

Artículo de Investigación/Research Paper

Sistema de Prototipado Rápido de Control para una Planta Didáctica Motor DC

Rapid Control Prototyping Platform for Didactic Plant Motor DC

Cristian Bazán-Orobo¹
Juan F. Flórez-Marulanda²

Fecha de recepción: 18 de septiembre de 2012
Fecha de aceptación: 24 de junio de 2013

1 Ing. Físico, Universidad del Cauca,
Popayán-Colombia
bazan@unicauca.edu.co

2 Ing. en electrónica y telecomunicaciones,
MSc. en Electrónica, Departamento de Electrónica,
Instrumentación y Control,
Universidad del Cauca, Popayán-Colombia
jflorez@unicauca.edu.co

Resumen

El artículo presenta el diseño, implementación y validación de una plataforma de Prototipado Rápido de Control (o RCP por siglas en inglés) para una planta motor DC. La plataforma RCP consta de hardware y software. El hardware: un PC, tarjeta electrónica, motor *DC* y sensores. El software basado en herramientas libres como *Linux*, *Scilab / Scicos* y *RTAI-Lab*. La plataforma RCP permite desarrollar prácticas de control de velocidad y posición simuladas y reales, usando diferentes tipos de controladores PID industriales y experimentar fenómenos como *Wind Up* y *Bump Transfer* en una plataforma de bajo costo. Se desarrolla una aplicación de control de velocidad organizada en cuatro pasos: identificación, diseño controlador, simulación y control en tiempo real, donde se destacan las ventajas pedagógicas de una plataforma que no solo permite realizar simulación sino también control en tiempo real de una planta motor DC.

Palabras clave

Prototipado Rápido de Control, Control PID, RTAI-Lab, *Wind Up*, *Bump Transfer*.

Abstract

In this paper a design, implementation and validation of a Rapid Control Prototype platform for a plant based on a DC motor is proposed. This low-cost prototype provides of an electronic card (with a motor DC and sensors) manipulated by PC with free software tools using Linux, Scilab / Scicos and RTAI-Lab. This RCP System allows developing speed -position control trainings by using different types of PID industrial controllers with anti – wind up and bump less transfer schemes. We develop a speed control application structured in four steps: identification, controller design, simulation and real time control, where there are pedagogical advantages of a platform that not only allows simulation but also real-time control of a plant.

Keywords

Rapid Control Prototyping, controller PID, RTAI-Lab, *Windup*, *Bump Transfer*.

1. INTRODUCCIÓN

El Prototipado Rápido de Control o RCP (por sus siglas en inglés Rapid Control Prototyping), surgió en la industria automotriz a mediados de 1990, como una solución para el manejo de las complejas tareas del ingeniero de control automotriz (PresicionMBA, 2005). La idea básica es desarrollar y validar nuevos algoritmos de control, con modelos en ambientes simulados, usando software de modelamiento matemático, una vez diseñado y con un resultado aceptable de simulación, se transforma en un prototipo de control en tiempo real, creado automáticamente en lenguaje C, el cual se usa para validar el algoritmo en la planta bajo condiciones reales de operación (Hölttä *et al.*, 2004) (Chen *et al.*, 2010).

El RCP es una variante de Hardware – in – the – loop (HIL), pero es diferente y ampliamente usado para justificar su nombre (PresicionMBA, 2005). En la actualidad el auge de laboratorios remotos, ha demostrado que es un complemento metodológico de los laboratorios o plantas tradicionales de prácticas (Dormido & Torres, 2010). Estas últimas implementadas como sistemas RCP. Existen RCP comerciales tales como: dsPACE (dsPACE GmbH, 2011), Concurrent (Concurrent, 2009), RT-LAB (Opal-RT, 2009), xPCTarget (MathWorks, 2010) y CompactRIO (National Instruments, 2011) pero a altos costos de licencias y equipos, no fácilmente accesibles en la cantidad adecuada a toda institución educativa. Reportes de software comercial desarrollando RCP se encuentran; en (Hong *et al.*, 2000) implementan un algoritmo de procesamiento para una señal digital usando Matlab y el módulo TMS320C30, en (Hercog & Jezernik, 2005) presentan un RCP de un motor basado en DSP utilizando Matlab / Simulink, en (Rajagopal *et al.*, 2008) presentan a LabView RT en aplicaciones de control de movimiento, en (Duma *et al.*, 2010) desarrollan un RCP para Matlab usando un sistema embebido Renesas M32C87 para control de un motor DC, en (Simard *et al.*, 2009) desarrollan un RCP basado en diseño FPGA entre Matlab / Simulink y la tarjeta Amirix AP1000 PCI FPGA, en (Magallan *et al.*, 2012) presentan un RCP para controladores de vehículos eléctricos usando Dymola y Simulink. Otra opción es usar RCP FOSS (*Free and Open Source Software*) como RTAI-Lab, una herramienta para diseño, construc-

ción, ejecución y monitoreo de sistemas de control en tiempo real (Bucher & Dozio, 2003), con reportes como: un RCP basado en plataforma ARM (Skiba *et al.*, 2008), un control en cascada de una planta de nivel (Florez *et al.*, 2008), automatización de una columna de destilación (Espinosa & Pérez, 2009), sintonización e implementación de controladores PID (Duma *et al.*, 2011), sin embargo no todas no están enmarcadas dentro de un contexto académico, documentan y sustentan una planta de pruebas, que se articule con cursos de control industrial y presente guías de prácticas como sucede con las costosas RCP comerciales (Meza *et al.*, 2009).

En este proyecto se presenta el diseño y desarrollo de una plataforma RCP de bajo costo, basada en componentes FOSS que permite realizar prácticas de control de velocidad y posición utilizando controladores industriales P, PI, PD y PID, integrada en un componente hardware constituido por una tarjeta de acondicionamiento electrónico sobre una planta motor DC, facilitando comprender fenómenos no lineales como Wind Up y Bump Transfer (Astrom & Hagglund, 2000).

