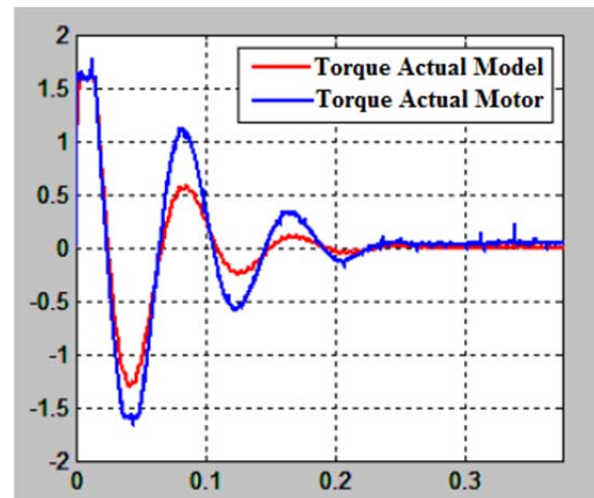


Fig. 8. Torque actual sobre la planta y el Motor [5].

TABLAIV
RESULTADOS DEL CONTROLADOR PID EXPERIMENTAL [5].

Parámetro	Motor	Planta
Tiempo de establecimiento en segundos	0.75	0.73
Error en estado estable	0.047%	0.015%
Sobre impulso	13.97%	12.81%

La Fig. 9 presenta la respuesta a la función paso del sistema con controlador PID ajustado experimentalmente, en la cual se observa una reducción significativa del sobre impulso y aumento del tiempo de establecimiento respecto al método de Ziegler-Nichols. La respuesta a la función paso es bastante similar para la planta y el motor.

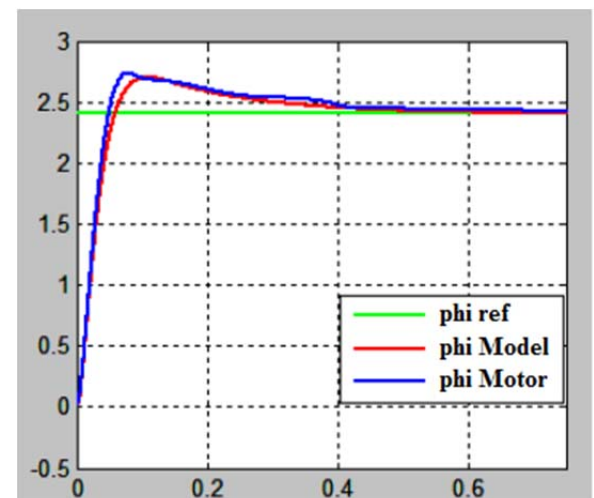


Fig. 9. Respuesta a la función paso con ajuste experimental del PID [5].

La Fig. 10 presenta el torque actual del motor y de la planta con el ajuste experimental del PID, como se observa en la

Fig. 10 el torque decrece sin oscilaciones del mismo modo que la respuesta a la función paso, la cual también tiene un sobre impulso bajo y sin oscilaciones. El torque actual en la planta y en el motor es bastante similar.

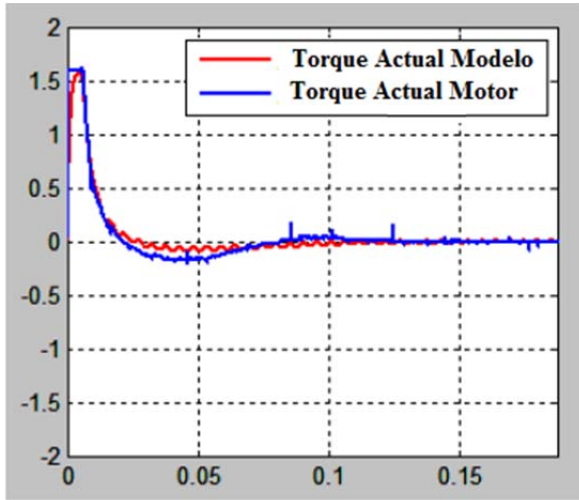


Fig. 10. Torque actual con ajuste experimental del PID [5].

B. Prueba de Funcionamiento del Controlador SVF Simple

El controlador con realimentación de estados SVF se presenta en la Fig. 11. De igual modo que en el caso del PID se realizan las pruebas sobre la planta y sobre el motor de DC con dos masas. El bloque superior de la Fig. 11 enmascara las señales de salida y de entrada, correspondientes a las variables de estado medidas en el motor, mientras que el bloque inferior de la Fig. 11 contiene el modelo matemático del motor. Las magnitudes de las ganancias de realimentación de estados son las mismas para la planta y para el motor.

La Tabla V presenta los resultados de la respuesta a la función paso con el controlador SVF. El sobre impulso con este controlador y el tiempo de establecimiento es mucho menor que con el controlador PID.

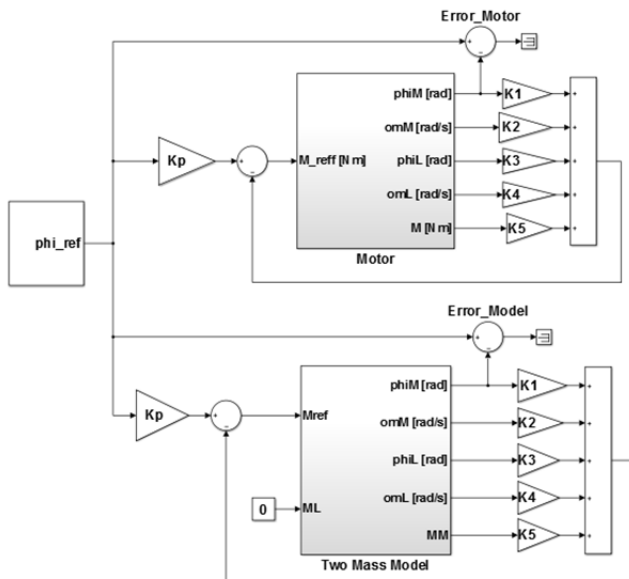


Fig. 11. Controlador SVF aplicado al Motor y a la Planta [5].

TABLAV
RESULTADOS DEL CONTROLADOR SVF SIMPLE [5].

Parámetro	Motor	Planta
Tiempo de establecimiento en segundos	0.063	0.035
Error en estado estable	0.028%	0%
Sobre impulso	8.73%	0.1%

La Fig. 12 presenta la respuesta a la función paso por medio del controlador SVF sobre el motor y sobre la planta. En el modelo matemático no se tiene sobre impulso pero al aplicar la misma función paso sobre el motor real, se tiene un sobre impulso del 8.73%.

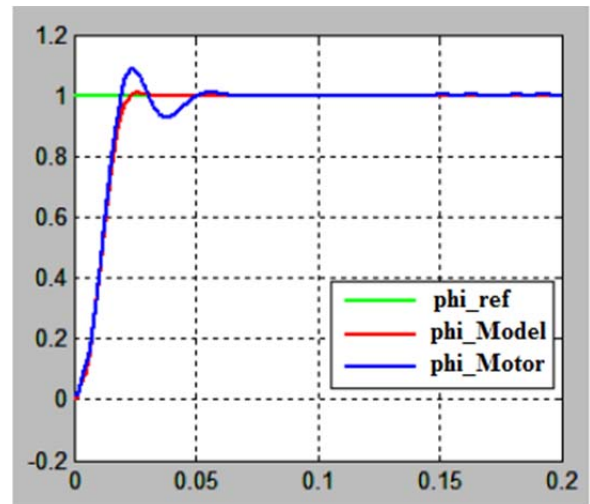


Fig. 12. Respuesta a la función paso con Controlador SVF [5].

La Fig. 13 presenta el torque actual sobre la planta y el motor con un controlador SVF. El torque del motor es limitado por el elemento de protección que satura la corriente que alimenta al motor, para no exceder el torque máximo de 1.6 [N].

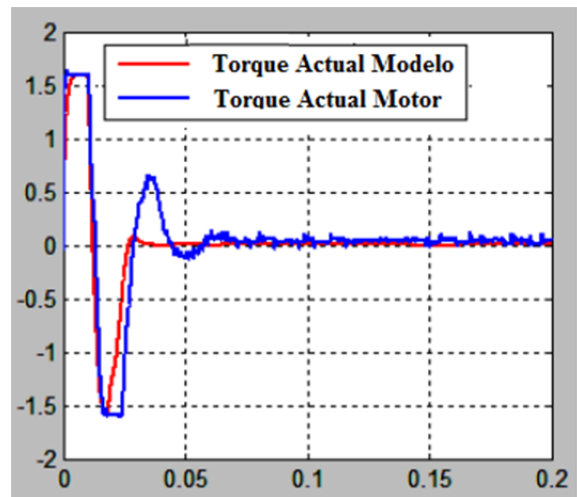


Fig. 13. Torque actual del sistema con controlador SVF [5].

C. Prueba de Funcionamiento del Controlador SVF con controlador PI integrado

El controlador con realimentación de estados SVF y controlador PI integrado se presenta en la Fig. 14. Se realizan

las pruebas sobre la planta y sobre el motor de DC con dos masas. El bloque superior de la Fig. 14 enmascara las señales de salida y de entrada, correspondientes a las variables de estado medidas en el motor; mientras que el bloque inferior de la Fig. 14 contiene el modelo matemático del motor. Los valores de las ganancias de realimentación de estados son las mismas para la planta y para el motor. El controlador PI recibe la señal de error entre la referencia del motor y la posición angular del motor. El torque de referencia para el motor se calcula mediante la diferencia entre la salida del PI y las variables de estado multiplicadas por sus respectivas ganancias de realimentación, como se indica en la Fig. 14.

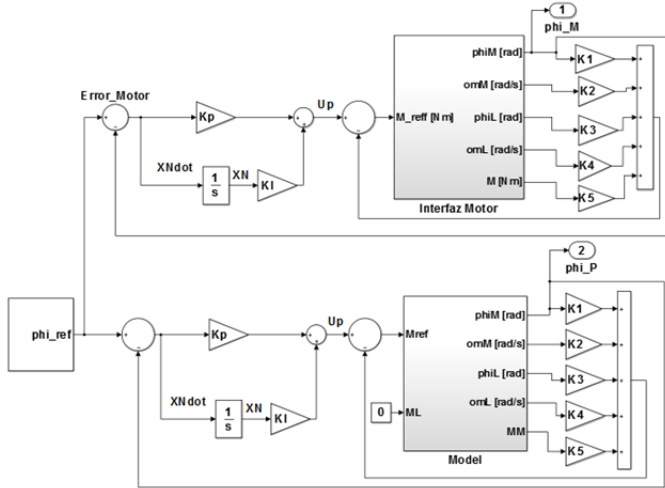


Fig. 14. Controlador SVF-PI aplicado al Motor y a la Planta [5].

La Tabla VI presenta los resultados de la respuesta a la función paso para el motor y para la planta con su respectivo controlador SVF-PI. El sobre impulso con este controlador para el motor es de 6.97% y el tiempo de establecimiento con este controlador es 0.08 segundos, lo cual es mucho menor que con el controlador PID.

TABLAVI
RESULTADOS DEL CONTROLADOR SVF CON CONTROLADOR PI [5]

Parámetro	Motor	Planta
Tiempo de establecimiento en segundos	0.080	0.075
Error en estado estable	0.003%	0%
Sobre impulso	6.97%	0%

La Fig. 15 presenta la respuesta a la función paso al aplicar el controlador SVF-PI sobre el motor y sobre la planta. En la planta no existe sobre impulso pero al aplicar la función paso sobre motor físico, se tiene un sobre impulso del 6.97%. Con el controlador SVF-PI el error en estado estable es prácticamente cero, esto debido al integrador en lazo cerrado que elimina el error. El tiempo de establecimiento es de 0.08 segundos, lo cual es ligeramente más lento que el SVF simple que tiene un tiempo de establecimiento de 0.063 segundos.

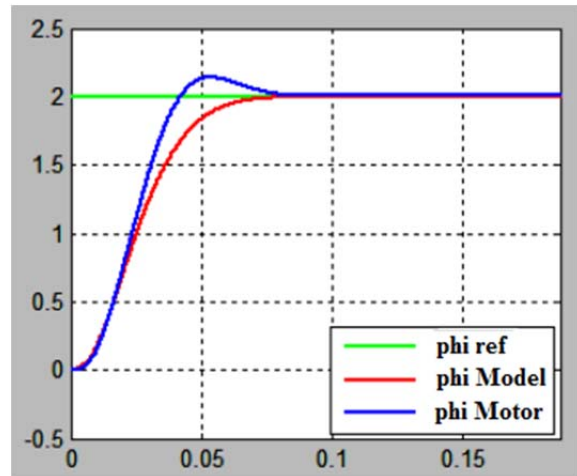


Fig. 15. Respuesta a la función paso con Controlador SVF-PI [5].

La Fig. 16 presenta el torque actual sobre la planta y el motor con un controlador SVF-PI. Para esta magnitud de torque en el motor no se activa la saturación por torque, debido a que el valor del torque actual es inferior al máximo torque del motor. El torque pico en la Fig. 16 es 1.4 [N], mientras que el torque máximo es 1.6 [N].

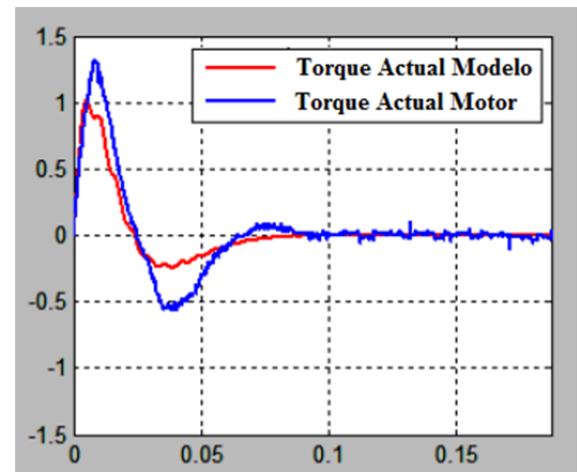


Fig. 16. Torque actual del sistema con controlador SVF-PI [5].

D. Comparación de Resultados

Los controladores implementados han sido realizados independientemente, con requerimientos diferentes de la amplitud de la posición angular del motor, son experimentos separados. Por lo tanto la comparación de resultados solo indica una aproximación de la tendencia en los resultados. No se puede concluir que se obtendrá los mismos resultados para condiciones iguales en la señal de referencia. Para una comparación que defina el mejor rendimiento entre estos controladores se debe realizar pruebas adicionales con la misma referencia.

La Tabla VII y Tabla VIII agrupan los resultados obtenidos con los tres controladores PID, SVF y SVF-PI; en estas tablas los títulos PID 1 corresponde a los resultados con el controlador PID ajustado por Ziegler-Nichols y PID 2 corresponde a los resultados con el PID ajustado experimentalmente mediante prueba y error.

