



Sistema de monitoreo y servidor WEB con Raspberry Pi para el control de un robot neumático

Cabrera Rufino Marco Antonio, ✉ Juan Manuel Ramos Arreguín, Rivas-Araiza E.A.,
Efren Gorrostieta Hurtado, Jesus Carlos Pedraza Ortega

Universidad Autónoma de Querétaro, Facultad de Ingeniería
✉ jsistdig@yahoo.com.mx

Resumen

El monitoreo de datos es indispensable para poder comprender y analizar parámetros con los que trabaja un controlador, ya sea clásico, difuso, neuronal, etc. Así, señales como el sobrepaso y el tiempo de respuesta pueden ser graficados ya sea en tiempo real o después de cierto tiempo para mostrarse desde una página WEB. En este proyecto se propone un sistema de monitoreo desde un servidor que permita facilitar al usuario las características de un controlador, para esto la etapa de control y de monitoreo están separadas, para la etapa de control se puede utilizar como en este proyecto microcontroladores de alta gama como los LAUNCHXL de Texas Instruments o podría usarse una FPGA para sistemas más complejos. Una tarjeta Raspberry Pi se utiliza como módulo de monitoreo, sobre la Raspberry se utiliza una base de datos (mariaDB) para guardar los parámetros más importantes que se vayan registrando, estos parámetros se pueden presentar como los avances del proyecto, se propone el framework Django para la elaboración del servidor WEB debido a que ofrece buenas herramientas para la elaboración de páginas WEB además de conectarse fácilmente con base de datos, este proyecto integra una gran cantidad de herramientas de software (todas ellas programadas principalmente en Python) que permiten su uso en aquellos proyectos de investigación que lo requieran.

Palabras clave: WEB server, internet de las cosas, control, Raspberry, Python, servidor web, monitoreo de señales.

Abstract

Data monitoring is essential to understand and analyze parameters in which controllers work. Can be classic (PID), fuzzy, neural, etc. Therefore, signals such as maximum overshoot and time response can be plotted either in real time or after certain time to be displayed on a WEB site. In this project, a monitoring system from a WEB server is proposed to make easier the achievement of the controller characteristics, for this the control stage and monitoring stage are separated. For the control stage it is proposed a high-end microcontroller such as LAUNCHXL from Texas Instruments or could be a FPGA for more complex systems, besides of monitoring these systems include a database (mariaDB) to save the most important data that are being plotted, this data can be considered as project advances. It is proposed the Django framework to make the WEB because it offers good tools to the server such as exceptionally good connectivity with the database. This project integrates a large number of tools (Django, plotting, communications, etc.) all of them programmed in python, this proposal is useful for those research projects that may require the same structure.

Keywords: WEB server, internet of things, control, Raspberry, Python, PID, signal monitoring.

1. Introducción

Los sistemas neumáticos son muy usados en la industria, especialmente en procesos de producción. Sus principales usos son acciones de tomar y colocar piezas (pick and place), manufactura (perforación, fresado, entre otras), líneas de producción (accionamiento de ejes para selección de productos, sujetar productos, cerrar recipientes, entre otros. También se aplican en desplazamiento de sistemas, posicionamiento, corte, estampado, generación de vacío, entre otros [1].

El uso de la neumática en la industria ha cobrado una gran relevancia, de tal manera que está impulsando el estudio de maneras de mejorar el comportamiento del aire en las cámaras [2]. Debido a la alta no linealidad en el comportamiento del aire, se vuelve complejo el poder controlar el comportamiento de actuadores neumáticos. Por lo tanto, se está trabajando también en el desarrollo de modelos matemáticos [3].

Uno de los aspectos más importantes para controlar el comportamiento de un actuador neumático, es el control del flujo de aire hacia las cámaras de los actuadores. Sin embargo, el control no es sencillo, ni fácil de implementar. En base a esto, se han desarrollado diversos trabajos enfocados a controlar la presión de aire en diversos actuadores, como los músculos artificiales neumáticos [4], control de flujo de aire en respiradores [5], entre otros.

En la Universidad Autónoma de Querétaro, se está trabajando en un proyecto de control de posición de un robot con actuadores neumáticos, en lugar de actuadores eléctricos. El principal problema que se tiene es el control de presión de aire, con el uso de reguladores de flujo. En este trabajo se presenta el desarrollo de una alternativa para manipular el grado de apertura de una válvula de flujo, usando tecnología IoT, sistemas embebidos y teoría de control.