En la sección Metodología se presentan los componentes hardware y software del RCP, en la sección Resultados y Discusión se diseña y discute la aplicación RCP de control de velocidad y en la sección final se consignan las conclusiones.

2. METODOLOGÍA

La metodología usada en el diseño del sistema RCP consiste en un proceso incremental de creación de prototipos, cubriendo tareas elementales hasta llegar a un prototipo de sistema RCP complejo (Jaramillo & Galíndez, 2011).

2.1 Diagrama Modular Sistema RCP Planta Didáctica Motor DC

El sistema RCP está organizado en dos componentes uno software y otro hardware, ver Fig.1. El software constituido por la plataforma FOSS para RCP RTAI-Lab (Bucher *et al.*, 2008). Esta proporciona un conjunto de herramientas para el diseño, construcción, ejecución y monitoreo de sistemas de control en tiempo real

(Flórez *et al.*, 2008) y el hardware compuesto por un computador personal y una planta didáctica motor DC.

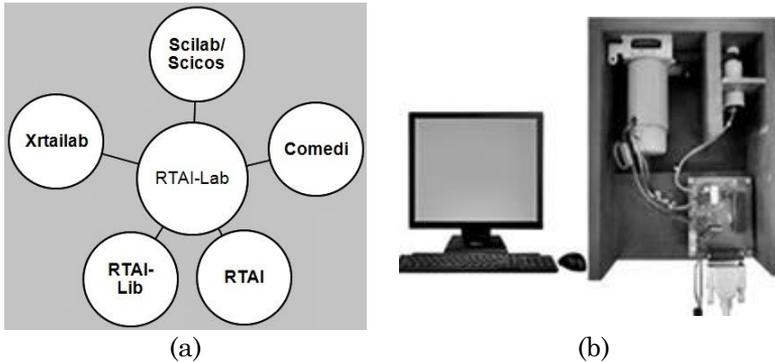


Fig. 1. Componentes sistema RCP Planta Didáctica Motor DC,
a) Software. b) Hardware. Fuente: Autores

El componente hardware comprende una planta didáctica motor DC integrada por un motor DC con accesorios y una tarjeta electrónica. El computador personal tiene una arquitectura 32-bit, procesador Athlon, memoria RAM 512 MB, un disco duro de 40GB con un sistema operativo GNU Linux Fedora 7.

2.2 Componente Hardware

La Fig.2 presenta el diseño del componente hardware del sistema RCP. Este consiste en un PC con RTAI-Lab y la planta didáctica, consistente en un motor de DC y la tarjeta electrónica que realiza la interfaz de comunicación. La tarjeta electrónica usa el puerto de comunicaciones paralelo o LPT como interfaz con el PC. La tarjeta está dividida en módulos: mecánico, comunicación, sensores, actuador planta y el alimentación.

Módulo mecánico: constituido por base en madera, motor DC, encoder incremental unido al motor (como sensor de velocidad) y un potenciómetro lineal (como sensor de posición) acoplado al motor a través de una banda, ver Fig.3.

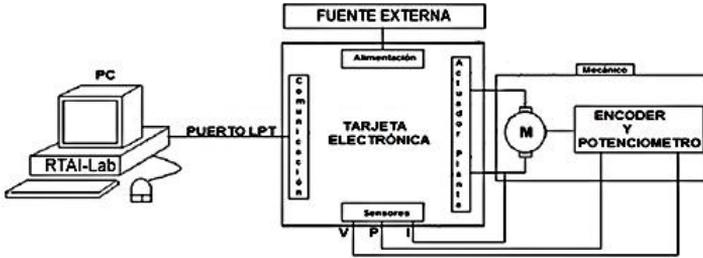


Fig. 2. Sistema RCP Planta Didáctica Motor DC propuesto. Fuente: Autores

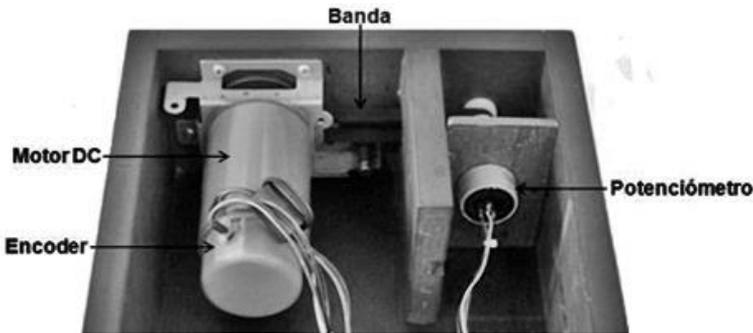


Fig. 3. Módulo mecánico Planta Didáctica Motor DC. Fuente: Autores

Módulo de comunicación: usa los registros de datos y control del LPT para coordinar la comunicación entre RTAI-Lab y la tarjeta electrónica. Esta usa un microprocesador (PIC16F877A) para generar una señal de PWM, digitalizar las variables de corriente y posición y realizar la medición del periodo del tren de pulsos del encoder. Estas tareas se realizan en un periodo de muestreo (PM1) donde: se envía el EC (esfuerzo de control, valor de PWM) y se lee la velocidad, posición y corriente, y así sucesivamente. Como dato es un registro de 8 bits, la lectura de velocidad, posición y corriente se realiza en dos bytes, primero se leen los 8 bits menos significativos y en el segundo los restantes, ver Fig.4.

Entre el registro de datos y control se implementa un protocolo de comunicación, ver Fig. 5, donde datos hace el intercambio de información entre RTAI-Lab y la tarjeta, y control coordina el flujo de información.