La Tabla VII presenta los resultados para las pruebas sobre el motor físico, en la cual se observa que el menor tiempo de establecimiento se logra con el controlador SVF simple con un valor de 0.063 segundos, seguido por el controlador SVF-PI con un valor de 0.08 segundos. El PID tiene un tiempo de establecimiento mucho más lento, cuyo valor es de 0.29 segundos con la calibración por Ziegler-Nichols y de 0.75 segundos para la calibración mediante prueba y error.

El menor error en estado estable se obtiene con el controlador SFV- PI y después con el controlador PID. Tanto el PID como el SVF-PI tienen menor error en estado estable que el SVF, debido a que disponen de un integrador en el lazo de realimentación.

El menor sobre impulso se obtiene con el SVF-PI con un valor de 6.97% y segundo con el controlador SVF simple con un valor de 8.73%. Al utilizar la técnica de calibración de Ziegler-Nichols, el PID tiene un sobre impulso muy alto de 80.81%, sin embargo con la calibración experimental del PID mediante prueba y error, se obtuvo mejores resultados con un valor de 13.97%; pero el tiempo de establecimiento aumentó a 0.75 segundos.

TABLAVII
RESULTADOS DE LOS CONTROLADORES SOBRE EL MOTOR [5]

Parámetro	PID 1	PID 2	SVF	SVF -PI
Tiempo de establecimiento[s]	0.29	0.75	0.063	0.080
Error en estado estable	0.013%	0.047%	0.028%	0.003%
Sobre impulso	80.81%	13.97%	8.73%	6.97%

La Tabla VIII presenta los resultados de las pruebas sobre la planta o modelo matemático del motor.

TABLAVIII
RESULTADOS DE LOS CONTROLADORES SOBRE LA PLANTA [5]

Parámetro	PID 1	PID 2	SVF	SVF -PI
Tiempo de establecimiento [s]	0.33	0.73	0.035	0.075
Error en estado estable	0.001%	0.015	0%	0%
Sobre impulso	68.55%	12.81	0.1%	0%

Al igual que con el motor, el menor tiempo de establecimiento se logra con el controlador SVF, el menor sobre impulso con el SVF-PI y el menor error en estado estable es cero con los controladores SVF y cercano a cero con el PID 1. Sin embargo la planta es una aproximación matemática del motor físico por lo que existe cierta diferencia en los resultados especialmente en el de sobre impulso.

VIII. CONCLUSIONES

Los resultados indican que los controladores con realimentación de variables de estado SVF tienen un tiempo de establecimiento mucho menor que los controladores PID. El sobre impulso de los controladores SVF es significativamente menor que el obtenido con los controladores PID. Los controladores SVF y SVF-PI solo pueden implementarse si se dispone de la realimentación de todas las variables de estado, si se requiere realizar un controlador SVF y no se dispone de todas las variables de estado, existe la alternativa de implementar un controlador con

observadores de estado para las variables que no se pueden medir. Se ha obtenido mejores resultados en sobre impulso con el método experimental de prueba y error que con el método de Ziegler-Nichols, debido a que el método de Ziegler-Nichols se basa en el estudio experimental de múltiples respuestas de sistemas para definir las ganancias del PID, mientras que la calibración experimental del PID mediante prueba y error se ha realizado específicamente para este sistema electromecánico.

RECONOCIMIENTOS

A la Universidad de Ciencias Aplicadas Esslingen de Alemania, por la prestación tecnológica para realizar este proyecto.

REFERENCIAS

- [1] K. Meah, S. Hietpas, y S. Ula, "Rapid Control Prototyping of a Permanent Magnet DC Motor Drive System using dSPACE and Mathworks Simulink," in APEC 07 - Twenty-Second Annual IEEE Applied Power Electronics Conference and Exposition, 2007, Anaheim, CA, USA, Feb. 2007, pp. 856-861.
- [2] U. Manwong, S. Boonpiyathud, y S. Tunyasirut, "Implementation of a dSPACE DSP-based state feedback with state observer using Matlab/Simulink for a speed control of DC motor system," in 2008 International Conference on Control, Automation and Systems, Seoul, South Korea, Oct. 2008, pp. 2433-2436.
- [3] Y. Shu, H. Li, L. Xu, y Q. Wu, "A Rapid Development Method on Brushless DC motor Controller," in 2007 IEEE International Symposium on Industrial Electronics, Vigo, Spain, Jun. 2007, pp. 3293-3297.
- [4] T. He y L. Peng, "Application of Neuron Adaptive PID on DSPACE in Double Loop DC Motor Control System," in 2010 International Conference on Computing, Control and Industrial Engineering, Wuhan, China, China, Jun. 2010, vol. 2, pp. 257-260.
- [5] A. Palacios, "Modeling, Identification and Controller Design of an Electrical Servo System with a Two-Mass Characteristic," M.S. thesis, University of Applied Science Esslingen, Germany, 2010.
- [6] S. E. Lyshevski, Electromechanical Systems and Devices. CRC Press, 2008.
- [7] R. L. W. II y D. A. Lawrence, Linear State-Space Control Systems. John Wiley & Sons, 2007.
- [8] A. Kremser, Elektrische Maschinen und Antriebe: Grundlagen, Motoren und Anwendungen. Springer-Verlag, 2013.
- [9] H. P. Geering, Regelungstechnik: Mathematische Grundlagen, Entwurfsmethoden, Beispiele. Springer-Verlag, 2013.
- [10] S. M. Shinnars, Modern Control System Theory and Design. John Wiley & Sons, 1998.
- [11] K. Ogata, Modern Control Engineering. Prentice Hall, 2010.

Desarrollo de un algoritmo en MATLAB para la optimización de la resolución de una tarjeta USRP B210 para aplicaciones SDRadar

Develop a MATLAB algorithm for the resolution optimization of a USRP B210 card for SDRadar applications

David Moreno, Julio Mejía y Hugo Moreno

Abstract— This paper analyzes the universal software radio peripheral (USRP) B210 limitations and characteristics to develop applications on Software Defined Radar (SDRadar). The developed algorithm uses a frequency-modulated constant wave (FMCW), which implements a chirp signal, considering that 25 MHz is the maximum bandwidth that the USRP can provide with a maximum resolution of 6 meters. The method improves the resolution through several signals transmitted with different bandwidths, obtaining different resolutions, storing them in a simple matrix, and analyzing them. After simulations, it is determined that the 14 measurements done with bandwidth spacing of 0.5 MHz between every signal is the best way to improve the resolution. Finally, two scenarios for the optimization procedure are described. The first scenario is limited by a measurement error of less than 1 meter, and the second scenario is limited by a measurement error between 1 and 2 meters, getting better effectiveness of the measurement under the first scenario with 69.15% of detected objectives compared with the 30.85% of the effectiveness of the second one. This paper demonstrates that the algorithm is practical in SDRadar applications to detect objectives for topographic applications or SAR systems.

Index Terms—USRP B210, algorithm, SDRadar, FMCW, resolution

Resumen—En este documento se analizan las limitaciones y características que tiene la tarjeta USRP (Universal Software Radio Peripheral) B210 para desarrollar aplicaciones de radar definidos por software (SDRadar). Para el desarrollo del algoritmo se utilizó un radar de onda continua modulada en frecuencia (FMCW) que implementa una señal chirp, al considerar que 25 MHz es el máximo ancho de banda que la tarjeta puede proporcionar se obtiene como resultado una

resolución de 6 metros. El método utilizado para optimizar la resolución de la tarjeta fue transmitir un determinado número de señales con distintos anchos de banda, lo que resulta en una resolución diferente, al detectar y almacenar el objetivo en una matriz. Después de las pruebas correspondientes el algoritmo estableció que con 14 mediciones con variaciones de ancho de banda de 0.5 MHz, entre cada señal, se obtiene la mayor optimización para mejorar la resolución de la tarjeta. Se obtuvo finalmente dos zonas de optimización, donde la primera zona está limitada por un error de medición menor a 1 metro, y en la segunda zona con un error de medición entre 1 y 2 metros con 69.15% y 30.85% de objetivos detectados respectivamente, lo que determina su efectividad y confiabilidad. Este trabajo ha demostrado que el algoritmo utilizado es útil en aplicaciones SDRadar para detectar objetivos para aplicaciones topográficas o sistemas SAR.

Palabras Claves—USRP B210, algoritmo, SDRadar, FMCW, resolución

I. INTRODUCCIÓN

LOS sistemas radares fueron diseñados para ambientes Lbólicos [1] y hasta la actualidad se han aplicado en diferentes áreas como: la meteorología, control de velocidad marítima/aérea, radares doppler, entre otros, y han complementado varios estudios relacionados con necesidades presentes en la sociedad [2], [3], que requieren de sistemas tecnológicos confiables.

La tecnología Software Defined Radio (SDR) se presenta como una alternativa para diseñar prototipos a bajo coste [4], debido a que es confiable y adaptable a diferentes necesidades. En otras investigaciones se han analizado las capacidades de aplicar tecnología SDR para sistemas radares, propuestas que han sido aceptadas [5], así mismo se han realizado investigaciones implementadas con FPGA (Field Programmable Gate Array) [6], [7] sin embargo, presentan costos son elevados.

Uno de los inconvenientes de la tecnología SDR, es el ancho de banda que proporcionan para ser aplicados en sistemas SDRadar, que es una característica importante que depende de la capacidad de detección de dos objetivos o blancos en el rango, lo que es mejor conocida como resolución.

D. Moreno es docente de la Escuela de Ingeniería Electrónica en Telecomunicaciones y Redes de la Facultad de Informática y Electrónica, Escuela Superior Politécnica de Chimborazo, Panamericana sur Km 1 y 1/2, Riobamba - Ecuador, pdmoreno@epoch.edu.ec

J. Mejía es egresado y graduado en la Escuela de Ingeniería Electrónica en Telecomunicaciones y Redes de la Facultad de Informática y Electrónica, Escuela Superior Politécnica de Chimborazo, Panamericana sur Km 1 y 1/2, Riobamba - Ecuador, jcesar.mejia.29@gmail.com

H. Moreno es docente de la Escuela de Ingeniería Electrónica en Telecomunicaciones y Redes de la Facultad de Informática y Electrónica, Escuela Superior Politécnica de Chimborazo, Panamericana sur Km 1 y 1/2, Riobamba - Ecuador, h_moreno@epoch.edu.ec

La tarjeta USRP B210, es una tarjeta con tecnología SDR que cuenta con características propias dadas por su fabricante ETTUS RESEARCH, las cuales han sido analizadas para determinar cuál es la mejor resolución que las tarjetas pueden proporcionar para ser aplicadas en sistemas SDRadar.

Al investigar y analizar las características operativas de la tarjeta USRP B210 [8], se ha desarrollado un algoritmo en MATLAB® que ha permitido optimizar la resolución de la tarjeta. Mediante un método cuantitativo, se ha escogido la mejor opción al evaluar la confiabilidad del algoritmo mediante simulaciones.

II. SISTEMA RADAR

A. Resolución en Sistemas Radares

Para la detección de objetivos aparece el concepto de resolución como la capacidad de un radar para distinguir dos o más blancos que están estrechamente espaciados, ya sea en rango, ángulo o frecuencia Doppler [9].

El rango de resolución puede ser calculado al considerar aspectos como ancho de banda, B , exactitud de la distancia, e_d , velocidad de la luz, c , frecuencia máxima, f_{max} , y frecuencia de muestreo, f_s , como se muestra en (1) y (2) [10].

$$f_s = 2 \cdot f_{max} = B \quad (1)$$

Donde la resolución del Sistema está dada por:

$$\Delta R = e_d = \frac{c}{B} = \frac{c}{2 \cdot f_{max}} \quad (2)$$

En base a (2) se observa que a mayor ancho de banda se obtiene una resolución menor.

B. Radar FMCW

El radar de onda continua en frecuencia modulada (FMCW) es un tipo de radar que irradia potencia de transmisión continua como un simple radar de onda continua. Estos radares son modulados en frecuencia lo que permite variar su frecuencia de operación durante la medida.

La Fig.1 muestra una señal chirp generada por un radar FMCW, llamándose up-chirp cuando se genera una señal desde una frecuencia baja a una alta y down-chirp cuando la frecuencia para generar la señal va de una alta a baja [11].

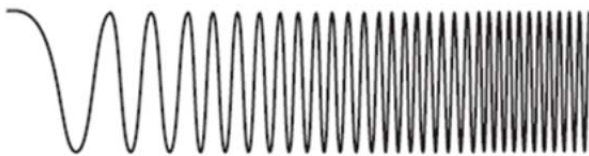


Fig. 1. Señal Chirp. [9]

La distancia máxima, d_{max} , con la que se puede detectar un objetivo es codificada en frecuencia de acuerdo al tiempo máximo, t_f , que genera la señal.

$$d_{max} = \frac{c \cdot t_f}{2} \quad (3)$$

Las características que presentan estos radares son:

simplifica el diseño del transmisor [12], procesamiento sencillo de la señal, sistema a bajo coste, en el dominio del tiempo la señal se presenta como un chirp, en el dominio de la frecuencia se representa como una rampa lineal, la medición de la distancia se obtiene al comparar la señal transmitida o de referencia con la señal recibida o señal eco [13], capacidad para medir simultáneamente el rango del objetivo y su velocidad y alta exactitud de la medición del alcance.

Bajo estos conceptos que se han adaptado a las necesidades de esta investigación, se optó por usar este tipo de radar por las ventajas y características mencionadas.

C. Señal Transmitida

La señal que transmite un radar FMCW se representa de acuerdo a la función $V(t)$

$$V(t) = V_0 \cdot \cos[2 \cdot \pi \cdot f(t) \cdot t + \phi_0], \quad (4)$$

donde V_0 es el valor de amplitud, ϕ_0 es el desfase existente de la onda sinusoidal y la función $f(t)$ representa el valor de la frecuencia que varía linealmente en función del tiempo, que viene dada por:

$$f(t) = \frac{k}{2} \cdot t + f_0 \quad (5)$$

donde k representa la constante de la cual depende la función para variar su frecuencia, que está dada en función de la frecuencia inicial, f_i , y la frecuencia final, f_0 , como sigue:

$$k = \frac{f_0 - f_i}{2} \quad (6)$$

El ancho de banda, B , del sistema, se obtiene de la diferencia $B = f_0 - f_i$. Al reemplazar el ancho de banda en (6) se obtiene:

$$k = \frac{B}{2} \quad (7)$$

D. Señal Recibida

La señal recibida es una señal similar a la transmitida, pero retardada un tiempo t_r debido al tiempo de escucha como se observa en la Fig. 2, lo cual es importante para detectar el objetivo.

$$V_r(t) = V_0 \cdot \cos\left\{2\pi \cdot \left[\frac{k}{2} \cdot (t - 2 \cdot t_r) + f_0\right] \cdot (t - 2 \cdot t_r) + \phi_0\right\} \cdot \mu(t - 2 \cdot t_r) \quad (8)$$

E. Detección del Objetivo

Cuando una señal de eco es recibida, esta variación de frecuencia causa un retardo Δt (i.e., desplazamiento de tiempo) tal como sucede en la técnica de pulsos radar, por lo que se realiza una comparación entre la señal transmitida y el eco de la señal recibida, esto se observa en la Fig. 3.