Este trabajo se enfoca en el desarrollo de herramientas de software y hardware para el control de las válvulas reguladoras de flujo, realizando un intercambio de información con sistemas remotos basados en Internet de las Cosas (IoT). También se hace uso de herramientas de hardware ya creadas.

2. Metodología

En la Figura 1, se presenta un diagrama a bloques con la metodología usada en este trabajo.

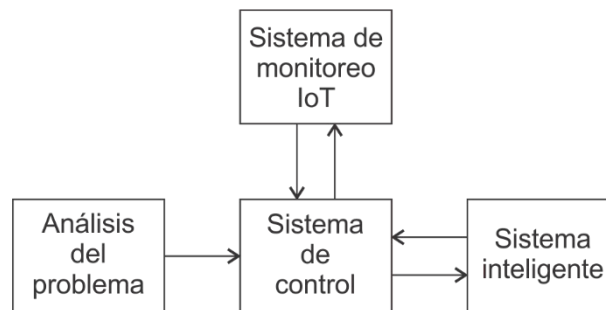


Figura 1. Diagrama a bloques de la metodología a usar.

A continuación, se detalla un poco más cada etapa de la metodología.

2.1 Análisis del problema

El objetivo es controlar la posición de un robot neumático. Se están usando actuadores neumáticos. La principal característica de un actuador neumático es su alta no linealidad en su comportamiento. Existen diferentes tipos de actuadores neumáticos, entre los cuales se tiene a los actuadores con amortiguamiento y sin amortiguamiento. La figura 2 muestra la imagen de un actuador con amortiguamiento en los extremos [6] [7].

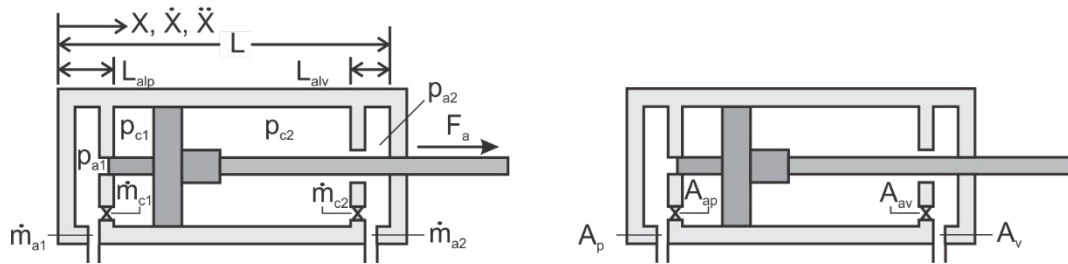


Figura 2. Modelo de un actuador neumático de 2 grados de libertad [6] [7].

Donde:

X \Rightarrow Desplazamiento del vástago, respecto a la base del cilindro.

\dot{X} \Rightarrow Velocidad de desplazamiento del vástago, respecto al cilindro.

\ddot{X} \Rightarrow Aceleración del vástago, respecto al cilindro.

L \Rightarrow Longitud total de la carrera del vástago a lo largo del cilindro.

L_{alv} \Rightarrow Longitud horizontal de la cámara de amortiguamiento de la base del cilindro.

L_{alp} \Rightarrow Longitud horizontal de la cámara de amortiguamiento del lado del vástago.

P_{a1} \Rightarrow Presión de aire en la cámara de amortiguamiento de la base del cilindro.

P_{a2} \Rightarrow Presión de aire en la cámara de amortiguamiento del lado del vástago.

P_{c1} \Rightarrow Presión de aire en la cámara de la base del cilindro.

P_{c2} \Rightarrow Presión de aire en la cámara del lado del vástago.

\dot{m}_{a1} \Rightarrow Flujo máscico de aire entre el exterior y la cámara de amortiguamiento de la base del cilindro.

\dot{m}_{a2} \Rightarrow Flujo máscico de aire entre el exterior y la cámara de amortiguamiento del lado del vástago.

\dot{m}_{c1} \Rightarrow Flujo máscico de aire entre las cámaras del lado de la base del cilindro.

\dot{m}_{c2} \Rightarrow Flujo máscico de aire entre las cámaras del lado del vástago.

A_p \Rightarrow Área disponible para flujo de aire en el lado del pistón.

A_v \Rightarrow Área disponible para flujo de aire en el lado del vástago.

A_{ap} \Rightarrow Área disponible para flujo de aire en la almohadilla del lado del pistón.