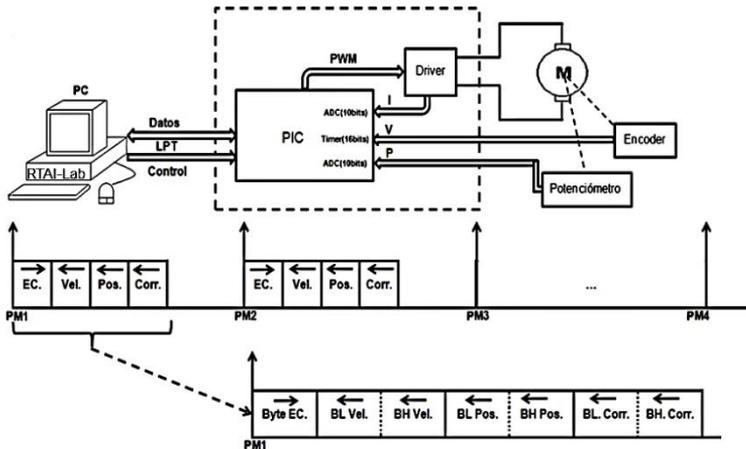


Fig. 4. Escritura y lectura de señales de comunicación en tarjeta electrónica.
Fuente: Autores

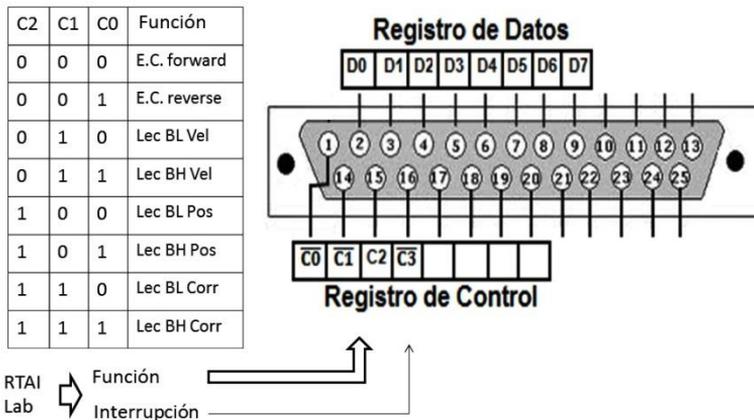


Fig. 5. Puerto LPT y funciones del protocolo de comunicación de la tarjeta electrónica. Fuente: Autores

Módulo de sensores: implementa un sensor de corriente y la electrónica de los sensores de posición y velocidad. El sensor de posición constituido por un potenciómetro lineal de 5 k Ω de una vuelta sin limite mecánico, una etapa de acondicionamiento formada por dos capacitores de 0,1 μ F y un amplificador LM358 que actúa como acoplador de impedancia.

El circuito manejador del motor ofrece la posibilidad de sensar la corriente consumida, usando una resistencia de $0,1 \Omega$ en serie con el motor DC y un circuito de acondicionamiento. Este se hace en dos etapas: filtro pasa bajos de primer orden con frecuencia de corte de $1,6 \text{ Hz}$ y amplificación del voltaje de la resistencia de $0,1 \Omega$, entregando un voltaje entre 0 y 5 V DC para corrientes de 0 a 2 A . El sensor de velocidad incluye un encoder incremental y una etapa de acondicionamiento. El encoder, un par transmisor – receptor de infrarrojos, genera un tren de pulsos de periodo variable proporcional a la velocidad de un disco ranurado que interrumpe un rayo infrarrojo. El disco se acopla al eje del motor DC y entrega 240 Pulsos / Revolución. El acondicionamiento usa una compuerta *Schmitt Trigger* de un $74\text{LS}14$.

Módulo Actuador-Planta: constituido por un motor DC y un CI L298N. Este es un circuito manejador o *driver* para cargas de hasta 46 V DC y 2 A por canal usando tres señales lógicas (cambio de giro y PWM). El motor DC es un modelo D06D304E de Hitachi (38 V DC , $1,9 \text{ A}$ y 3700 RMP).

Módulo de alimentación: este dispone de dos salidas de voltaje, 5 V y 12 V , para la tarjeta electrónica y el motor DC, respectivamente, y previene daños por cortocircuito o inversión de polaridad en la tarjeta.

2.3 Componente Software

En un sistema de control en lazo cerrado, ver Fig. 6, se identifican los módulos: controlador, genera la señal esfuerzo de control (EC) a partir del set-point (SP), la señal realimentada (SR) de la variable controlada (VC) y una ley de control, actuador, transforma el esfuerzo de control (EC) en el fluido energético o variable manipulada (VM) aplicada a la planta, planta, con el proceso a controlar y transmisor, mide y acondiciona la variable de salida de la planta y la realimenta al controlador.



Fig. 6. Sistema de control en lazo cerrado. Fuente: Autores

En el sistema diseñado, las funciones que realizan los módulos de la Fig. 6 están divididos entre los componentes hardware y software RCP, ver Fig. 7. El módulo netamente software es el controlador. El módulo actuador está dividido en tres submódulos, el submódulo software realiza una transformación lineal del EC para que el submódulo PIC genere el PWM que se aplica al motor a través del submódulo L298N. La planta está dividida en dos submódulos, el motor DC y el software que implementa un protocolo de comunicación y el transmisor dividido en tres submódulos, el primer submódulo medición realiza la medida de velocidad, posición y corriente, el segundo realiza un prefiltrado de la corriente con filtro pasa bajos de primer orden y un submódulo software realiza un filtrado de mediana y pasa bajos para las variables medidas y acondiciona la señal de realimentación. Adicionalmente se usan dos submódulos software complementarios: Presentación para la visualización de las variables y Parámetros para configurar los parámetros de control.

