

COLOMBIAN JOURNAL OF ADVANCED TECHNOLOGIES

RAPID CONTROL PROTOTYPING USING FREE SOFTWARE OF THE
POSITION CONTROL OF A TWO DEGREE OF FREEDOM ROBOTPROTOTIPADO RÁPIDO EN CONTROL EMPLEANDO SOFTWARE LIBRE DE
LA POSICIÓN DE UN ROBOT DE DOS GRADOS DE LIBERTAD

PhD. José Gabriel Hoyos Gutiérrez, PhD. Jaiber Evelio Cardona, PhD. Pablo
Andres Muñoz, MSc. Diego Ramirez

Universidad del Quindío

Armenia, Quindío, Colombia.

Tel.: 57 -6- 7359300 , Ext. 344

E-mail: {josegabrielh,jaibercardona,pamunoz,dframirez}@uniquindio.edu.co

Abstract: A Rapid control prototyping of the angular position control of a two degree of freedom robot are presented, it was implemented in the Scilab/Scicos Software, X2C library and a DSP development board. The control strategy used was a double loop for a each degree, velocity as inner loop and position as outer loop. The model of the robot, as well as the electronics and programs used are all described, error responses and some conclusions are also shown in this paper.

Keywords: *Embedded computer control systems and applications, rapid control prototyping, Control education using laboratory equipment.*

Resumen: Se presentan los resultados del prototipado rápido en control de la posición angular de un robot de dos grados de libertad. Este fue implementado en el programa Scilab/Scicos, la librería X2C y una tarjeta de desarrollo de DSP, la estrategia de control de posición es un control de doble lazo por cada articulación, un lazo interno de velocidad y uno externo de posición. El modelo del robot, la electrónica y programas empleados son descritos, también se muestran las respuestas del error y algunas conclusiones.

Palabras clave: *Computadores empotrados para control y aplicaciones, prototipado rápido en control, equipos de laboratorio para educación en control.*

1. INTRODUCCION

El prototipado rápido para control (RCP de sus siglas en inglés Rapid Control Prototyping) (Hanselmann, 1999) trata sobre las técnicas que permiten el desarrollo de la estrategia de control para una aplicación o planta de laboratorio de manera rápida y sencilla, para lo anterior se aprovechan elementos de hardware y software previamente desarrollados. En RCP no es necesario codificar programas en lenguajes como 'c' y manejar directamente los puertos del procesador que posea la plataforma o tarjeta de desarrollo, si no que de manera gráfica se realiza un esquema y luego con este se programa directamente la plataforma de desarrollo. Para la programación gráfica se emplea Scilab/Scicos.

Scilab es un programa de uso libre para facilitar el cálculo matemático, fue creado en 1990 en el Instituto de Investigación Nacional de Informática y Automática de Francia (INRIA del francés Institut National de Recherche en Informatique et en Automatique), Scicos es el entorno de programación gráfica por diagramas de bloques del Scilab, el cual también se puede emplear de manera gratuita. Como desventaja en comparación con su similar comercial Matlab es que no posee tantas librerías desarrolladas, pero por otro lado existe una comunidad trabajando en fortalecerlo con más librerías.

En este trabajo, el prototipado rápido fue desarrollado en Scicos y una librería llamada X2C (Grabmair, 2014), el cual genera código en 'c' que se descarga a una tarjeta de desarrollo para un microcontrolador a 32 bits de Texas, el cual se describe más adelante y módulos de electrónica de potencia BTS-7960 que acopla los moto-reductores con encoder que mueven las articulaciones de un robot de dos grados de libertad. Si bien este proyecto es una aplicación en robótica, es posible emplearlo en otros procesos como temperatura, nivel, velocidad.

Un RCP tanto para el control de articulaciones como para la ubicación inicial y estados de un robot simulado de seis ejes, fue presentado en (Chen, 2004). Emplearon Matlab/Simulink y un PC con tarjeta de adquisición como plataforma de descarga del código. Como conclusiones es que ellos tuvieron que adaptarse de la programación de código en texto a la diagramación de bloques en RCP para el diseño de estrategias de control.