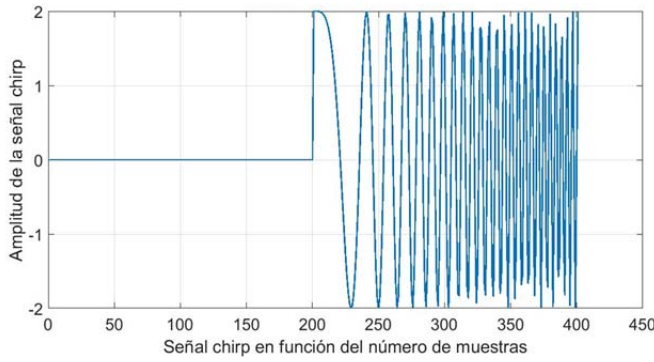


Fig. 2. Señal Recibida.

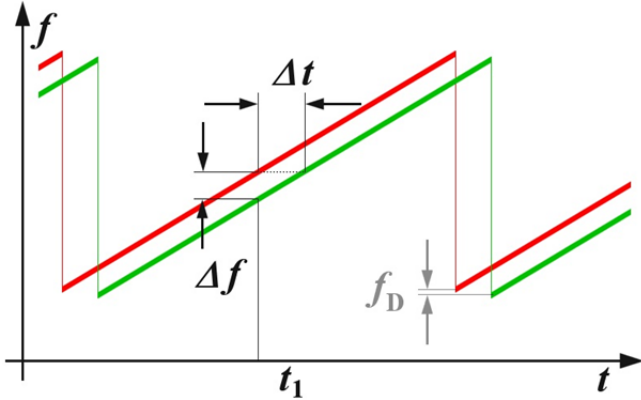


Fig. 3. Oscilaciones de un Sistema FMCW. [13]

La distancia R hacia el objeto reflectante puede ser determinada mediante:

$$R = \frac{c \cdot \Delta t}{2} = \frac{c \cdot \Delta f}{2 \cdot (df/dt)}, \quad (9)$$

donde c es la velocidad de la luz (i.e., 3×10^8 [m/s]), Δt es el retardo de tiempo [s], Δf es la diferencia de frecuencia medida [Hz], R es la distancia entre la antena y el objeto reflectado [m] y df/dt es el cambio de frecuencia por unidad de tiempo.

Si el cambio en la frecuencia es lineal sobre un intervalo amplio del espectro, entonces el rango del radar puede ser determinado por una simple comparación de frecuencia. La diferencia de frecuencia Δf es proporcional a la distancia R .

Como solamente se puede medir el valor absoluto de la diferencia de frecuencia (i.e., no existen valores negativos de frecuencia), los resultados de un incremento lineal de frecuencia son iguales a los de un decremento lineal de frecuencia, esto en un escenario estático, es decir sin efecto Doppler. En el caso de un objeto reflectante con una velocidad radial con respecto a la antena receptora, se debe realizar el análisis del efecto Doppler correspondiente en el cual ocurrirá un desplazamiento de la frecuencia portadora correspondiente [13], este efecto se observa en la señal diente de sierra equivalente a la imagen reflejada de la Fig. 3

Al escoger correctamente la desviación de frecuencia por unidad de tiempo, se puede determinar la resolución del radar, y mediante la selección de la duración del incremento de la frecuencia (i.e., el flanco más largo de la señal diente de sierra,

línea roja en la Fig. 3), se puede calcular la máxima distancia no ambigua medible. La máxima variación de frecuencia y la pendiente del flanco se pueden variar al considerar las capacidades tecnológicas del circuito implementado.

La detección de un objetivo dependerá del valor de la resolución, su resultado será en múltiplos de ΔR . El radar FMCW, para ubicar el objetivo, realiza la multiplicación de las dos señales (transmitida/recibida) FMCW en el dominio del tiempo (10), para esto es necesario de la utilización de la Transformada de Fourier Rápida (FFT), la misma que permite pasar del dominio del tiempo al dominio de la frecuencia, y permite representar al objetivo como un pico con el valor máximo según su radargrama, el cual se muestra en la Fig. 4.

$$V_f(t) = V(t) \cdot V_r(t) \quad (10)$$

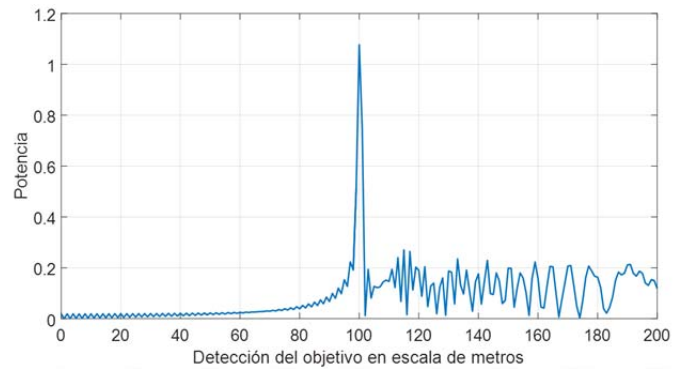


Fig. 4. Radargrama para detectar objetivo.

III. ANÁLISIS USRP B210

A. Software Defined Radar (SDRadar)

La tecnología de Sistemas SDRadar reduce el uso de hardware (e.g., generadores de señal, mezcladores, moduladores, filtros), mediante la implementación vía software mediante sistemas embebidos o FPGA.

Estos sistemas conocidos como SDRadar (Software Defined Radar), reducen considerablemente los costos de implementación con una gran variedad de versatilidades, capacidades y aplicaciones, como se muestra en la Fig. 5.

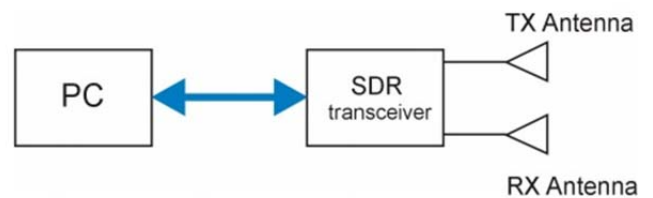


Fig. 5. Radargrama para detectar objetivo.

Un sistema de medición clásico de radar se compone de hardware dedicado, y depende de la aplicación (e.g., Doppler, clima, Radar de penetración terrena, entre otros). La nueva concepción de Software Defined Radar provee un alto grado de flexibilidad y adaptabilidad para varias técnicas de procesamiento de señal. En un sistema de SDRadar la mayoría de las operaciones concernientes al procesamiento de señal se

llevan a cabo vía software; esto permite la creación de un radar multi propósito cuyos diferentes algoritmos, para cada aplicación, toman lugar en el mismo dispositivo. Los algoritmos necesarios de procesamiento de datos están directamente conectados a un terminal de PC.

La clave de SDRadar es la generación de la señal, la cual está relacionada al tipo de señal que será transmitida y después procesada. Existen varias técnicas que permiten la generación de una señal adecuada para SDRadar: la codificación a través de la modulación de un tren de pulsos binarios, modulación a través de señales ortogonales (OFDM), o Modulación de Frecuencia a Onda Continua (FMCW), entre otras.

Cada una de estas técnicas utilizan diferentes modelos matemáticos; e.g., FMCW genera una señal sinusoidal cuya frecuencia cambia con respecto al tiempo transcurrido. Esto significa que la frecuencia se incrementa tanto como dura la señal. Este tipo de señal y su respectivo procesamiento en frecuencia puede ser muy útil para determinar distancias de objetivos, el movimiento de los mismos e incluso su velocidad.

Con estas aplicaciones, es posible realizar ciertas variaciones en el desarrollo del software del radar y crear diversos sistemas, solamente mediante el cambio de la programación y procesamiento de datos, tal como el cálculo de la permitividad relativa de un sustrato dieléctrico.

B. Características USRP B210

Las características de la tarjeta USRP B210 que han sido utilizadas para elaborar el algoritmo se detallan en la Tabla I.

TABLA I
CARACTERÍSTICAS USRP B210

Características	Descripción
Frecuencia de Operación	70 - 600 MHz
Ancho de Banda	56 MHz
Transmisión / Recepción	2 TX, 2 RX, Half duplex o full duplex
Potencia	> 10 dBm
Conectividad	USB 3.0
Configuración	Matlab. GNU Radio
FPGA	Xilinx Spartan 6 XC6SLX150
Velocidad Máxima	50 Mbps

C. Conectividad

La tarjeta para ser configurada por software necesita de una conexión por medio de un puerto USB 3.0, que corresponde a una velocidad de 3.2 Gbps equivalente a 56 MHz de ancho de banda máximo de la señal FMCW [14]. Sin tomar en cuenta que los sistemas radar usan el mismo puerto USB para transmisión y recepción lo cual divide su capacidad de transmisión.

Cabe mencionar que las tarjetas B210 también pueden trabajar a través de un puerto USB 2.0, pero la velocidad (480 Mbps equivalente a 8 MHz de ancho de banda máximo de la señal FMCW) es menor en comparación al puerto USB 3.0, por lo que es recomendable usar el puerto USB 3.0 para aplicaciones SDRadar.

D. Transmisión / Recepción

Una de las principales características de la tarjeta USRP B210 es que cuentan con 2 puertos de transmisión (TX) y 2 puertos de recepción (RX), para una comunicación half-dúplex o full-dúplex. Uno de los inconvenientes de usar la B210 como transceptor es el uso de recursos computacionales.

E. Configuración (software)

USRP B210 presenta dos opciones de configuración de software, por medio de MATLAB® y GNU RADIO.

Para la configuración del software mediante MATLAB® existen paquetes desarrollados por ETTUS que hacen posible la comunicación MATLAB-USRP B210, pero una desventaja que presenta esta comunicación es que MATLAB® al ser un potente software ocupa recursos computacionales [15], lo cual incrementa los recursos necesarios ya que la USRP B210 en si necesita de la conectividad USB 3.0 que representa más carga computacional al sistema.

Para reducir la carga computacional que provoca un sistema SDRadar, GNU RADIO es una alternativa a este problema ya que al ser un conjunto de herramientas de desarrollo de software libre y de código abierto que proporciona bloques de procesamiento de señales para implementar radios de software [16], reduce notablemente el uso de recursos computacionales. Aunque existen distribuciones de GNU RADIO para sistemas operativos como Linux, Windows o MAC, es recomendable usar LINUX ya que por características este Sistema Operativo es más robusto, rentable y rápido [17].

F. Velocidad Máxima de Transferencia de datos

Ya en otras investigaciones se ha analizado sobre como la velocidad máxima de transferencia de datos influye en el ancho de banda para sistemas SDRadar [18], [19], en este caso para USRP B210 se tiene una velocidad de transferencia de 50 Mbps, sin embargo, debido a que la USRP utiliza muestras complejas I/Q, el ancho de banda B con el cual se puede trabajar es de 25 MHz. Al reemplazar el valor de ancho de banda en (2) se obtiene $\Delta R = 6$ m.

El valor de ΔR es la capacidad que tiene el radar para distinguir un blanco en múltiplos de 6 m. La Tabla II detalla cómo se puede distinguir un blanco de acuerdo a la resolución.

TABLA II
RESOLUCIÓN USRP B210

Distancia del blanco (m)	Distancia dada por el software (m)
$0 < d \leq 6$	6
$6 < d \leq 12$	12
$12 < d \leq 18$	18

G. Potencia de Salida

Es importante que los sistemas de comunicaciones como los sistemas SDRadar cuenten con una potencia de salida adaptable a los diferentes escenarios, utiliza una menor potencia cuando se requiere cubrir una distancia corta y una mayor potencia para distancias mayores. A pesar de que la tarjeta USRP B210 en su hoja técnica [20] menciona que

puede irradiar una potencia $> 10\text{dBm}$, estudios han demostrado su inconsistencia al momento de ser ejecutada en tiempo real, y se recomienda que las antenas deben estar separadas a pocos centímetros por sus problemas en potencia [21], lo que dificulta su implementación para SDRadar debido a que necesita una gran cantidad de potencia para cubrir diferentes escenarios [22]. Para solucionar este problema se simula retardos en el receptor para procesar los datos recibidos, que se necesita para el desarrollo del sistema.

H. Sincronización Tx / Rx

Para realizar el análisis de la señal fue necesario establecer un punto de referencia para transmisión, y a partir de este punto con los datos procesados del receptor, detectar uno o más objetivos. Por lo tanto, es necesario que el transmisor y el receptor de la tarjeta USRP B210 se encuentren sincronizados.

Uno de los problemas que la tarjeta USRP B210 presenta es la inestabilidad en la sincronización entre transmisor y receptor al momento de usar la tarjeta como transceptor, ya que el transmisor se activa en tiempos diferentes al receptor sea antes o después y no se puede obtener un punto de referencia para calibrar la tarjeta.

Ante los problemas de sincronización de la tarjeta, el fabricante ha incluido una solución, se trata de un puerto de entrada en la tarjeta para GPSDO, que es un GPS controlado por un oscilador de cristal que permite crear sistemas útiles para aplicaciones que requieren de una mejor precisión en frecuencia y tiempo [23].

IV. DISEÑO DEL ALGORITMO

Para el desarrollo del algoritmo, se partió en base al análisis de las limitaciones de la tarjeta y principalmente del estudio de la velocidad máxima de transmisión de la tarjeta, donde se determinó que el ancho de banda con el cual la tarjeta puede trabajar es de $B = 25\text{ MHz}$, el cual entrega como resultado una resolución $\Delta R = 6\text{ m}$.

El objetivo del algoritmo es optimizar la resolución de la tarjeta de $\Delta R = 6\text{ m}$ a un valor menor, por lo que fue necesario realizar n mediciones, para generar n señales cada una con un diferente ancho de banda, B_n , que varía en $m\text{ MHz}$ con respecto a B_{n-1} , que corresponde a una diferente resolución $\Delta R > 6\text{ m}$. En cada una de las mediciones se aplicó la teoría clásica del funcionamiento básico de un sistema radar [24].

En la Fig. 6, se puede observar el contexto de la detección de un objeto ubicado a una distancia d . Como cada ancho de banda B_n de cada medición provoca una nueva resolución ΔR_n , el resultado de distancia arrojado por el sistema es de R_n en cada uno de los casos, todas estas mediciones fueron almacenadas en una matriz. En la columna 2 de la matriz se almacenaron todos los resultados dados por software para la detección del objetivo (distancia R_n) y en la columna 1 se restó la resolución correspondiente del objetivo detectado de la columna 2 ($R_n - \Delta R_n$).