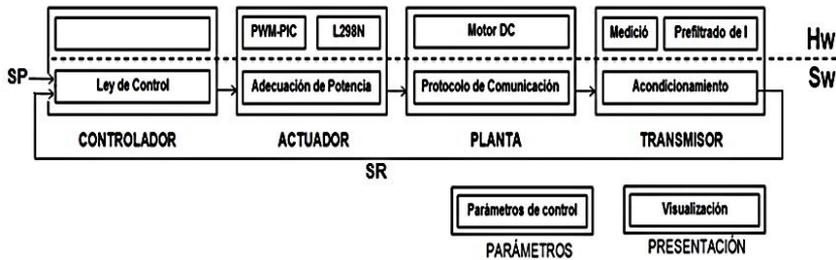


Fig. 7. Distribución de funciones entre los componentes del sistema RCP. Fuente: Autores

Se optó por llamar cada submódulo software con el nombre asociado al módulo que lo contiene. En Scicos estos submódulos se denominan bloques. Para crear bloques en Scicos hay dos formas:

a partir de un bloque ya existente creando bloques normales o súper-bloques, y a partir de la construcción de una función interfaz, escrita en código Scilab y una función computacional escrita en un lenguaje de alto nivel (Delicado, 2003).

La Fig. 8 presenta la paleta de bloques Scicos diseñada para el sistema RCP Planta Didáctica Motor DC. Los bloques construidos son: parámetros, controlador, motor DC (planta), actuador y transmisor. El bloque presentación es provisto por la paleta RTAI-Lib. Los bloques se agrupan en una paleta llamada motor-DC, disponible en Scicos, para realizar RCP de velocidad y posición.

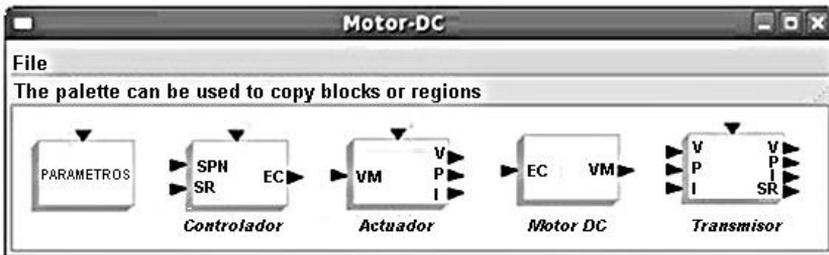


Fig. 8. Paleta Scicos Motor-DC del Sistema RCP. Fuente: Autores

3. RESULTADOS Y DISCUSIÓN

Esta sección detalla el procedimiento usado para la validación del sistema RCP Planta Didáctica Motor DC. El procedimiento RCP propuesto en el trabajo se organiza en cuatro (4) actividades consistentes en:

Actividad 1: identificación y validación de un modelo del proceso que se desea controlar, velocidad o posición, a partir de un método de identificación de la literatura técnica. Actividad 2: selección y sintonización de un controlador, utilizando un algoritmo de control de la literatura técnica, para lograr un comportamiento deseado en el modelo del proceso obtenido en la actividad 1. Actividad 3: diseño y simulación de un sistema de control, con el modelo y controlador obtenidos en las actividades anteriores, mediante diagramas de bloques en Scicos. Actividad 4: implementación de un lazo de control, utilizando la planta real y los resultados del sistema de control diseñado y simulado en la actividad 3,

mediante diagramas de bloques en Scicos ejecutando una tarea de tiempo real. Se presentan resultados del procedimiento RCP aplicado al control de velocidad del motor DC.

Actividad 1: el proceso del control de velocidad es un sistema SISO, entrada voltaje aplicado al motor y salida velocidad del eje. Se modela como un sistema de primer orden más tiempo muerto (o POMTM por sus siglas en inglés) como el descrito en (1) (Gil & Diaz, 2009).

$$G_P(s) = \frac{K_P e^{-\tau m}}{1 + \tau * s} \tag{1}$$

La identificación del modelo POMTM del proceso de velocidad en el motor DC, se realiza usando el método de dos puntos de Alfaro (2001). La Fig. 9, presenta la curva de reacción de la velocidad ante un cambio escalón de voltaje del 20 %. La traza inferior es el EC y la traza superiores la velocidad del motor DC.

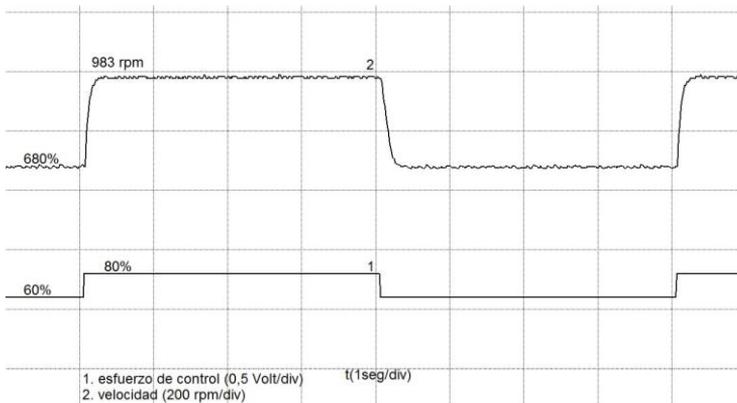


Fig. 9. Curva de reacción del proceso de velocidad visualizado en Xrtailab.
Fuente: Autores

Realizando los cálculos según Alfaro (2001) se obtiene el modelo POMTM presentado en (2):

$$G_P(s) = \frac{5.941e^{-0.016}}{1 + 0.054 * s} \tag{2}$$

La Fig. 10 presenta el diagrama de bloques Scicos diseñado para comparar el modelo POMTM identificado y con el motor DC real. El bloque *Motor DCS* de la Fig. 10 implementa el modelo POMTM.