En (Ramos, 2005) se presentó el control de la planta de presión FeedBack PTC14 mediante el RCP para un DSP TMS320C31, para lo cual emplearon Scilab/Scicos. Los autores concluyen en resumen que con sistemas embebidos fue posible realizar tiempo real hardware a bajo costo.

(Jang, 2008) presentaron una aplicación de RCP en robot móviles que juegan fútbol. Para la programación de la interfaz gráfica que permite el diseño del RCP emplearon una herramienta propia de su Universidad llamada SIMTool. El robot fue programado para planear rutas hacia la pelota y pegarle, también para esquivar rivales, los programas fueron realizados en un PC con windows y probados en un simulador de fútbol robótico llamado SimuroSot.

(Lapusan, 2008) expusieron un sistema de RCP para el control angular de un robot paralelo planar con Matlab/Simulink y la plataforma dSpace, este fue construido con servomotores a los que se les removió la tarjeta de control y se empleó el motor y el sensor. Además de la planta real, también implementaron los modelos cinemáticos y dinámicos. Tanto los modelos como el algoritmo RCP están bien descritos por lo que es posible su replicación. Mediante las respuestas al seguimiento de una trayectoria comprobaron el correcto funcionamiento de la tecnología propuesta.

Un sistema hardware-software de Prototipado rápido para la programación de tareas realizables

por robots simples fue presentado en (Hwang, 2011), el sistema es didáctico y fue empleado en pregrado de ingeniería para la enseñanza del curso de sistemas embebidos orientados a robótica. Su programación gráfica es por bloques, parecido a diagramas de flujo, esta programación desde un display táctil embebido en el mismo sistema RCP o desde un PC. Comentaron además dos proyectos finales de los estudiantes, el primero permite la programación de un robot humanoide para que ante sonidos captados con un micrófono, se mueva con el ritmo de estos (Baile) y el segundo es un robot móvil que colabora con un brazo robótico para desplazar y ubicar un vaso. Los resultados muestran que los alumnos obtienen mejor desempeño al trabajar en equipo y en la solución de problemas y destrezas profesionales.

En (Kielan, 2014), desarrollaron el prototipado rápido para el control articular de dos de las cuatro articulaciones de un brazo robot didáctico, emplearon Matlab/Simulink con la plataforma dSpace y una interfaz gráfica realizada en C++. El sistema permite el manejo del robot desde internet tanto en modo manual, como en modo automático (ejecución de trayectoria articular programada).

En (Yongqiang, 2015) aplicaron prototipado rápido en control y un script de Matlab para el control angular de un brazo robótico de seis grados de libertad, implementaron en una plataforma PC con tarjeta de adquisición, una técnica para ajustar un controlador mediante aprendizaje iterativo llamada ILC (del inglés iterative learning control), el script de Matlab realizaba el ajuste del controlador iterativo, con Simulink realizaban el control en el espacio angular y llevaban el robot también a su posición de inicio, por lo que los autores lo denominaron control híbrido. Los autores concluyen que se demostró la eficiencia del Matlab/Simulink para resolver problemas complejos y que el proyecto es una oportunidad para que los estudiantes puedan aprender la implementación real de sistemas que involucren programación híbrida, control por aprendizaje iterativo.

En (Flórez, 2017), publicaron el desarrollo de un RCP para el control de un brazo robótico simulado, para lo cual emplearon Simulink y la librería State-flow para programar la secuencia que ejecuta el brazo y la pinza para tomar un libro. Los bloques para la realización de la tarea de tomar el libro fueron descritos en el artículo. El funcionamiento del RCP fue probado con el componente de realidad virtual del Simulink.

Los anteriores trabajos la mayoría emplean software comercial y esquemas de control relativamente sencillos y algunos se quedan en la simulación, por lo cual este trabajo es novedoso.

En lo que resta del documento se presenta la descripción del modelo dinámico del robot, los programas y hardware empleado, la implementación en Scicos y X2C del Control Dinámico Angular en cascada. Por último se presentan las respuestas en el tiempo de los valores de las dos articulaciones y su discusión y las conclusiones.