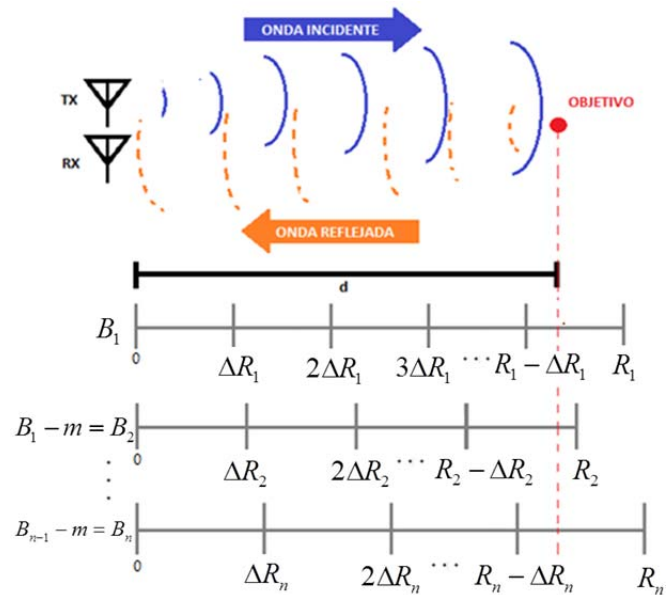


Fig. 6. Método de optimización de resolución.

De los valores almacenados en la matriz se encontró el valor mayor de la columna 1 y el menor valor de la columna 2, promediándose estos valores para obtener el resultado de optimización de resolución. El proceso del algoritmo se detalla en el diagrama de flujo de la Fig. 7.

A. Determinación de zonas para mediciones y valores

Dado que el objetivo es optimizar la resolución de la tarjeta USRP B210, se tomó en cuenta que el mayor error de medición fue 6 metros, y se fijaron diferentes zonas posibles, limitadas por un margen de error, e , las cuales representan la diferencia entre la distancia real del objetivo y la distancia arrojada por el sistema, dichos valores se encuentran dentro las zonas mostradas en la Tabla III.

TABLA III
ZONAS SEGÚN EL MARGEN DE ERROR

Zonas	Margen de Error (m)
1	$e < 1$
2	$1 \leq e < 2$
3	$2 \leq e < 3$
4	$3 \leq e < 4$
5	$4 \leq e < 5$
6	$5 \leq e \leq 6$

Cuando se detectó un objetivo según el valor de la resolución del sistema, entregó como respuesta de detección de objetivo un valor múltiplo de la resolución. Por ejemplo, en la Tabla IV se muestra los valores mínimo/máximo (Columna 1 y Columna 2, respectivamente) que se obtendría al realizar la detección al utilizar el máximo ancho de banda, es decir, una resolución de $\Delta R = 6\text{ m}$. Además, se indica los posibles rangos de ubicación del objetivo correspondientes a la resolución de la tarjeta USRP B210 sin optimización.

TABLA IV
VALORES MÍNIMO / MÁXIMO PARA USRP B210

Distancia al Objeto (m)	Distancia Software (m)	Mínimo (m)	Máximo (m)
$0 < d \leq 6$	6	0	6
$6 < d \leq 12$	12	6	12
$12 < d \leq 18$	18	12	18

B. Proceso para calcular el número de muestras tomadas

Se analizó el procedimiento para encontrar los valores de las incógnitas m y n mencionadas anteriormente, que se determinó en el método para optimizar la resolución de la tarjeta USRP B210. Para el análisis se realizó la simulación de 100 mediciones distintas de un objetivo ubicado desde 1 a 100 metros de distancia de la antena. Se simularon diversos escenarios en los cuales se modificó la distancia real del objetivo, d , la cantidad de mediciones, n , y la variación del ancho de banda, m , en cada una, para encontrar el valor de las incógnitas m y n en un punto que permita mayor precisión de medida del objetivo. Los valores de m están dados en MHz y n en número de mediciones.

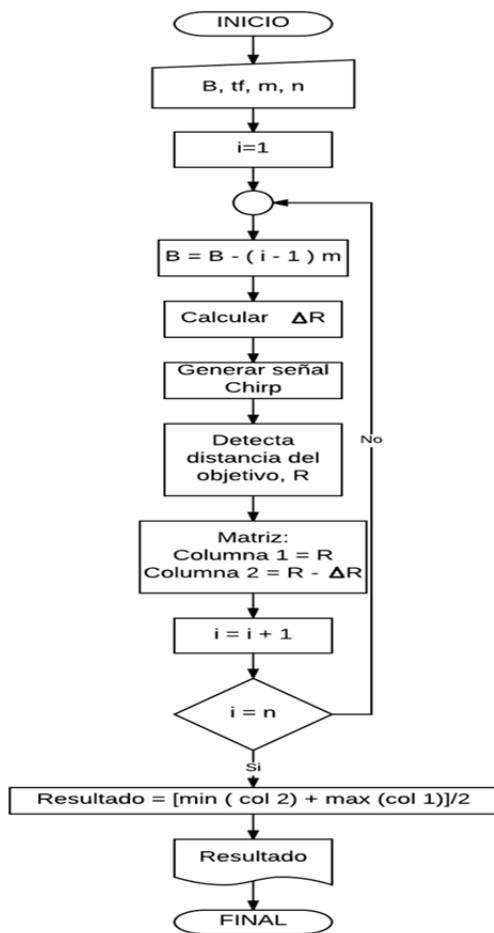


Fig. 7. Diagrama de flujo del algoritmo.

El proceso realizado para determinar los valores de las incógnitas m y n se lo hizo en base a la estimación de detectar una mayor cantidad de objetivos dentro de la zona 1, es decir, con un margen de error inferior a 1 metro. Para esto se realizó

la aplicación del algoritmo de la Fig. 7 en mediciones con diferentes variaciones de ancho de banda, las cuales fueron desde 0.1 MHz a 2 MHz de 2 a 20 mediciones para cada variación de ancho de banda. La Fig. 8 muestra el resultado de las mediciones en donde se representa el mayor número de objetivos detectados en la zona 1 para cada uno de los valores de m , eligiéndose el valor de m que brinda mayor detección de objetos en la zona 1, en este caso $m = 0.5$ MHz, que representa un mayor número de objetivos encontrados con menor margen de error.

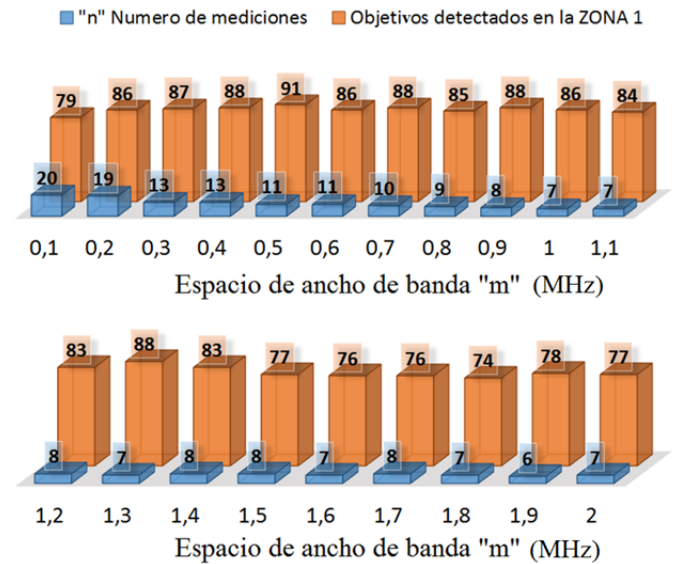


Fig. 8. Análisis para encontrar valor de m y n

Una vez encontrado el valor de m se hizo un análisis para encontrar el valor de n al considerar que la cantidad de mediciones a partir de un cierto valor provoca la misma cantidad de objetivos detectados en la zona 1, por lo tanto se continúa con los siguientes valores de n para determinar en este caso la mínima cantidad de objetos detectados en las zonas de mayor margen de error. El resultado se muestra en la Fig. 9.

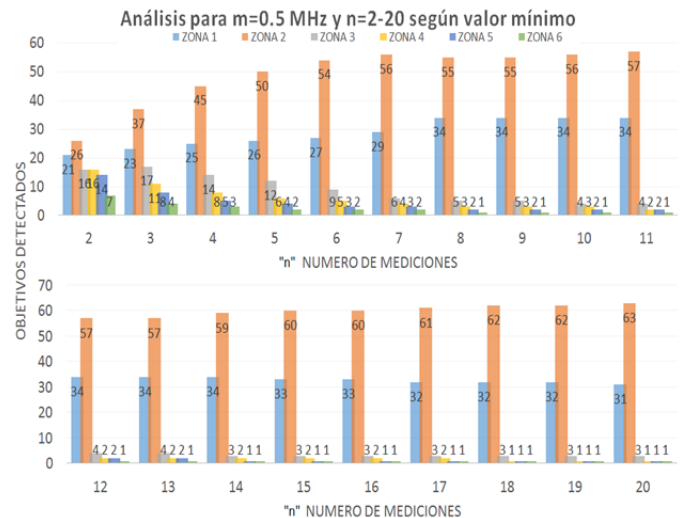


Fig. 9. Análisis para encontrar valor de n según valores mínimos

Una vez analizados los datos se determinó, por un lado, que a partir de $n = 11$, se obtiene la mayor cantidad de objetos detectados en la zona 1. Por otro lado, al analizar según la menor cantidad de objetos detectados en las zonas de mayor margen de error se obtiene que para $n = 14$ se detectó únicamente un objetivo en la zona 5 como se observa en la Fig. 9.

Al relacionar el análisis anterior la optimización del algoritmo se obtuvo a $n = 14$ mediciones con $m = 0.5$ MHz, según los valores mínimo/máximo de objetivos detectados.

C. Corrección de la Resolución

Una comparación de la diferencia que existe entre una resolución de 6 m en relación a una resolución de 1 m se la ha representado en la Fig. 10, en donde la resolución de 1 metro tiende a una función lineal de pendiente unitaria.

Posteriormente con los valores encontrados de n y m en la elección de número de muestras y variación de ancho de banda se aplicó al algoritmo.

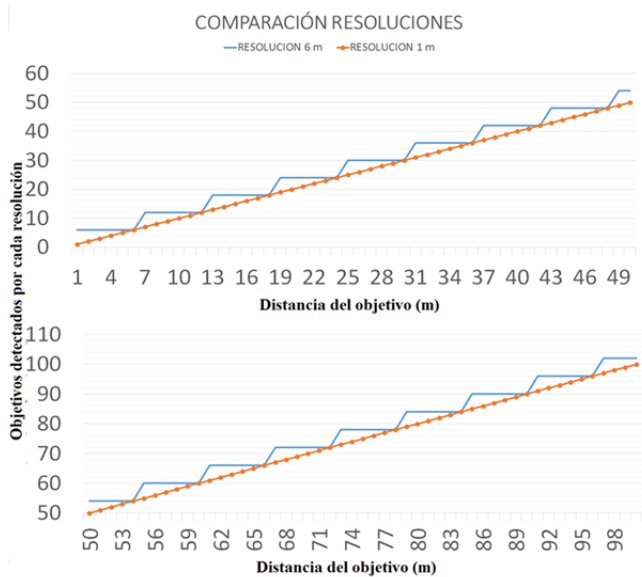


Fig. 10. Corrección de resolución al aplicar el promedio de valores mínimo/máximo

En la Fig. 11 se observan cuatro líneas; la línea azul es la resolución inicial del sistema $\Delta R = 6$ m, la línea roja es una referencia de una resolución de $\Delta R = 1$ m que se la gráfica para poder estimar resultados, la línea gris y amarilla son los valores mínimo y máximo, respectivamente, corregidos de acuerdo a cada punto, por el contrario el área de color celeste es toda la corrección de resolución realizada con el algoritmo.

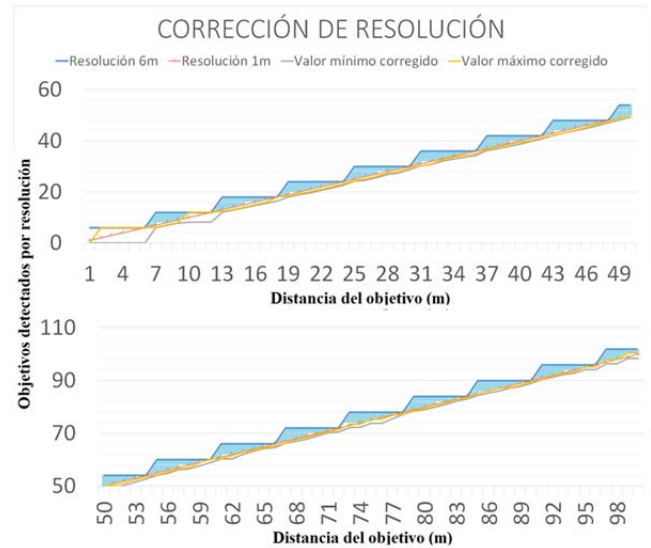


Fig. 11. Corrección de resolución

Los parámetros para generar la señal chirp y optimizar la resolución se citan en la Tabla V.

TABLA V
PARÁMETROS PARA GENERACIÓN DE LA SEÑAL

Parámetros de la Señal	Valores
Número de muestras m (MHz)	14 0.5
Anchos de banda (MHz)	25, 24.5, 24, 23.5, ... 18.5
Tiempo máximo de la señal	$6.67 * 10^{-6}$ s

Una vez obtenidos los valores mínimos y máximos de cada objetivo detectado se computó un promedio para imprimir el resultado de ubicación de cada objetivo, como resultado se obtuvo un margen de error $e < 2$. Para interpretar de mejor manera los resultados se determinaron dos zonas de optimización de resolución como se muestra en la Tabla VI, y la resolución dependiendo de cada zona.

TABLA VI
RESOLUCIÓN CORRESPONDIENTE A CADA ZONA DETERMINADA

Zonas	Margen de Error (m)	Resolución ΔR (m)
1	$e < 1$	1
2	$1 \leq e < 2$	2

La Fig. 12 muestra los resultados obtenidos con el promedio obtenido entre el valor mínimo y máximo de cada objetivo.

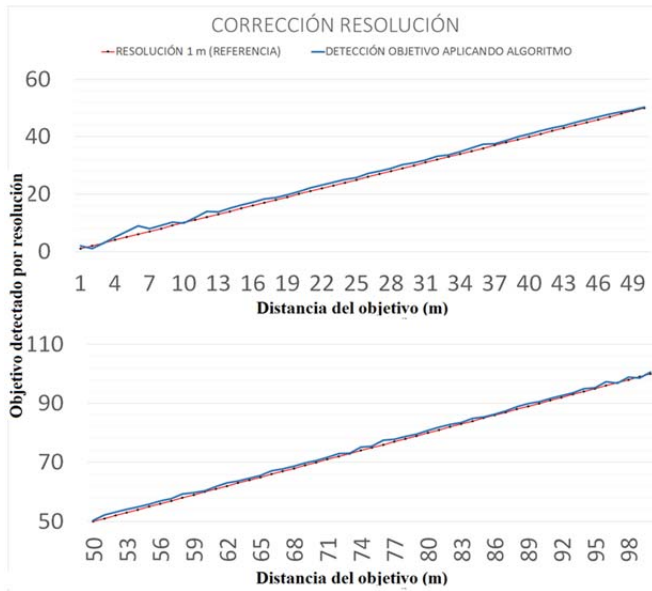


Fig. 12. Corrección de resolución al aplicar el promedio de valores mínimo/máximo

D. Limitaciones del Algoritmo

Para el análisis final se redujo el número de objetivos de 100 a 94 ya que se restaron los valores de las primeras seis medidas debido a que con estas medidas no es posible mejorar la resolución al encontrarse dentro del primer rango máximo de resolución sin optimización. El análisis siguiente se realizó con el nuevo número de muestras.