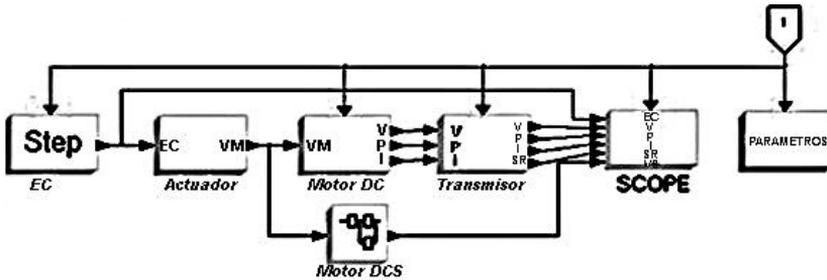


Fig. 10. Diagrama de bloques Scicos para comparar modelo POMTM identificado con Motor DC real. Fuente: Autores

La Fig. 11 presenta la curva de reacción del proceso real de velocidad (traza 3) y la del modelo POMTM (traza 2). La traza 1 representa el EC. La respuesta del modelo POMTM es similar aunque no exacta a la velocidad real del motor DC.

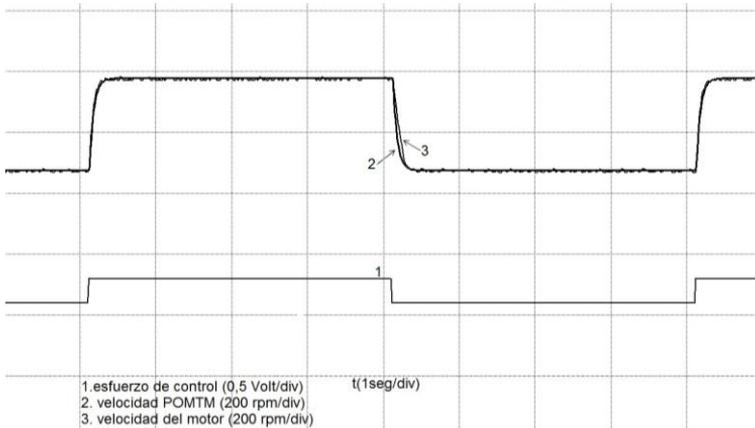


Fig. 11. Respuesta modelo POMTM Vs. Respuesta Motor DC visualizado en Xrtailab. Fuente: Autores

Actividad 2: la sintonización de un controlador depende de los requerimientos del sistema de control, la estructura del controlador y la información del modelo POMTM. Teniendo en cuenta esta información, se debe elegir un método de sintonización válido, tanto para el modelo POMTM, como para la estructura y tipo de PID (Jaramillo & Galíndez, 2011). Para el presente caso, la sintonización del controlador se hace con base en: a) Operación en lazo servocontrolado, b) Controlador PID con filtro derivativo (PID Paralelo Industrial) y AWBT (*Anti- Wind up y Bump Transfer*) y c) Método de identificación POMTM según Alfaro (2001).

La Tabla 1 resume el método de ajuste escogido para sintonizar el controlador PID con filtro derivativo en lazos de control servocontrolado (O'Dwyer, 2009).

Tabla 1. Método de sintonización para lazos de control servocontrolado.
Fuente. Autores

Ecuaciones de sintonización	Estimación de parámetros
Criterio de desempeño:	<i>MinimunIAE</i>
Método de identificación:	Arrieta Orozco - Alfaro Ruiz
Modelo del proceso:	$G_p(s) = \frac{5.941e^{-0.016}}{1 + 0.054 * s}; \frac{t_m}{\tau} = 0.296$
Validez:	$0.05 \leq \frac{t_m}{\tau} \leq 2.0$
PID con Filtro Derivativo	$G_c(s) = K_c \left[1 + \frac{1}{T_i s} + \frac{T_d s}{N s + 1} \right]$
$K_c = \frac{1}{K_m} \left[0.3295 + 0.7182 \left(\frac{\tau}{t_m} \right)^{0.9971} \right]$	$K_c = 0.462$
$T_i = \tau \left[0.9781 + 0.3723 \left(\frac{t_m}{\tau} \right)^{0.8456} \right]$	$T_i = 0.060s$
$T_d = 0.3416\tau \left(\frac{t_m}{\tau} \right)^{0.9414}$	$T_d = 0.006s$
	$N = 10$
$T_t = \sqrt{T_i T_d}$	$T_t = 0.019s$

Actividad 3: se implementa una tarea de control en tiempo real con el modelo POMTM obtenido en el proceso de identificación.

Esta se somete a una serie de eventos para observar el desempeño del mismo, bajo la acción del controlador PID seleccionado y sintonizado en la actividad 2, operando en un lazo de control servocontrolado. La validación se hace en dos pruebas sin y con AWBT:

Prueba 1: se observa la respuesta del modelo ante una serie de conmutaciones en el modo de control (Automático / Manual) con el AWBT desactivado según la secuencia:

- 1) Desactivar protección AWBT.
- 2) Establecer SP en 810 RPM.
- 3) Seis segundos (6s) después establecer SP en 960 RPM.
- 4) Ochos segundos (8s) después conmutar de modo de control automático a manual con EC manual en 0,5.
- 5) Diez segundos (10s) después conmutar de modo de control manual a automático.

La Fig. 12 presenta la respuesta del sistema en lazo de control servocontrolado con protección AWBT desactivada. Donde la traza 1 corresponde al modo de control (Automático / Manual), la traza 2 EC manual, la traza 3 EC automático, la traza 4 SP y la traza 5 VC. Se observa como un cambio abrupto en el SP satura el controlador (traza 3), esto se debe a la saturación del término integral en el controlador. Igualmente, como al conmutar de automático a manual la VC no sigue al SP debido a que el EC manual es menor al EC automático, también se observa como el EC automático (traza 3) no sigue el EC manual (traza 2), saturándose debido a la acción integral. Al conmutar a modo automático se presenta un cambio repentino en el EC aplicado al motor, debido a la diferencia de valores entre el EC manual y el EC automático saturado, que genera un sobre salto de la VC (traza 5), esto es lo que se denomina *wind up* de la VC.