2. DESCRIPCIÓN DEL MODELO DINÁMICO DEL ROBOT

En la Figura 1, se muestra un dibujo del robot 2D, donde se aprecian las articulaciones $\mathbf{q} = (q_1, q_2)$. La ecuación de torque (Reyes, 2011), (Spong, 2006):

$$\tau = M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q}) + F_f(\mathbf{q}) \quad (1)$$

Definiendo las constantes α, β, γ como :

$$\alpha = I_{z1} + r_1^2 m_1 + l_1^2 m_2 \quad (2)$$

$$\beta = I_{z2} + r_2^2 m_2 \quad (3)$$

$$\gamma = l_1 m_2 r_2 \quad (4)$$

Con m_i, l_i, r_i la masa, longitud y mitad de la longitud del i -esimo enlace.

El momento de inercia para una platina como:

$$I_{zi} = \frac{1}{12} m_i (4l_i^2 + w_i^2) \quad (5)$$

Con w_i el peso del i -esimo enlace.

$$M(\mathbf{q}) = \begin{bmatrix} \alpha + \beta + 2\gamma \cos q_2 & \beta + \gamma \cos q_2 \\ \beta + \gamma \cos q_2 & \beta \end{bmatrix} \quad (6)$$

$$C(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} \gamma \sin q_2 \dot{q}_2 & \gamma \sin q_2 (\dot{q}_1 + \dot{q}_2) \\ \gamma \sin q_2 \dot{q}_1 & 0 \end{bmatrix} \quad (7)$$

$$g(\mathbf{q}) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (8)$$

Despejando $\ddot{\mathbf{q}}$ a partir de la ecuación (1), despreciando la gravedad por la forma de construcción y no teniendo en cuenta el componente de fricción $F_f(\mathbf{q})$:

$$\ddot{\mathbf{q}} = M^{-1}(\mathbf{q}) \{ \tau - C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - g(\mathbf{q}) \} \quad (9)$$

A partir de las ecuaciones (2 a 9) y los datos de la Tabla 1, se implemento el modelo del robot 2D en Scicos y se probaron en simulación dos controladores PD, uno por cada articulación, pero los resultados no fueron buenos, por lo cual se decidió implementar un control en cascada o doble lazo de velocidad y posición.

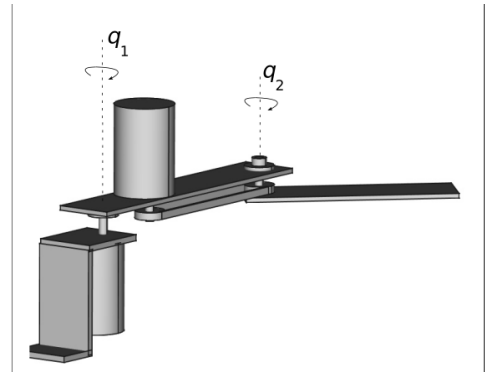


Figura 1. Esquema del robot 2D. (Fuente: Autores.)

Tabla 1: Parámetros del Robot.

m_1	0.280 kg
l_1	0.215 m
m_2	0.050 kg
l_2	0.16 m

3. ESTRATEGIA DE CONTROL Y SU IMPLEMENTACIÓN

En la Figura 2, se muestra el esquema de control en cascada (Ellis, 2012) que finalmente se empleo.

A continuación se presenta el hardware empleado y los programas que se desarrollaron para poder hacer posible el prototipado rápido y los cuales servirán luego para otro tipo de plantas y estrategias de control.