En la Fig. 13 los objetivos detectados en la zona 2 se representan en color verde y los objetivos restantes (representados sin color) están en la zona 1.

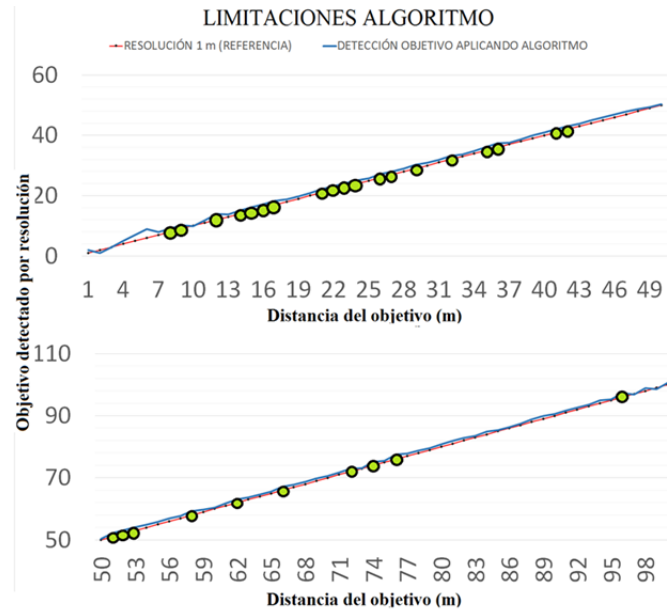


Fig. 13. Limitaciones Algoritmo

En la Tabla VII se muestra en resumen el desempeño del algoritmo desarrollado en MATLAB®.

TABLA VII
RESUMEN DE RESULTADOS OBTENIDOS

Zonas	Margen de error (m)	Objetivos Detectados	Distancias en zonas
2	$1 \leq e < 2$	29	8, 9, 12, 14 - 17, 21 - 24, 26, 27, 29, 32, 35, 36, 41 42, 51 - 53, 58, 62, 66, 72, 74, 76, 96
1	$e < 1$	65	Valores Restantes

V. RESULTADOS

Para evaluar los resultados, el sistema SDRadar propuesto consta con las siguientes etapas como se muestra en la Fig. 14.

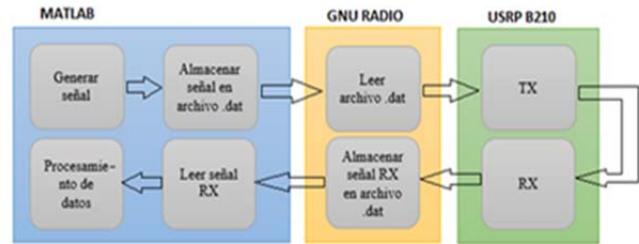


Fig. 14. Diagrama de bloques sistema SDRadar.

El sistema propuesto se lo realizó con la ayuda de MATLAB® para generar la señal y procesar los datos recibidos de las etapas realizadas con GNU RADIO y USRP para poder estimar la distancia del objetivo. Se propuso este sistema para reducir la carga computacional que requiere implementar un sistema SDRadar.

A. Evaluación Zona 1

Se simuló en MATLAB® la distancia de 93 metros del objetivo a encontrar como se muestra en la Fig. 15. Al tomar en cuenta que la distancia simulada no es múltiplo de la resolución de la tarjeta $\Delta R = 6$ m, por lo tanto sin aplicar el algoritmo la detección del objetivo tuvo un error en su medida.

La Fig. 16 muestra el radargrama con la posición del objetivo de acuerdo a la resolución de 6 m. El resultado al aplicar el algoritmo desarrollado para optimizar la resolución de la tarjeta USRP B210, con un error de detección $e < 1$ m, se muestra en la Fig. 17.

```

Command Window
ingrese la distancia del objetivo en un rango de 6 a 100 metros: 93
El objetivo con respecto a una resolución de 6m se encuentra a:
96.3855

```

Fig. 15. Detección de objetivo en zona 1 sin aplicar algoritmo de corrección.

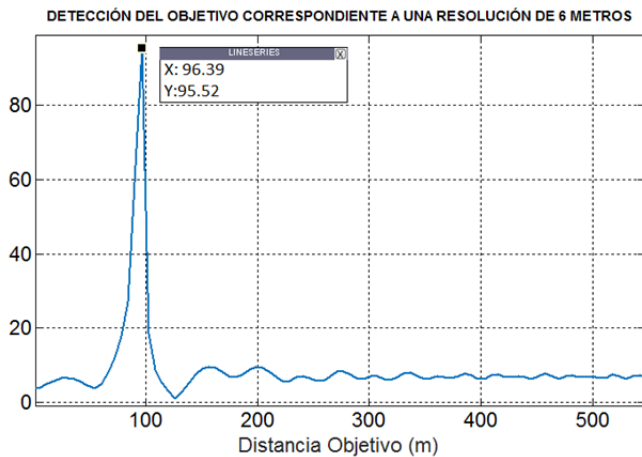


Fig. 16. Radargrama para detección del objetivo en zona 1 con una resolución de 6 metros.

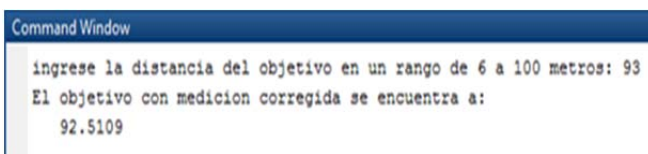


Fig. 17. Resultado de medición de objetivo de 93 metros aplicando algoritmo.

B. Evaluación Zona 2

Se simuló en MATLAB® la distancia de 51 metros del objetivo a encontrar como se muestra en la Fig. 18. Al tomar en cuenta que la distancia simulada no es múltiplo de la resolución de la tarjeta $\Delta R = 6$ m, por lo tanto sin aplicar el algoritmo la detección del objetivo tuvo un error en su medida. En la Fig. 19 se muestra el radargrama con la posición del objetivo de acuerdo a la resolución de 6m. Los resultados obtenidos, al aplicar el algoritmo desarrollado para optimizar la resolución, se muestra en la Fig. 20.

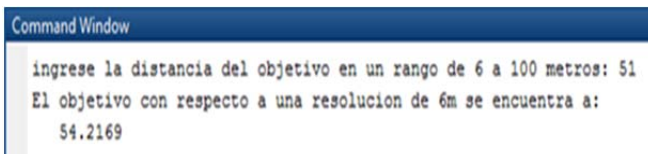


Fig. 18. Detección de objetivo en zona 2 sin aplicar algoritmo de corrección.

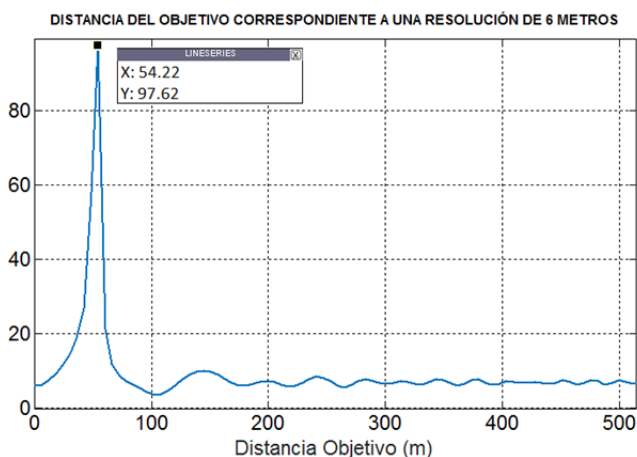


Fig. 19. Radargrama para detección del objetivo en zona 2 con una resolución de 6 metros.

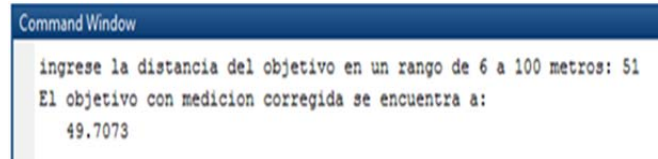


Fig. 20. Resultado de medición de objetivo de 51 metros aplicando algoritmo.

C. Discusión de Resultados

De las noventa y cuatro muestras evaluadas en las dos zonas determinadas, se obtuvieron 65 objetivos en la zona 1 con un error $e < 1$ m el cual representa un 69.15% de mediciones, así mismo se obtuvieron 29 objetivos en la zona 2 con un error de $1 \text{ m} \leq e < 2 \text{ m}$ igual a 30.85% de mediciones con este error, como se muestra en la Tabla IX.

VI. CONCLUSIONES

Para optimizar la resolución de la tarjeta USRP B210 se desarrolló un algoritmo en MATLAB®, al considerar el mejor ancho de banda que proporciona la tarjeta, se determinó que con $n = 14$ mediciones sobre el objetivo a detectar con una diferencia entre anchos de banda $m = 0.5$ MHz de las señales generadas el resultado de optimización fue semejante a una resolución de 2 metros. El algoritmo desarrollado en MATLAB® optimizó la resolución a partir de 6 metros de distancia del objetivo a detectar, con una muestra de 94 compuesta por objetivos simulados situados de 6 a 100 metros de distancia donde se determinaron dos zonas: la primera que estuvo limitada por un error de medición de $e < 1$ m, la segunda zona con un error de medición $1 \text{ m} \leq e < 2 \text{ m}$ con 69.15% y 30.85% de objetivos detectados respectivamente, lo que determinó su efectividad y confiabilidad. Para reducir la carga computacional que un sistema SDRadar comprende, se utilizó MATLAB® y GNUradio para generar la señal además de procesar los datos obtenidos de la tarjeta USRP B210 que fue usada como transceptor, donde se concluyó que por el número de mediciones que se deben realizar el algoritmo desarrollado puede ser empleado en aplicaciones SDRadar como detección de objetivos, aplicaciones topográficas o en radares de apertura sintética.

RECONOCIMIENTOS

Este trabajo fue realizado por parte del proyecto de investigación en Radares de la ESPOCH.

REFERENCIAS

- [1] J. Sanfuentes., "Historia del radar". Revista de Marina. Armada de Chile. Santiago. p. 1-7, 2000.
- [2] D. de la Mata, "Diseño de detectores robustos en aplicaciones radar". Ph.D. dissertation, Universidad de Alcalá, Madrid, España. 2012.
- [3] C. Gutierrez and A. Nieto, "Teledetección: nociones y aplicaciones". Salamanca, España, Tech. Rep. Jul 2006.
- [4] S. Constanzo et al., "High Resolution Software Defined Radar System for Target Detection," *Journal of Electrical and Computer Engineering*, p. 7, Ene 2013.
- [5] E. Patiño and L. Serrano, "Programación de una plataforma sdr (software defined radio) para la detección de minas antipersonas". Tech. Rep. Mar 2014.

- [6] E. Hernández et al., "Uso de FPGA para realizar compresión del pulso de radar", *Científica*, vol. 9, pp. 73-81, Oct 2005.
- [7] Y. Yang and A. Fathy, "Development and implementation of a real-time see-through-wall radar system based on FPGA", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, pp. 1270-1280, May 2009.
- [8] L. Aldaz., "Análisis y estudio de la simulación de los parámetros del radar en el programa MATLAB", Escuela Superior Politécnica del Ejército, Latacunga, Ecuador, Tech. Rep., Ene 2005.
- [9] M. Richards. "Principles of modern Radar". Raleigh NC United States of America: Scitec Publishing Inc. 2010.
- [10] J. Barberán and R. Dominguez, "Sistema Radar para reconstrucción de objetos 3D", Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador, Tech. Rep., Mar 2016.
- [11] R. Alvarez and S. Shagñay, "Sistema Radar Para Análisis Y Reconstrucción De Perfiles Topográficos A Media Escala", Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador, Tech. Rep., May 2015.
- [12] G. Charvat. "Small And Short Range Radar System". United States of America: CRC Press. 2014.
- [13] C. Wolff, "Radar Tutorial", Febrero 2017. [online]. Disponible: <http://www.radartutorial.eu/02.basics/Frequency%20Modulated%20Continuous%20Wave%20Radar.en.html>.
- [14] J. Requena and F. Beltrán, "Modulaciones digitales", Londres, UK and Bogotá, Colombia, 2015.
- [15] Mathworks, "System Requirements & Platform Availability". Marzo 2017.
- [16] Ettus, "USRP B210 (Board Only)", Mayo 2017. [online]. Disponible: <https://www.ettus.com/product/details/UB210-KIT>.
- [17] F. Mejía et al., "Implementar un DOMAIN controller basado en el protocolo LDAP, que incluya servicios GATEWAY bajo plataformas LINUX, para ser usado en las aulas informáticas educativas gubernamentales del Municipio de Metapán", Universidad Francisco Gavidia, San Salvador, El Salvador, Tech. Rep., Feb. 2012.
- [18] S. Constanzo et al., "Potentialities of USRP Based Software Defined Radar Systems", *Progress in Electromagnetism Research B*, vol. 53, pp. 417-435, Jun 2013.
- [19] C. Merchán, "Diseño e implementación de un sistema radar utilizando sistema sdr a través de tecnología usrp para aplicaciones topográficas", Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador, Tech. Rep., Nov 2016.
- [20] Ettus, "USRP B200/B210 Bus Series", Mayo 2017. [online]. Disponible: https://www.ettus.com/content/files/b200-b210_spec_sheet.pdf.
- [21] B. Saraguro, "Implementación de un sistema de transmisiones para televisión digital terrestre en el Ecuador a través de módulos USRP", Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador, Tech. Rep., Jun 2016.
- [22] J. Fominaya, "Nuevas técnicas de localización, clasificación e identificación para radares en vigilancia superficial y alta resolución en escenarios LPI", Ph.D. dissertation, Universidad Politécnica de Madrid, Madrid, España, 2004.
- [23] Ettus, "Board Mounted GPSDO (OCXO)", Mayo 2017. [online]. Disponible: <https://www.ettus.com/product/details/GPSDO-MINI>.
- [24] J. Zozaya, "Simulador básico de un radar de apertura sintética", Marzo 2015. [online]. Disponible: https://www.researchgate.net/profile/Alfonso_Zozaya/publication/280287287_Simulador_básico_de_un_radar_de_apertura_sintética/links/55afc94408ae32092e06ec19.pdf

Análisis e implementación de diferencial GPS en configuración simple y doble

Analysis and implementation of differential GPS in single and double configurations

Mónica Zabala, Franklin López, y Ángel Ortega.