Prueba 2: se observa la respuesta del modelo ante una serie de conmutaciones en el modo de control (Automático / Manual) con el AWBT activado según la misma secuencia de la prueba 1.

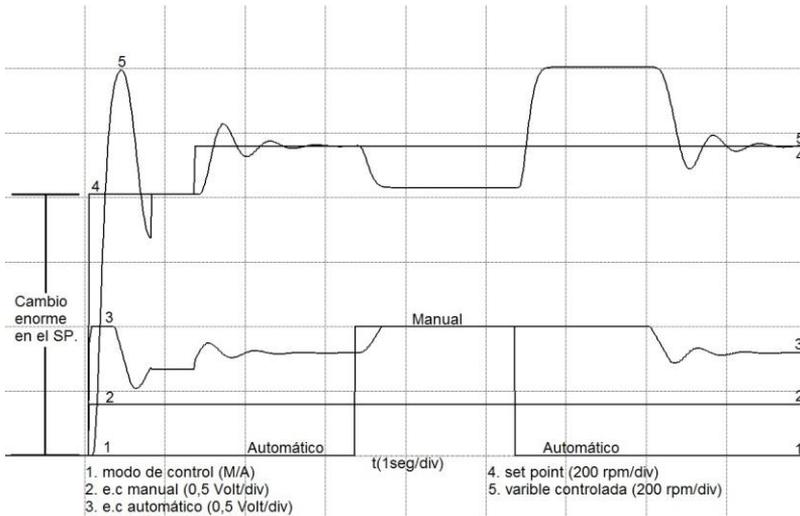


Fig. 12. Respuesta del control de velocidad en el modelo POMTM bajo la acción del controlador PID con la protección AWBT desactivada en Xrtailab. Fuente: Autores

La Fig. 13 presenta la respuesta del sistema en lazo de control servocontrolado con la protección AWBT activada. Se observa como ante el mismo cambio en el SP y la conmutación entre modos de control con la activación del AWBT que se aplica en la prueba 1, el tiempo que el controlador (traza 2) pasa en saturación es mucho menor que en la Fig. 12. Igualmente se observa como al conmutar de modo automático a manual, el EC automático intenta seguir el EC manual, lo que hace que al momento de la conmutación la diferencia entre el EC manual y el EC sea mínima, produciendo igualmente un efecto bajo de *wind up* en la VC. Esto se traduce en un sistema más estable y un menor tiempo de estabilización.

Actividad 4: se implementa una tarea de control en tiempo real con el motor DC, sometido a la misma serie de eventos de la actividad 3.

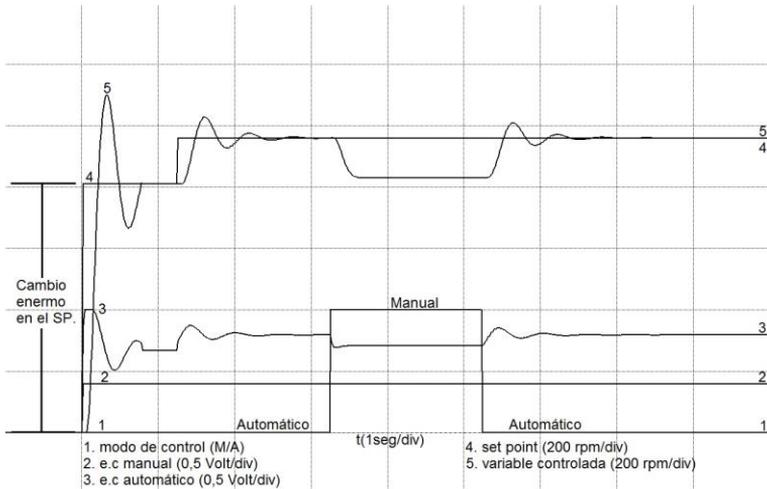


Fig. 13. Respuesta del control de velocidad en el modelo POMTM bajo la acción del controlador PID con la protección AWBT activada en Xrtailab. Fuente: Autores

Prueba 1: la Fig. 14 presenta la respuesta del sistema en lazo servocontrolado a los cambios propuestos en la actividad 3, prueba 1 para el modelo POMTM conmutación auto-manual sin AWBT. Donde la traza 4 es el SP, la traza 5 es la VC, la traza 3 el EC automático, la traza 2 el EC manual y la traza 1 el modo de control (Automático / Manual). Se observa como el comportamiento del sistema real con el AWBT desactivado se comporta de forma idéntica, con leves diferencias, al sistema simulado bajo las mismas condiciones realizadas en la simulación de la prueba 2 de la actividad 3. Esto es, se presenta un cambio abrupto en el SP que satura el controlador, en la conmutación de modos la salida del controlador conmuta al valor en manual, mucho menor del valor en automático, sin embargo el algoritmo PID se satura ya que no tiene el AWBT activado, cuando se conmuta de nuevo a automático, la diferencia de la salida en manual difiere en gran medida del valor saturado, por lo que se presenta el *wind up* en la VC, el cual se elimina después que el controlador en automático contrarresta el área acumulada bajo la curva por la acción integral.