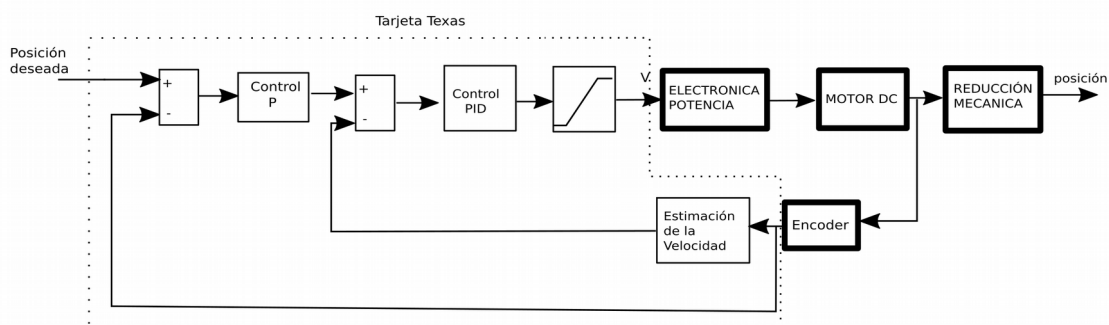


Figura 2. Esquema de control para una articulación (Fuente. Autores.)

3.1 Hardware Empleado

Se emplea una tarjeta de desarrollo del microcontrolador a 32 bits TMS320F28379D de Texas Instruments, el cual de manera abreviada, posee algunas capacidades como por ejemplo:

- Doble núcleo.
- Procesador en punto flotante.
- Memoria Flash de 512K words.
- Memoria RAM de 204Kb.
- Puertos para ADC, PWM, DAC, encoders.
- Puertos de comunicación: bus CAN, Serial, SPI, I2C.

Dado que la tarjeta de desarrollo trabaja la mayoría de sus pines a 3.3V se diseñó y construyó una tarjeta de acople desde niveles de tensión de 10V y 5V en el caso analógico y digital respectivamente. En la Figura 3 se muestran las tarjetas de desarrollo empleada y la tarjeta de puertos de acople de voltajes diseñada. La tarjeta de desarrollo encaja mediante unos conectores en la parte inferior a la de puertos.

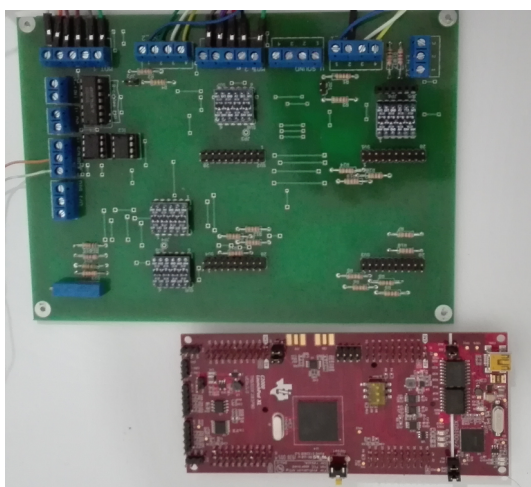


Figura 3. Tarjeta de Puertos de Entrada y Salida (arriba) y de desarrollo (abajo). (Fuente: Autores).

3.2 Programas

Para la programación del proyecto que permite compilar código ejecutable para la tarjeta de desarrollo se emplea el Code Composer Studio versión 9.1 y la librería X2C, a la cual se le hicieron algunas modificaciones y se adaptó para el procesador específico de Texas. Como ejemplo se muestra la función para la escritura de un dato al puerto PWM #1, en la Tabla 2.

Tabla 2: Función para sacar un dato por el PWM #1.

```
void saca_PWM_1(int16_t dato)
// dato entre 0 y 32767
{
    float percent=(abs(dato)/4096.0);
    percent=percent+0.07;
    Dato=dato;
    if(dato>10){ GPIO_writePin(123, 1);
                 GPIO_writePin(122, 0); }
    if(dato<=-10){GPIO_writePin(123, 0);
                  GPIO_writePin(122, 1); }

    EPWM_setCounterCompareValue(EPWM1_B
    ASE,EPWM_COUNTER_COMPARE_A,porcen
    t*EPWM_TIMER_TBPRD);
}
```

El tiempo de muestreo es posible variarlo desde el parámetro del bloque 'clock' de Scicos, el mínimo valor posible es 10 ms, este límite es debido a que se requiere transmitir por el puerto serial cuatro valores tipo palabra (16bits), los cuales permiten conocer respuestas temporales de señales escogidas por el diseñador del esquema de control.