Abstract— The standard positioning service (SPS), or undifferentiated measurements, contains errors in its measurements, affecting the final positioning solution of the receiver. The application of GPS differentiation techniques in simple (SD) and double (DD) configurations allows the elimination of errors that affect the GPS signal during the propagation path; among the most critical are errors produced by the atmosphere and troposphere. Between the reference receiver and the mobile GPS receiver (rover), errors are correlated in time and space according to the distance between them, and the elimination of common errors increases the accuracy of the final solution. The DGPS-Lab application is designed to execute the differentiation process in both configurations and verify the results by minimizing the errors in the positioning through the implemented DGPS algorithm.

Index Terms—GPS, DGPS, Raw Data, single difference, double differences.

Resumen—El servicio de posicionamiento estándar (SPS), o medidas sin diferenciar, contiene errores en sus medidas que perturban en la solución final de posicionamiento del receptor. A través de la aplicación de técnicas de diferenciación de GPS en configuración simple (SD) y doble (DD) es posible la eliminación de errores que afectan a la señal de GPS en el trayecto de propagación, se considera los más críticos a los errores producidos por la atmósfera y tropósfera. Entre el receptor de referencia y el dispositivo receptor GPS móvil (rover) existen errores correlacionados en tiempo y espacio de acuerdo a la distancia que exista entre éstos. La eliminación de errores correlacionados incrementa la precisión en la solución final. Se diseña la aplicación denominada DGPS-Lab creada para ejecutar el proceso de diferenciación en ambas configuraciones y verificar los resultados al minimizar los errores en el posicionamiento a través del algoritmo de DGPS implementado.

Palabras Claves—GPS, DGPS, Datos Brutos, diferencial simple, diferencial doble.

M. Zabala. docente en la Escuela Superior Politécnica de Chimborazo, en la Facultad de Informática y Electrónica, Escuela de Ingeniería Electrónica en Telecomunicaciones y Redes, Panamericana Sur Km 1 1/2, Riobamba – Ecuador, m_zabala@esPOCH.edu.ec.

F. López, Á. Ortega. ingenieros en la Escuela de Ingeniería en Electrónica, Telecomunicaciones y Redes de la Facultad de Informática y Electrónica, Escuela Superior Politécnica de Chimborazo. Riobamba – Ecuador, franklink16@yahoo.es, angelo@gmail.com.

I. INTRODUCCIÓN

DIFERENCIAL de GPS (DGPS) es la técnica que incrementa drásticamente la precisión e integridad de la posición en un receptor [1]. El principio de funcionamiento consiste en utilizar un receptor de referencia de coordenadas conocidas y de un receptor de usuario (rover) para que a través de la diferenciación de sus medidas se elimine errores correlacionados en espacio y tiempo [2], con excepción a los errores de multipath por ser errores locales y exclusivos de cada receptor. Se considera errores correlacionados entre receptores si la distancia entre ellos alcanza a los cientos o miles de kilómetros, así los errores provocados debido a la ionósfera, tropósfera y errores debido al reloj del satélite son mitigados con la utilización de la diferenciación simple (SD), de la misma manera la diferenciación doble (DD) reduce errores producidos debido al reloj del receptor [2]. Con el fin de disminuir estos errores, existen varias técnicas de corrección que incluye la transmisión de las correcciones en tiempo real y por postproceso. El alcance de este documento se enmarca en el desarrollo del algoritmo que permita ejecutar el diferencial de GPS mediante el uso de las medidas de código de un receptor GPS de bajo costo al considerar la diferenciación en configuración simple y doble.

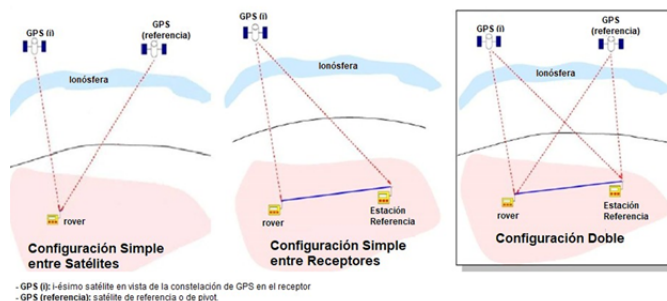


Fig. 1. Configuración simple y doble entre satélites y receptores: La eliminación de errores comunes es posible debido a la diferenciación de las medidas entre satélites en vista y receptores.

II. DGPS CONCEPTO Y ALGORITMO

La técnica de DGPS consiste en la diferenciación de las medidas de código o fase de GPS entre un receptor de referencia y un rover para eliminar errores comunes al

considerar la distancia que existe entre ellos, con el fin de mejorar la precisión del posicionamiento final.

A. Estación de Referencia

La ciudad de Riobamba cuenta con dos estaciones de referencia, denominadas EREC y RIOP, las mismas que forman parte de la red de Monitoreo Continuo del Ecuador (REGME) [3]. La estación EREC, de propiedad de la Empresa Eléctrica Riobamba S.A., es un receptor GNSS (Global Navigation Satellite System) Trimble Net R9 con capacidad multiconstelación, alta versatilidad, diseñado para proveer alto rendimiento como un simple receptor. También es utilizado para la descarga de datos brutos usados para la ejecución de las correcciones de posicionamiento de un receptor GPS por el método de postproceso [4].



Fig. 2. Estación de referencia EREC: Receptor de GPS marca Trimble NetR9. Acceso desde el sitio web administrado por la Empresa Eléctrica S.A.

B. Receptor GPS móvil (rover)

El receptor de GPS de bajo costo U-blox neo 6M V2 forma parte de la familia dispositivos de posicionamiento de precisión estándar [5]. El diseño compacto, la fácil manipulación, y tamaño representa una opción viable de adquisición al ofrecer precisión de 2.5 m y Time-To-First-Fix (TTFF) de menos de 1 s. Sin embargo, existen algunas limitaciones respecto a la obtención de los datos brutos, el protocolo de transmisión por defecto para el rover es NMEA0183 [6] que es incompatible con el formato de estandarizado de almacenamiento Receiver INdependent Exchange (RINEX) de la estación de referencia, es por eso necesario utilizar herramientas de conversión entre formatos. RTKLIB es el software de posicionamiento estándar y preciso que entre las funciones principales tiene la de convertir archivos en diversos formatos estándares de navegación.

C. Ecuaciones de Navegación

La posición estimada del receptor es el baricentro del volumen de la intersección de esferas que tienen un radio igual al pseudorange. Por definición [2], el pseudorange es el tiempo que demora la señal durante la propagación entre el satélite y el receptor δt multiplicada por la velocidad de la luz.

$$R = c\delta t \quad (1)$$

El pseudorange no coincide con la distancia de geometría debido a algunos errores que incluyen la sincronización entre

los relojes del receptor y satélite, a los efectos de propagación de la señal que incluye a factores ionosférico, entre otros. Con el fin de determinar la posición del receptor $\mathbf{r} = (x_r, y_r, z_r)$ y δt desde el pseudorange, al menos 4 satélites en vista son necesarios, así, el pseudorange del satélite i se define [7]:

$$R^i = \rho + c(\delta t_u - \delta t^i) + \alpha_{iono}^i + \beta_{tropo}^i + M + K + \varepsilon^i \quad (2)$$

donde ρ es el rango geométrico entre el satélite y el receptor; δt_u y δt^i son el desfase entre el reloj en el receptor y en el satélite, respectivamente; α_{iono}^i es el retardo ionosférico; β_{tropo}^i es el retardo troposférico; M es el error debido al multipath; K es el retardo instrumental del satélite y del receptor; y ε^i es una variable que representa otros errores locales del receptor.

El pseudorange puede ser reescrito al desprestigiar los términos de multipath y ruidos locales tanto en el receptor como en el satélite.

$$R^i = \rho^i + D^i + c(\delta t_r) \quad (3)$$

donde ρ^i es el rango geométrico entre el receptor y el satélite, D^i es la distancia que contiene los aportes del error debido a la ionósfera, tropósfera y de errores locales del receptor y $c\delta t_r$ es la variable que contiene los errores derivados al reloj del satélite y del receptor.

El rango geométrico ρ^i contiene las 4 coordenadas desconocidas del receptor. Por lo que para determinar la posición, es necesario formar un sistema de ecuaciones uniforme con al menos 4 ecuaciones para resolver las incógnitas de posición x, y, z y t .

Para $i = 1, 2, 3, \dots, n$ ($n \geq 4$) satélites la solución de navegación se define [7]:

$$\begin{aligned} \rho^1 &= \sqrt{(x^1 - x_r)^2 + (y^1 - y_r)^2 + (z^1 - z_r)^2} + c(\delta t_r) \\ \rho^2 &= \sqrt{(x^2 - x_r)^2 + (y^2 - y_r)^2 + (z^2 - z_r)^2} + c(\delta t_r) \\ &\vdots \\ \rho^i &= \sqrt{(x^i - x_r)^2 + (y^i - y_r)^2 + (z^i - z_r)^2} + c(\delta t_r) \end{aligned} \quad (4)$$

donde x_r, y_r, z_r son las coordenadas del receptor desconocidas, y x, y, z son las coordenadas cartesianas del i -ésimo satélite calculadas en base a sus efemérides.

El sistema de ecuaciones (4) no es lineal, por lo que su solución no es inmediata. Normalmente se cuenta con una posición aproximada del receptor y se procede a linealizar las ecuaciones en torno a una posición aproximada $\mathbf{r}_o = (x_o, y_o, z_o)$, de la siguiente forme:

$$\rho^i = \rho_o^i + \frac{x_o - x^i}{\rho_o^i} dx + \frac{y_o - y^i}{\rho_o^i} dy + \frac{z_o - z^i}{\rho_o^i} dz \quad (5)$$

con $dx = x_r - x_o$; $dy = y_r - y_o$ y $dz = z_r - z_o$.

Al sustituir (5) en (3), se obtiene:

$$R^i - \rho_o^i - D^i = \frac{x_o - x^i}{\rho_o^i} dx + \frac{y_o - y^i}{\rho_o^i} dy + \frac{z_o - z^i}{\rho_o^i} dz + c(\delta t_r) \quad (6)$$

Al considerar las ecuaciones para i satélites, en notación matricial está dada por:

$$\begin{bmatrix} R^1 - \rho_o^1 - D^1 \\ \vdots \\ R^n - \rho_o^n - D^n \end{bmatrix} = \begin{bmatrix} \frac{x_o - x^1}{\rho_o^1} & \frac{y_o - y^1}{\rho_o^1} & \frac{z_o - z^1}{\rho_o^1} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{x_o - x^n}{\rho_o^n} & \frac{y_o - y^n}{\rho_o^n} & \frac{z_o - z^n}{\rho_o^n} & 1 \end{bmatrix} = \begin{bmatrix} dx \\ dy \\ dz \\ c(\delta t_r) \end{bmatrix} \quad (7)$$

que en su forma compacta se representa como:

$$\mathbf{y} = \mathbf{G}\mathbf{x}, \quad (8)$$

donde \mathbf{y} es un vector que contiene la diferencia de medida entre los pseudorangos observados y estimados, \mathbf{G} es una matriz que contiene la geometría que existe entre el receptor y el satélite y \mathbf{x} es un vector que contiene la desviación entre las coordenadas verdaderas y las coordenadas estimadas, además del offset del reloj del receptor.

La solución del sistema provee las coordenadas desconocidas estimadas del receptor:

$$\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} + \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} \quad (9)$$

La ecuación (8) no incluye las medidas de ruido, si se toma en cuenta, el modelo lineal:

$$\mathbf{y} = \mathbf{G}\mathbf{x} + \boldsymbol{\varepsilon} \quad (10)$$

Debido al término de error el sistema no provee una solución exacta. En este caso la solución está dada por un vector de estimación $\hat{\mathbf{x}}$ el cuál minimiza el error:

$$\mathbf{y} \approx \mathbf{G}\hat{\mathbf{x}} \quad (11)$$

Least Square (LS) método de estimación para calcular la solución de sistemas sobre determinados. Es un criterio común utilizado en GNSS que minimiza la suma de los cuadrados de los residuos:

$$\min \|\mathbf{y} - \hat{\mathbf{y}}\|^2 = \min \left[\sum_{i=1}^n (y_i - \hat{y}_i)^2 \right] \quad (12)$$

donde:

$$\hat{\mathbf{y}} = \mathbf{G}\hat{\mathbf{x}} \quad (13)$$

La diferencia entre las medidas \mathbf{y} y el modelo estimado $\hat{\mathbf{y}}$ es llamado vector residual.

$$\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}} = \mathbf{y} - \mathbf{G}\hat{\mathbf{x}} \quad (14)$$

La solución de mínimos cuadrados ordinaria viene dada por:

$$\hat{\mathbf{x}} = (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{y} \quad (15)$$

D. Técnica de DGPS

La precisión que se alcanza en la estimación de la posición del receptor puede verse mejorada cuando se utilizan técnicas diferenciales, puesto que pueden eliminarse muchas de las fuentes de error. Para que sean aplicables estas técnicas es

necesario contar con un receptor fijo, del que se conoce con precisión su posición. El rover y el receptor de referencia toman medidas que se ven afectadas por los mismos tipos de errores. La técnica de diferenciación aprovecha los errores comunes de ambos receptores para cancelarlos. A continuación, se describe cómo se estima la posición del receptor móvil (rover) con las técnicas diferenciales, mediante el uso de las medidas de pseudorango [7].

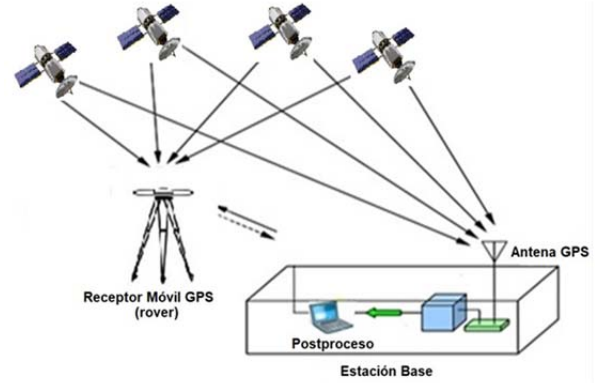


Fig. 3. Concepto de Diferencial de GPS: Estación de referencia y el rover separadas entre sí y con visibilidad de los mismos satélites en las mismas épocas.