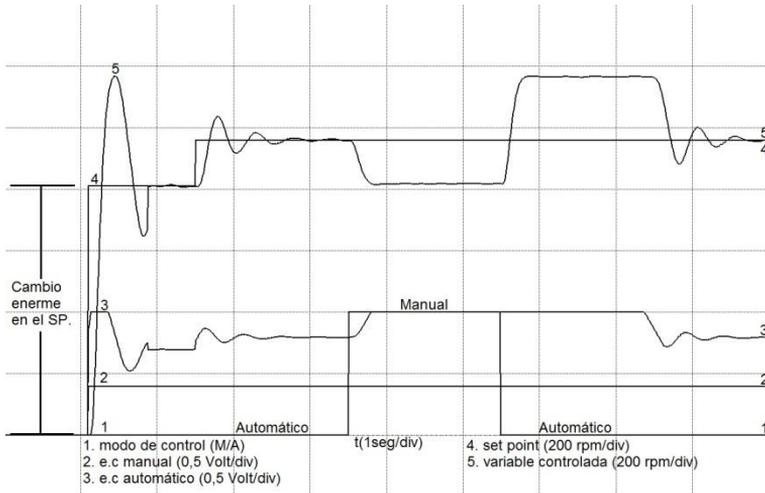


Fig. 14. Respuesta del control de velocidad en el motor DC bajo la acción del controlador PID con la protección AWBT desactivada en Xrtailab. Fuente: Autores

Prueba 2: la Fig. 15 presenta la respuesta del sistema en lazo servocontrolado a los cambios propuestos en la actividad 3, prueba 2 para el modelo POMTM conmutación auto-manual con AWBT.

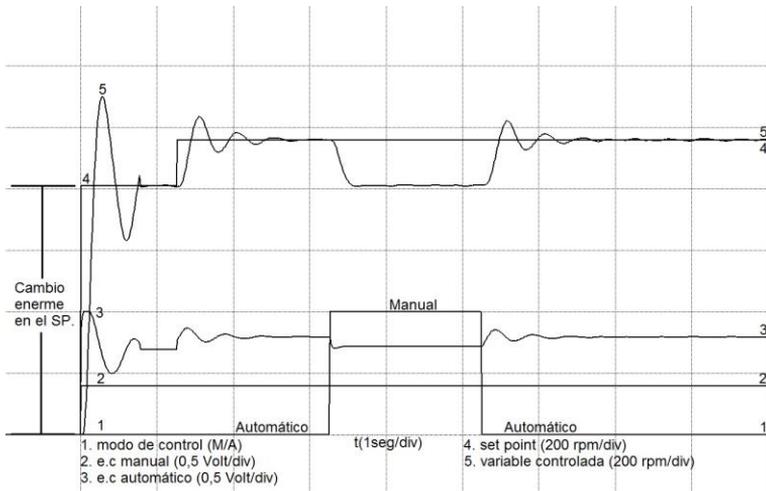


Fig. 15. Respuesta del control de velocidad en el motor DC bajo la acción del controlador PID con la protección AWBT activada en Xrtailab. Fuente: Autores

En la Fig. 15 se observa un comportamiento en la planta real similar, con pequeñas diferencias, a los resultados obtenidos en simulación de la prueba 2 de la actividad 3. Esto es, la saturación del controlador ante un cambio abrupto del SP es leve, en la conmutación de modos de control el EC automático trata de seguir las variaciones del EC manual, por lo que al conmutar de nuevo a automático las diferencias son mínimas y no se presenta *wind up* en la VC, a pesar de una oscilación en la misma que se puede reducir con mejor sintonización del AWBT.

4. CONCLUSIONES

Se diseñó e implementó un sistema de Prototipado Rápido de Control (RCP) FOSS, para un Motor DC basado en las herramientas de software libre RTAI-Lab. Este sistema permite realizar prácticas de control de velocidad y posición siguiendo un procedimiento RCP organizado en cuatro pasos: 1) identificación de un modelo, 2) selección y sintonización del controlador, 3) simulación del prototipo de control y 4) control en tiempo real de la planta. El sistema RCP se compone de una Planta didáctica Motor DC, compuesto de una tarjeta electrónica con comunicación LPT, sensores y manejadores conectados a un motor DC. El sistema se complementa con el aporte de una paleta de bloques en *Scicos* que comprende: controlador PID industrial, planta Motor DC, actuador, Transmisor, entre otros. La cual permite diseñar, simular y ejecutar esquemas de control *feedback* con control PID industrial AWBT configurable en un ambiente gráfico.

Se realizó en el sistema RCP un control de velocidad usando un PID industrial AWBT sintonizado como servocontrol, bajo tres pruebas: variación del SP con AWBT activado, cambio del SP y conmutación A-M-A tanto con AWBT desactivado como activado. Las pruebas se aplicaron en simulación (actividad 3) y con la planta real (actividad 4), los resultados permiten que el estudiante verifique en simulación el comportamiento del sistema y tome correctivos si es el caso, para finalmente ejecutar el control en la planta real. Las Fig. 12 y 14 junto con las Fig. 13 y 15 ilustran el

comportamiento del sistema tanto simulado como real, ante un cambio abrupto del SP y la conmutación A-M-A cuando la protección AWBT se encuentra desactivada y activada, respectivamente. Estas figuras, generadas en tiempo real, permiten que el estudiante experimente las consecuencias del *Windup* y del *Bump Transfer*, y observe que tanto en simulación como en el sistema real el comportamiento es similar, lo que demuestra la importancia y relevancia del sistema RCP como una herramienta de laboratorio en la formación de los ingenieros, pero que se ha conseguido no con costosas herramientas comerciales sino con software libre y componentes electrónicos de bajo costo.