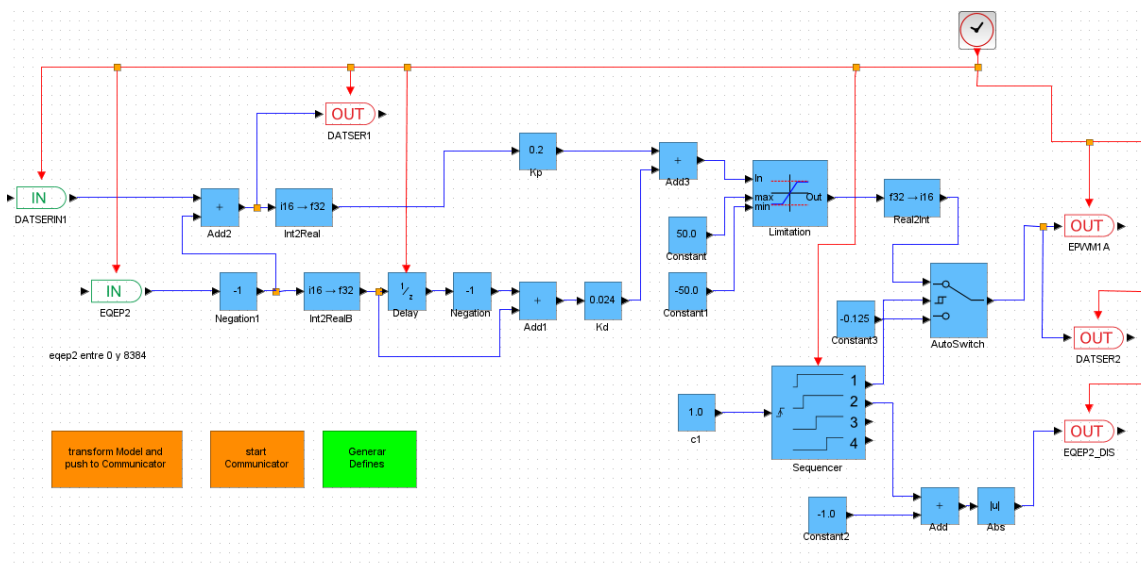


Figura 3. Diagrama de control en cascada para una articulación. (Fuente: Autores).

3.3 Implementación en SCICOS y X2C con Prototipado Rápido en Control

En la Figura 3 se muestra el diagrama de Scicos implementado, para la articulación q_1 .

Se empleó un bloque secuenciador para llevar las articulaciones a la posición de inicio, este funciona activando los motores durante un par de segundos y su movimiento causa su propia desactivación mediante finales de carrera (micro-switches). Pasado el tiempo de dos segundos, el secuenciador entrega el control al esquema de control en cascada, los cuales llevan el brazo a los valores articulares deseados.

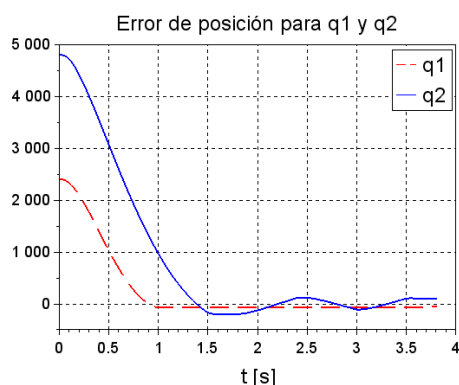


Figura 4: Respuestas articulares en el tiempo. (Fuente: Autores).

4. Resultados

En la Figura 4 se presentan las repuestas temporales del error para los valores deseados de encoder 2400 y 4800 (valores límite de los encoders). La obtención de esta respuesta se da gracias a los puntos de salida de datos serial "DATSERx" y el envío desde un PC por el puerto serial virtual (USB) de los valores deseados. La Figura 5 presenta una imagen del robot construido.