La medida de pseudorango R_A^i expresada en unidades de longitud, que toma el receptor fijo A, en un periodo de tiempo denominado época, correspondiente al satélite i :

$$R_A^i = \rho_A^i + c(\delta t_A) - c(\delta t^i) + w_A^i + \alpha_{iono}^i + \beta_{tropo}^i \quad (16)$$

donde ρ_A^i es la distancia geométrica entre el receptor y el satélite; δt_A y δt^i son las derivas de los relojes del receptor A; w_A^i es una variable que contiene los errores debido al ruido del receptor A y al efecto multipath; α_{iono}^i y β_{tropo}^i son los errores producidos por la ionosfera y la troposfera, respectivamente

El pseudorango del rover B:

$$R_B^i = \rho_B^i + c(\delta t_B) - c(\delta t^i) + w_B^i + \alpha_{iono}^i + \beta_{tropo}^i \quad (17)$$

Si la distancia entre la estación de referencia y el rover se considera corta, los errores causados por la ionósfera y tropósfera son comunes o están correlacionados espacialmente. La sustracción de las medidas de los pseudorangos (16) y (17) permite la cancelación de dichos errores y se obtiene lo que se conoce como la diferenciación simple (SD).

$$R_{AB}^i = \rho_{AB}^i + c(\delta t_{AB}) + w_{AB}^i \quad (18)$$

Se ha cancelado la deriva del reloj del satélite. Al repetir el proceso para un satélite j se obtiene:

$$R_{AB}^j = \rho_{AB}^j + c(\delta t_{AB}) + w_{AB}^j \quad (19)$$

La ecuación (19) relaciona las medidas de dos receptores y

dos satélites. A esto se denomina diferenciación doble (DD).

$$\begin{aligned} R_{AB}^{ij} &= R_{AB}^i - R_{AB}^j \\ R_{AB}^{ij} &= \rho_{AB}^{ij} - w_{AB}^{ij} \end{aligned} \quad (20)$$

A través de esta configuración se elimina errores generados por los relojes de los satélites y receptores.

Se detalla el diagrama de flujo en la Fig. 4 del algoritmo desarrollado para la ejecución de DGPS en configuraciones simple y doble.

E. Error cuadrático medio

El Mean Square Error (MSE) [9] se utiliza para evaluar el rendimiento de un algoritmo de posicionamiento. Al considerar que se estima una posición en n veces, de modo que se tiene una población de coordenadas calculadas, $(x_1, y_1), \dots, (x_n, y_n)$. La media de las coordenadas calculadas es:

$$\begin{aligned} \bar{x} &= \frac{1}{n} \cdot \sum_{k=0}^n x_k \\ \bar{y} &= \frac{1}{n} \cdot \sum_{k=0}^n y_k \end{aligned} \quad (21)$$

donde x y y son las coordenadas estimadas de la posición.

La ecuación de error cuadrático medio de la posición estimada está definida por:

$$MSE = (\bar{x} - x)^2 + (\bar{y} - y)^2 \quad (22)$$

El resultado de la ecuación anterior es una expresión cuadrada de error, para una mejor interpretación del error el Root Mean Square Error (RMSE) [9] representa la desviación estándar de la muestra de las diferencias entre los valores estimados y los valores observados. Estas diferencias individuales se llaman residuos de los cálculos ejecutados sobre la muestra de datos utilizados para la estimación y, se llaman errores de predicción cuando se calcula fuera de la muestra. El RMSE sirve para agregar las magnitudes de los errores en las predicciones varias veces en una sola medida de poder predictivo. RMSE es una medida de precisión. Para comparar los errores de predicción de los diferentes modelos para un determinado dato y no entre conjuntos de datos, ya que depende de la escala [13].

$$RMSE = \sqrt{(\bar{x} - x)^2 + (\bar{y} - y)^2} \quad (23)$$

III. METODOLOGÍA

El procedimiento a seguir para el desarrollo del algoritmo de DGPS consta de etapas descritas en la Fig.5

A. Obtención de datos brutos

Se configura el receptor móvil mediante el software U-center [5] para la extracción de datos brutos, para ello es necesario instalar el parche en la memoria volátil para la activación de la lectura de datos brutos a través de los mensajes UBX "RXM-RAW" y "RXM-SFRB" propios del dispositivo ublox. Ver Fig. 7.

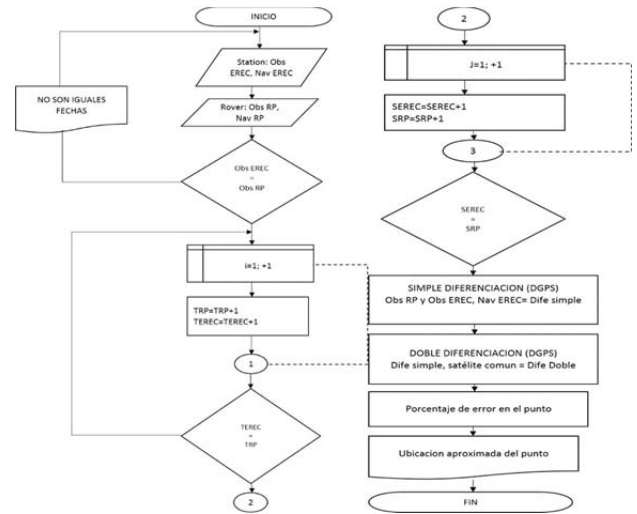


Fig. 4. Diagrama flujo del algoritmo DGPS

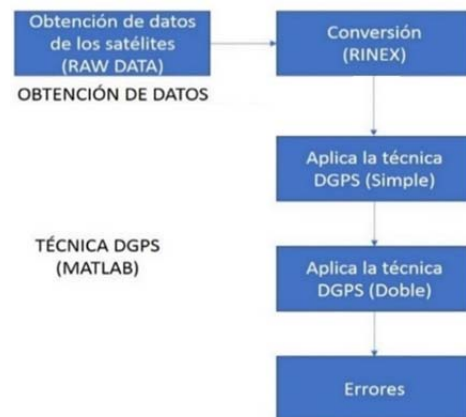


Fig. 5. Diagrama de bloques del algoritmo de DGPS: Extracción de información de las medidas pseudorange de los archivos de navegación y observación de los receptores para la diferenciación en configuración simple y doble.

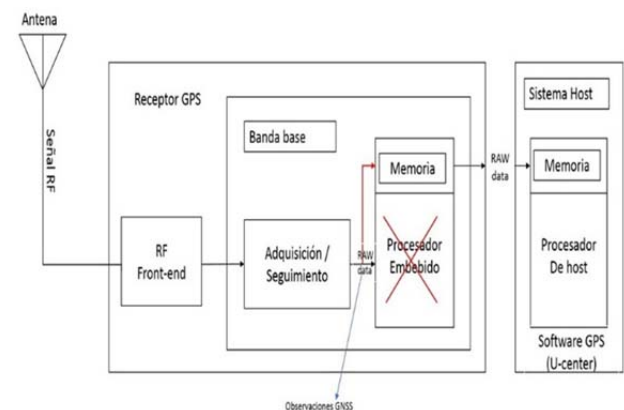


Fig. 6. Señal de GPS en datos brutos: Proceso de recepción de señal de GPS en el rover u-blox NEO 6M, se prescinde de la utilización del procesador propio debido al procesamiento hecho por la computadora.

```

For ublox6 ROM7.03 use:
to enable RXM-RAW
b5 62 09 01 10 00 c8 16 00 00 00 00 00 97 69 21 00 00 00 02 10 2b 22

to enable RXM-SFRB
b5 62 09 01 10 00 0c 19 00 00 00 00 00 83 69 21 00 00 00 02 11 5f f0

```

Fig. 7. Comando de activación de datos brutos: La secuencia de código activa la lectura y extracción de los datos brutos en el receptor.

Una vez parchado el receptor móvil, es necesario configurar el puerto de comunicación para la lectura y almacenamiento de los datos brutos a través de RTKNAVI que a través de la interfaz gráfica se visualiza los datos recibidos en términos de latitud, longitud, intensidad de señal y tipo de posicionamiento.

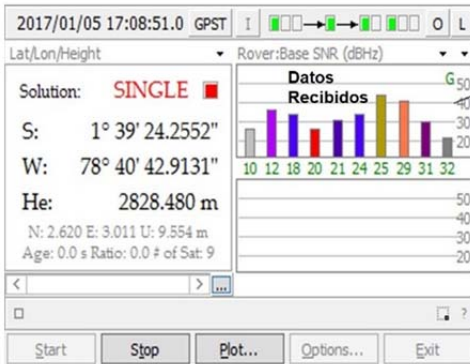


Fig. 8. GPS Estatus: La aplicación RTKNAVI se muestra la intensidad de las señales de los satélites en vista para el posicionamiento al igual que la latitud, longitud y tipo de posicionamiento

El procesamiento de la información de navegación es extraído de los datos brutos obtenidos de los receptores GPS, se trabaja bajo el formato estándar RINEX. Los dispositivos u-blox neo 6M [5] no son capaces de almacenar los datos brutos y presentarlos en este formato, como se mencionó en la Sección II.B. La conversión entre formatos se realiza a través de RTKCONV que al igual que RTKNAVI son subaplicaciones de RTKlib.

B. Escenario y Análisis

La correlación de los errores depende de la distancia que exista entre receptores, en la recolección de datos se considera escenarios lejanos y cercanos respecto al receptor de referencia y afectados por factores de densidad de vegetación y edificios.

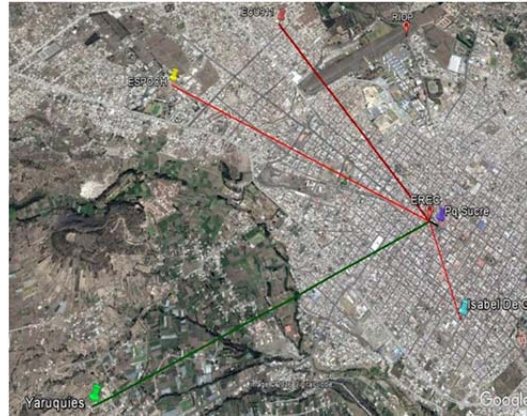


Fig. 9. Sectores urbanos de la ciudad Riobamba: Se considera cuatro sectores a distancias deferentes respecto a la estación de referencia.

TABLA I
PUNTOS DE OBSERVACIÓN RESPECTO A LA ESTACIÓN DE REFERENCIA

Lugar	Distancia [km]	Tiempo de Observación [min]
Isabel de Godin	0,94	136
ECU911	2,92	43
ESPOCH	3,44	65
Yaruquies	3,72	204

IV. RESULTADOS

Al aplicar el algoritmo de DGPS en post-proceso en los diferentes puntos de recolección de datos, se valora la precisión que alcanza en comparación al posicionamiento estándar [1].

Los resultados obtenidos al aplicar el diferencial GPS en configuración simple y doble de los puntos de observación elegidos dentro de la ciudad de Riobamba en el sistema de coordenadas Norte y Este (NE) que permite la visualización de latitud y longitud de los puntos escenarios propuestos de los cuales se presenta dos de ellos en diferentes condiciones. Ver Tabla I.

A. Punto de Observación: ESPOCH

Se tomaron los datos en el sector de la ESPOCH, en el parque del estudiante y a una distancia de 3.44 km desde el receptor de referencia (EREC), mediante el método de posicionamiento estático y el diferencial en configuración simple y doble.

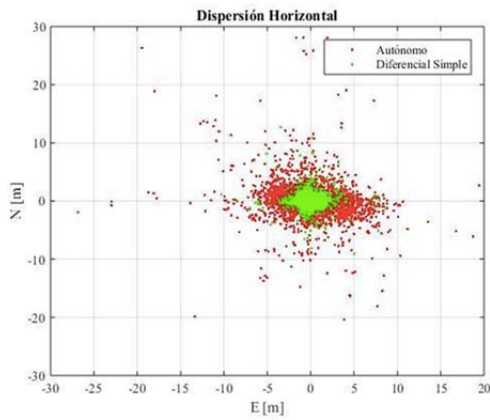


Fig. 10. Diferencial simple - ESPOCH: Comparación entre las medidas de posicionamiento autónomo y diferenciación simple.

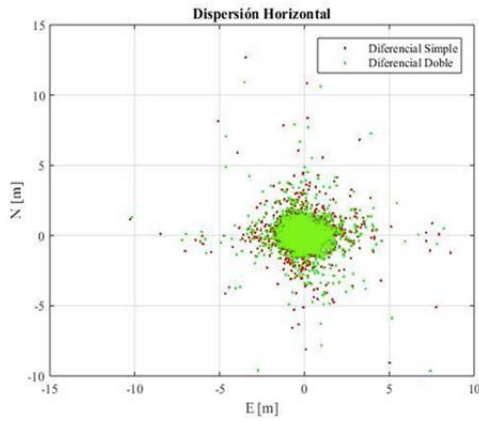


Fig. 11. Diferencial doble - ESPOCH: Comparación del error entre diferencial simple y doble.

TABLA II
ERROR DE POSICIÓN – ESPOCH

	Autónomo	Diferencial Simple	Diferencial Doble
RMSE	3.3869	1.0317	1.0163
Desviación Estándar	2.9905	0.9770	0.9614
Máximo	32.7412	13.1476	12.1553
Mínimo	0.0193	0.0102	0.0166
Rango	32.7219	13.1374	12.1388
Épocas	3280	3280	3280

Las épocas totales en este punto fueron de 3900 que representa la dispersión total durante 65 min. El posicionamiento autónomo muestra un error 3.3869 m, al aplicar el algoritmo DGPS en diferencial simple el posicionamiento se alcanza a 1.0317 m lo que produce una reducción a través de esta técnica 2.3552 m que favorece a la reducción del 69.54% a través de la eliminación de errores comunes. En configuración doble la reducción del error alcanza a 2.3706 m que equivale al 69.99% con respecto a la media. El aporte de DD sobre SD es de 0.0154 m lo que evidencia que en el diferencial doble la reducción de los errores debidos a los relojes del satélite y del receptor son pequeños al contrario del diferencial simple que se reduce más de la mitad el error. Ver Tabla II.

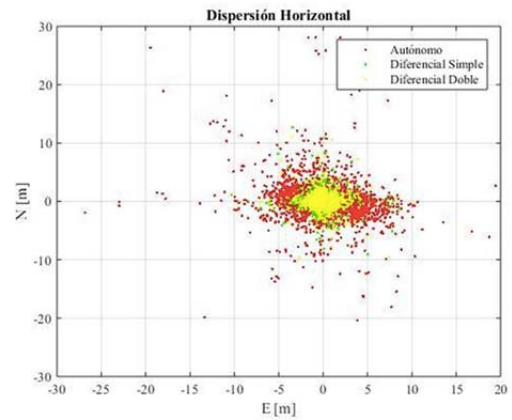


Fig. 12. Solución de posicionamiento final: Comparación de medidas sin corrección y con diferenciación simple y doble – ESPOCH

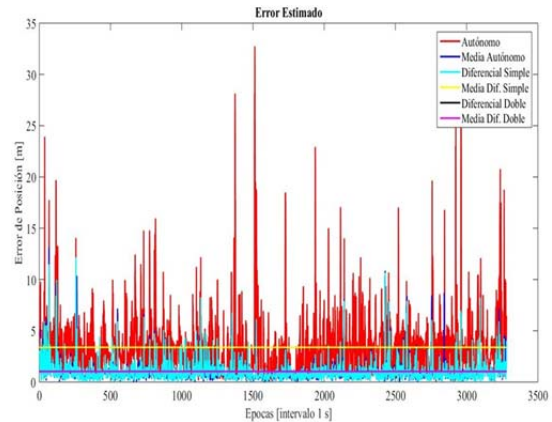


Fig. 13. Error medio estimado en modos autónomo y diferencial – ESPOCH

B. Punto de observación: Yaruquíes

Yaruquíes es una parroquia urbana ubicada en las calles vía a Cacha y 12 de Julio alejada de 3.72 km desde la estación de referencia (EREC). Los datos son medidos al utilizar el método de posicionamiento estático y el diferencial en configuración simple y doble.