5. REFERENCIAS

- Alfaro, V., (2001). Identificación de Procesos Sobre amortiguados Utilizando Técnicas de Lazo Cerrado, Ingeniería, 11(1-2), 27-40.
- Aström, K.J. & Hägglund T., (2000). Benchmark Systems for PID Control. IFAC Workshop on Digital Control: Past, Present and Future of PID Control, Terrassa, España, 2000.
- Bucher, R., & Dozio, L., (2003). CACSD with Linux RTAI and RTAI-Lab, Proceedings of the Real Time Linux Workshop 2003, Valencia, España.
- Bucher, R., Mannori, S., & Netter, T., (2008). RTAI-Lab tutorial: Scilab, Comedi, and Real-TimeControl. Recuperado el 26 junio del 2012, de <https://www.rtai.org/RTAILAB/RTAI-Lab-tutorial.pdf>
- Chen, X.M., Gong, X.L., Zhou, H. X., Xu, Z. B., Xu, Y. G., & Kang, C.J., (2010). An Economical Rapid Control Prototyping System Design with Matlab/Simulink and TMS320F2812 DSP, In Proceedings of the International MultiConference of Engineers and Computer Scientists, 2.
- Concurrent, (2009). Concurrent Technology Partners. Recuperado el 31 de Julio de 2012, de <http://www.ccur.com>
- Delicado, B., (2003). Introducción al Modelo y Simulación de Sistemas Físicos con Toolbox Scicos de Scilab, Doctorado, Universidad Nacional de Educación a Distancia, Departamento de Informática y Automática, Madrid, España.

- Dormido, S., & Torres, F., (2010). Introducción a la Sección Especial de Laboratorios Virtuales y Remotos en Automática: Realizaciones y Experiencias, *Revista Iberoamericana de Automática e Informática*, 7(1), 5-9.
- dsPACE GmbH, (2011). dSPACE International. Recuperado el 31 de julio de 2012, de <http://www.dspaceinc.com>
- Duma, R., Trusca, M., and Dobra, P., (2010). BLDC Motor Control using Rapid Control Prototyping, In *Journal of Control Engineering and Applied Informatics*, 12, 1, 55-61.
- Duma, R., Trusca, M., and Dobra, P., (2011). Tuning and Implementation of PID Controllers using Rapid Control Prototyping, In *Journal of Control Engineering and Applied Informatics*, 13, 4, 64-73.
- Espinosa, J., & Pérez, O., (2009). Automatización de una Columna de Destilación, Maestría, Centro Nacional de Investigación y Desarrollo Tecnológico, Departamento de Mecatrónica, Cuernavaca, México.
- Flórez, J.F., Díaz, E., & Cabezas, Y., (2008). Simulación y Control en Cascada de una Planta POMTM en Tiempo Real con RTAILab, XIII Congreso Latinoamericano de Control Automático, 16, Mérida, Venezuela.
- Gil, J A., & Díaz, R., (2010). Control de Sistemas Continuos, Fundamentos de Control Automático de Sistemas Continuos y Muestreados, Unico- pia, 42-43, San Sebastián, España.
- Hercog, D., & Jezernik, K., (2005). Rapid Control Prototyping MATLAB/Simulink and a DSP-Based Motor controller, *International Journal of Engineering Education*, 21(4), 596.
- Hong, K. H., Gan, W. S., Chong, Y. K., Chew, K. K., Lee, C. M., & Koh, T. Y., (2000). An Integrated Environment for Rapid Prototyping Of Dsp Algorithms Using Matlab and Texas Instruments Tms320c30- Microprocessors and Microsystems, 24(7), 349-363.
- Hölttä V., Palmroth L. & Eriksson L., (2004). Rapid control prototyping tutorial with application examples, Sim-Serv, www.sim-serv.com.
- Jaramillo, O.F., & Galindez, J.R., (2011). Sistema de Prototipado Rápido de Control para el Módulo De Prácticas Mic955 de la Empresa Feedback, Pregrado, Universidad del Cauca, Departamento de Electrónica, Instrumentación y Control. Popayán, Colombia.
- Magallán G. A., Silva L. I., Angelo C. H., & García G. O., (2012). Rapid prototyping of electric vehicle controllers combining Dymola and Sim-

- ulink, International Journal of Electric and Hybrid Vehicles, 4(2), 197-215.
- Mathworks, (2010); xPCTarget. Recuperado el 31 de julio de 2012, de <http://www.mathworks.com/products/datasheets/pdf/xpc-target.pdf>
- Meza, C. Ayala, Vega. G. & Florez, J., (2009). Latin-American HeDiSC Network: an initiative to promote research collaboration for the use and development of FOSS for control systems, IFAC Workshop on Knowledge and Technology Transfer to Developing Countries.
- National Instruments, (2011). Developing a Rapid Control Prototyping Platform for Implementing Advanced Engine Control Systems. Recuperado el 31 de Julio de 2012, de <http://www.ni.com>
- O'Dwyer, A., (2009). Handbook of PI and PID Controller Tuning Rules, 2, edición 3a, 599, Imperial College Press, London, United Kingdom.
- OPAL-RT Technologies, (2009); RT-LAB Professional. Recuperado el 31 de julio de 2012, de <http://www.opal-rt.com>
- PrecisionMBA, LLC, (2005). Rapid Control Prototyping. Recuperado el 26 junio de 2012, de http://www.precisionmba.com/Public/rapid_controls_prototyping.pdf
- Rajagopal, R., Ramamoorthy, S., Wenzel, Lothar., & Andrade, H., (2008). A Rapid Prototyping Tool for Embedded, Real-Time Hierarchical Control Systems. EURASIP Journal on Embedded Systems.
- Simard S., Mailloux J.G., & Beguenane R., (2009) Prototyping Advanced Control Systems on FPGA. EURASIP Journal on Embedded Systems.
- Skiba G.; Żabinski, T., Bozek A., (2006) Rapid Control Prototyping with Scilab/Scicos/RTAI for PC-based and ARM-based Platforms. International Multiconference on Computer Science and Information Technology, Wisla, Poland.