Figura 5: Robot construido. (Fuente: Autores).

5. Discusión

Algunos errores de estado estacionario persisten en las respuestas de la Figura 4, esto es debido a que se empleó un tiempo de muestreo de 10 ms y la lectura del encoder se podría estar haciendo para un valor posterior al deseado. También debido a que el motor requiere un voltaje mínimo para comenzar a funcionar, lo cual hizo necesario agregar un pequeño offset al valor de PWM.

Uno de los inconvenientes que tiene la librería X2C es que solo se pueden emplear los bloques suministrados por el desarrollador de esta, al contrario de Matlab/Simulink donde el

prototipado rápido se puede hacer con casi cualquiera de los bloques que trae el Simulink y sus librerías.

Conclusiones

El prototipado rápido en control permitió la realización del control de posición de un robot de dos grados de libertad, estas técnicas permiten que el estudiante dedique más tiempo a sintonizar el controlador y incluso pueda probar otras estrategias de control.

El prototipado rápido en control para software libre (Scilab/Scicos) todavía está en fase de desarrollo y faltan librerías y características de los programas comerciales como Labview o Simulink, pero con este trabajo se comprueba que ya es posible emplearlo para cursos de control clásico e incluso para estrategias de espacio de estados, además la librería X2C permite que se agreguen nuevos bloques según las necesidades que se tengan, por ejemplo para cursos de control inteligente.

REFERENCIAS

- Hanselmann, H., U. Kiffmeier, L. Koster, M. Meyer, and A. Rukgauer. (1999) 'Production quality code generation from simulink block diagrams', In Proceedings of the 1999 IEEE International Symposium on Computer Aided Control System Design, pages 213–218.
- Grabmair, S. Mayr, M. Hochwallner, and M. Aigner. (2014) 'Model based control design a free tool-chain'. In 2014 European Control Conference (ECC).
- Chen hwan-Hsen, H.-L. T. J.-C. T. (2004), Robot control system implementation with rapid control prototyping technique, in '2004 IEEE International Symposium on Computer Aided Control Systems Design'.
- Flórez Frank, J. A. T. G. (2017), 'Prototipado virtual de un controlador por etapas para un brazo robótico en una aplicación as/rs', Revista ONTARE, Univ. EAN, Colombia.
- Junwon Jang, Soohee Han, H. K. & Ahn, C. K. (2008), Rapid control prototyping for robot soccer, in 'Proceedings of the 17th World Congress The International Federation of Automatic Control Seoul, Korea'.
- Kao-Shing Hwang, Wen-Hsu Hsiao, G.-T. S. & Chen, K.-J. (2011), 'Rapid prototyping platform for robotics applications', IEEE TRANSACTIONS ON EDUCATION, VOL. 54, NO. 2.
- Lapusan, C., Matis, V., Balan, R., Hancu, O., Stan, S. & Lates, R. (2008), Rapid control prototyping using matlab and dspace. application for a planar parallel robot, in '2008 IEEE International Conference on Automation, Quality and Testing, Robotics', Vol. 2, pp. 361–364.
- Lee, Y. S., Jo, B. & Han, S. (2017), 'A light-weight rapid control prototyping system based on open source hardware', IEEE Access 5, 11118–11130.
- Kielan, Pawel, G. K. (2014), The idea of laboratory stand for rapid prototyping of robot control systems for remote operation via internet, in '15th International Workshop on Research and Education in Mechatronics(REM)'.
- Yongqiang Ye, A. T. & Liu, X. (2015), Rapid-prototyping of iterative learning control using matlab/simlink hybrid-programming, in 'Canadian Conference on Electrical and Computer Engineering'.
- Reyes Fernando, (2011), 'Robótica, Control de robots manipuladores', Ed. Alfaomega.
- Spong, Mark, S. Hutchinson, M. Vidyasagar, (2006) 'Robot Modeling and Control', Ed. Wiley.
- Ellis, George, (2012) 'Control System Design Guide Using Your Computer to Understand and Diagnose Feedback Controllers', Fourth Edition, Ed. Elsevier.