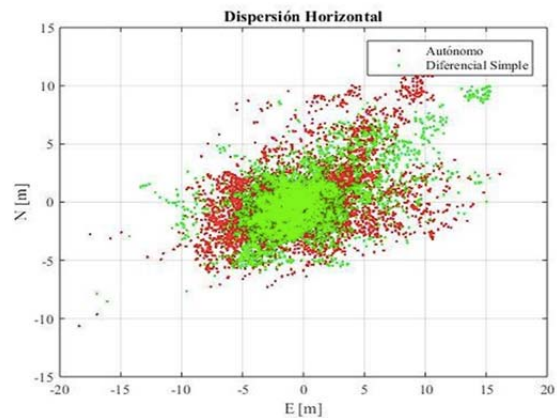


Fig. 14. Diferencial simple - Yaruquíes: Comparación entre las medidas de posicionamiento autónomo y diferenciación simple.

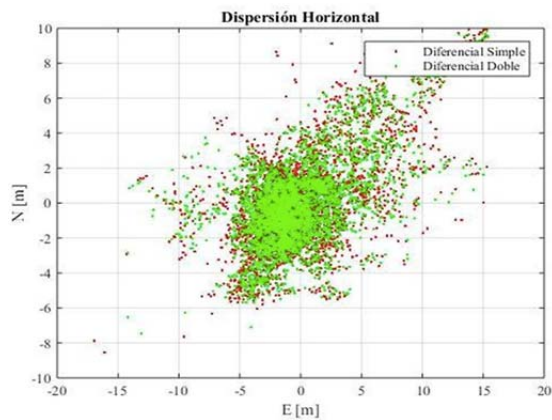


Fig. 15. Diferencial doble - Yaruquíes: Resultado de la reducción de la doble diferencia respecto a la simple

TABLA III
ERROR DE POSICIÓN - YARUQUÍES

	Autónomo	Diferencial Simple	Diferencial Doble
RMSE	4.2473	3.8574	3.7421
Desviación Estándar	2.6839	2.9450	2.8723
Máximo	21.2546	18.7092	18.1696
Mínimo	0.0512	0.1037	0.0581
Rango	21.2034	18.6055	18.1115
Épocas	6098	6098	6098

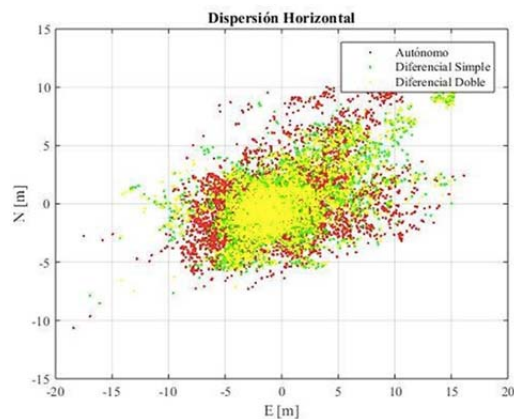


Fig. 16. Medidas Autónomo, Simple y Doble – Yaruquíes

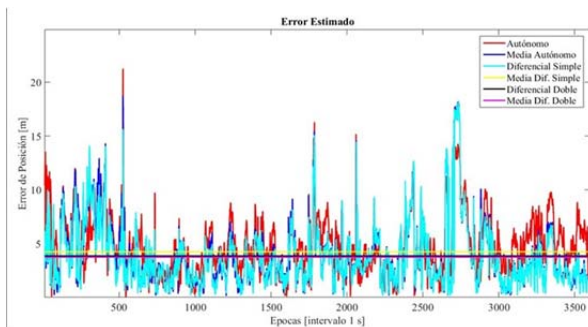


Fig. 17. Error medio estimado en modo autónomo y diferencial - Yaruquíes

Las 12240 épocas medidas durante 204 min permite evaluar la precisión alcanzada por el modo de posicionamiento autónomo de valor 4.2473 m, al aplicar el algoritmo DGPS la corrección de error en el diferencial simple es de 0.3899 m lo

que mejora su precisión en 9.18%. El diferencial doble muestra una reducción de error de 0.5052 m que reduce el error en un 11.89% con respecto a la media. Debido al ambiente de vegetación densa se produce errores de multipath los cuales no están correlacionados y son difíciles de eliminar. Ver Sección I.

C. DGPS-LAB

DGPS-LAB es el software de postprocesamiento para la eliminación de errores en las medidas del receptor GPS, es creado para ser utilizado como herramienta de postproceso para la corrección de errores y soporta las configuraciones de diferencial simple y doble basado en las medidas de pseudorange. Posee una interfaz gráfica que permite cargar los archivos RINEX de la estación de referencia y del rover para su posterior procesamiento y visualización de resultados. Su utilización está limitada para fines educativos y de verificación de las técnicas en las diferentes configuraciones, no se recomienda su utilización para aplicaciones críticas. El algoritmo soporta la versión RINEX 2.11.

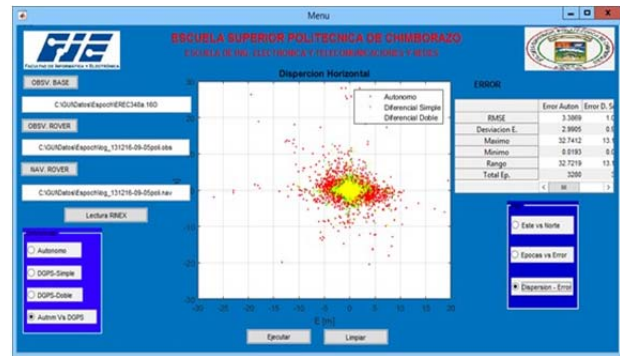


Fig. 18. Interfaz gráfica de DGPS-LAB

V. CONCLUSIONES

Los receptores de bajo costo debido a factores como su arquitectura, capacidad de procesamiento y tecnología, brinda baja precisión por lo que no son ideales para aplicaciones específicas de georeferenciación. Con la implementación del diferencial de GPS en configuración simple y doble se determinó que la reducción del error respecto al posicionamiento estándar es mayor esto se debe a que existe errores correlacionados entre el receptor de referencia y el usuario y permite la eliminación de errores de gran magnitud producidos principalmente por la atmósfera y tropósfera. DGPS-LAB es una aplicación que permite cargar archivos RINEX versión 2.11 de cualquier receptor de GPS y postprocesarlo para su posterior análisis de errores. En los receptores de bajo costo se debe considerar cambiar la frecuencia de recepción de datos por defecto a 1 Hz para de esta manera forzar al dispositivo a adquirir los datos cada segundo e igualarlo a la misma frecuencia de adquisición de datos del receptor de referencia con la finalidad de evitar procesamiento innecesario y la saturación del programa. En este trabajo se determinó que existe un desfase en el intervalo de registro de datos que presentó el dispositivo U-blox NEO 6M fue de 0.9 s registrado en el fichero RINEX. Los

escenarios propuestos fueron considerados debido a sus condiciones de observación y toma de datos basados en los resultados, (Ver Subsecciones IV-C y IV-B), se determina que la señal de GPS en mejor calidad para su recepción se recomienda se realice en días despejados, zonas de poca densidad de vegetación y edificios para valorar el impacto de todos los factores externos y de su influencia sobre la precisión en el posicionamiento de un rover.

DGPS-LAB está diseñado para fines educativos y de verificación de las técnicas en las diferentes configuraciones basadas en las medidas de pseudorange, en orden de alcanzar mayor precisión se optaría por la diferenciación por fase y doble frecuencia y tomar como referencia el proceso descrito. Verificar la versión del archivo de lectura ya que está limitado a la versión 2.11.

RECONOCIMIENTOS

La realización del presente trabajo fue posible gracias a la colaboración de la Empresa Eléctrica Riobamba S.A., a través de la autorización para el acceso a los datos brutos de la estación de referencia (EREC), permitiendo así llevar a cabo este tema de desarrollo e implementación.

REFERENCIAS

- [1] Consentino, R.J. Diggie D.W. "Understanding GPS Principles and Applications", 2nd Edition, Boston London, Artech House Publishers, 2006.
- [2] J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares, GNSS Data processing. Volume 1: Fundamentals and Algorithms. ESA TM-23/1. ESA Communications, May 2013.
- [3] Red GNSS de Monitoreo Continuo del Ecuador - REGME, Noviembre 2017. [Online]. Disponible en: <http://www.geoportaligm.gob.ec/wordpress/>.
- [4] Trimble NetR9 GNSS Reference Receiver, Diciembre 2017. [Online]. Disponible en: <http://www.trimble.com/Infrastructure/Trimble-NetR9.aspx>.
- [5] NEO-6 series Versatile u-blox 6 GPS modules, Diciembre 2017, [Online]. Disponible en: <https://www.u-blox.com/en/product/neo-6-series>.
- [6] National Marine Electronics Association, Enero 2017. [Online]. Disponible en: <http://www.nmea.org>
- [7] M. Zabala, "Determination of monitor threshold for a GBAS ground station". M.S. thesis, Telecom Lab / Signal Processing Department, École Nationale de l'Aviation Civile. Toulouse, Francia, 2015.
- [8] Takasu, T. RTKLIB: An Open Source Program Package for GNSS Positioning. [Online]. Disponible en: <http://www.rtklib.com/>.
- [9] Steven M. Kay, "Fundamentals of Statistical Signal Processing Estimation Theory, Signal Processing Series", Volume I, 1st Edition, PTR Prentice Hall, 1993.
- [10] Steven M. Kay, "Fundamentals of Statistical Signal Processing Detection Theory, Signal Processing Series", Volume II, 1st Edition, PTR Prentice Hall, 1993.
- [11] López F., Ortega A., "Análisis e Implementación del diferencial de GPS con configuración simple y doble, para mejorar la precisión de un receptor GPS en la ciudad de Riobamba", Tesis de ingeniería, Escuela de Ingeniería en Electrónica, Telecomunicaciones y Redes, Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador, Tech. Report., 2017.
- [12] Borre, Kai and Strang, Gilbert. "Algorithms for Global Positioning",., Wellesley-Cambridge Press, 2012.
- [13] J. Sanz Subirana, J.M. Juan Zornoza, M. Hernández-Pajares, GNSS Data processing. Volume 2: Laboratory Exercises. ESA TM-23/2. ESA Communications, May 2013.



Departamento de Eléctrica y Electrónica
Universidad de las Fuerzas Armadas ESPE
<http://maskay.espe.edu.ec/>

LLAMADO A PRESENTACIÓN DE ARTÍCULOS

La Universidad de las Fuerzas Armadas ESPE y el Departamento de Eléctrica y Electrónica, conscientes de su responsabilidad social y con el objetivo de impulsar iniciativas tendientes a promover el desarrollo integral de la Ciencia y Tecnología en el Ecuador, invitan a investigadores y estudiantes a participar en la próxima edición de la revista *Maskay* (Junio/2018). Para ello, la revista pone a disposición de la comunidad científica su plataforma en-línea para la recepción de trabajos sobre investigación, desarrollo e innovación en las siguientes áreas:

- Telecomunicaciones
- Automatización y control
- Redes y comunicación de datos
- Electromagnetismo y óptica aplicada
- Procesamiento de señales
- Sistemas digitales
- Sistemas electrónicos
- Sistemas eléctricos

Información para el Envío de Artículos

Los artículos pueden ser escritos en español, portugués o inglés, y deben ser enviados en formato PDF a través de la plataforma de la revista. Los trabajos presentados deben incluir resultados originales o reportar experiencias relevantes y no deben estar publicados ni en proceso de evaluación en otras revistas o conferencias técnicas. Todos los artículos aceptados, después de la revisión por pares correspondiente, serán publicados en la revista *Maskay* que se encuentra indexada en *Latindex*, *SciELO-Ecuador*, *LatAmPlus*, *RootIndexing*, *MIAR* y *Directory of Open Access Journals (DOAJ)*.

La extensión máxima de los artículos es 10 páginas y el estilo de los mismos debe respetar el formato de la revista. El archivo de estilo \LaTeX y el *template* para *Microsoft Word* están disponibles en el sitio Web de la revista.

Fechas Importantes número Junio/2018

Fecha máxima de postulación: 01/04/2018*

* Los artículos serán procesados conforma sean subidos a la plataforma. Luego del proceso de revisión los artículos serán publicados inmediatamente en sitio Web de la revista.

Comité Editorial

- Diego Arcos Avilés (ESPE-Ecuador)
- Diego Benítez (USFQ-Ecuador)
- Enrique V. Carrera (ESPE-Ecuador)
- Ana Guamán (ESPE-Ecuador)
- Daniel Ochoa (ESPOL-Ecuador)
- Lauro Ojeda (UMich-USA)
- Martha Cecilia Paredes (EPN-Ecuador)
- Vinicius Petrucci (UFBA-Brasil)
- Juan Pablo Robelly (Rohde & Schwarz)
- José Luis Rojo-Álvarez (URJC-España)
- Carlos Julio Tierra (UFRJ-Brasil)
- Alexis Tinoco (ESPE-Ecuador)

Mayor Información

Para obtener mayor información sobre la revista *Maskay* o remitir artículos para su evaluación, por favor visitar el sitio Web de la revista o contactarse con su Editor General: *Diego Arcos Avilés* (dgarcos@espe.edu.ec, +59323989400 ext. 1867).

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

Universidad de las Fuerzas Armadas – ESPE

El Departamento de Eléctrica y Electrónica (antigua Facultad de Ingeniería Electrónica) fue creado el 25 de abril de 1977 e inicia sus labores en octubre del mismo año en la modalidad presencial, con sus planes y programas de estudio encaminados a la formación del Ingeniero Electrónico. A partir de octubre de 1990, ofrece dos carreras con perfiles profesionales definidos: Ingeniería Electrónica en Telecomunicaciones e Ingeniería Electrónica en Computación.

En la propuesta actual del Departamento se integran en un sólo perfil estas aspiraciones, ofreciendo las carreras de Ingeniería en Electrónica y Telecomunicaciones, e Ingeniería en Electrónica, Automatización y Control.

Además oferta los siguientes programas de posgrado: Maestría en Redes de Información y Conectividad, y Maestría en Gerencia de Redes y Telecomunicaciones. En la actualidad, el DEEE se encuentra diseñando los programas de Maestría en Ciencias y Doctorado en Ciencias en Ingeniería Eléctrica.



**Departamento de Eléctrica y Electrónica
Universidad de las Fuerzas Armadas ESPE
Copyright © 2017**