A_{av} \Rightarrow Área disponible para flujo de aire en la almohadilla del lado del vástago.

Todas las variables anteriores, están relacionadas por el modelo matemático que se muestra en (1) [6] [7].

Para $0 \leq X \leq L$:

$$\dot{X} = \frac{d}{dt} X \quad (1a)$$

$$D\dot{X} = \frac{d^2}{dt^2} X \quad (1b)$$

Para $0 \leq X \leq L_{alp}$:

$$\dot{P}_{a1} = \frac{kRT_0}{A_{ap} \left(X + \frac{\Delta A_p}{A_{ap}} \right)} \left(\dot{m}_{1a} - \dot{m}_{1c} - \frac{A_{ap}}{RT_0} P_{a1} DX \right) \quad (1c)$$

$$\dot{P}_{c1} = \frac{kRT_0}{(A_p - A_{ap})X} \left(\dot{m}_{1c} - \frac{A_p - A_{ap}}{RT_0} P_{c1} DX \right) \quad (1d)$$

Para $L_{ap} < X \leq L$:

$$\dot{P}_{a1} = \frac{kRT_0}{A_p(X + \Delta)} \left(\dot{m}_{1a} - \frac{A_p}{RT_0} P_{a1} DX \right) \quad (1e)$$

$$\dot{P}_{c1} = \frac{kRT_0}{A_p(X + \Delta)} \left(\dot{m}_{1c} - \frac{A_p}{RT_0} P_{c1} DX \right) \quad (1f)$$

Para $0 \leq X < (L - L_{av})$:

$$\dot{P}_{c2} = \frac{kRT_0}{(A_p - A_v)(L - X + \Delta)} \left(\dot{m}_{2c} + \frac{A_p - A_v}{RT_0} P_{c2} DX \right) \quad (1g)$$

$$\dot{P}_{a2} = \frac{kRT_0}{(A_p - A_v)(L - X + \Delta)} \left(\dot{m}_{2a} + \frac{A_p - A_v}{RT_0} P_{a2} DX \right) \quad (1h)$$

Para $(L - L_{av}) \leq X \leq L$:

$$\dot{P}_{c2} = \frac{kRT_0}{(A_p - A_v)(L - X)} \left(\dot{m}_{2c} + \frac{A_p - A_v}{RT_0} P_{c2} DX \right) \quad (1i)$$

$$\dot{P}_{a2} = \frac{kRT_0}{L - X + \frac{\Delta A_p}{A_{av} - A_v}} \left(\frac{\dot{m}_{2a} - \dot{m}_{2c}}{A_{av} - A_v} + \frac{P_{a2} DX}{RT_0} \right) \quad (1j)$$

Fuerza del Pistón:

$$F_a = p_1 A_p - p_2 (A_p - A_v) - F_{sv} \quad (1k)$$

Como se puede observar en las ecuaciones (1a-k), la apertura para el flujo de aire es lo que va a permitir controlar el movimiento del actuador neumático.

2.2 Sistema de control

Para controlar el movimiento del actuador, se ha considerado el uso de una electroválvula de 3 posiciones y 5 terminales (válvula 5/3), así como dos reguladores de flujo de aire, como se muestra en la figura 3. Las señales D_1 y D_2 se usan para controlar el sentido del movimiento del vástago del actuador, acorde con los valores de la tabla 1. La señal A_1 se usa para controlar el flujo del aire a presión que entra al pistón, y la señal A_2 se usa para regular el flujo de aire que sale del actuador hacia la atmósfera.

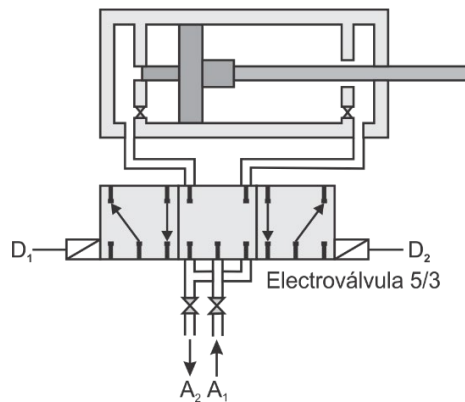


Figura 3. Esquema para controlar el movimiento del actuador neumático.

Tabla 1. Valores de D_1 y D_2 , para el comportamiento del actuador neumático.

D_1	D_2	Movimiento del vástago
0	0	No se mueve. La válvula está en punto neutro.
0	1	Hacia dentro del actuador.
1	0	Hacia fuera del actuador.
1	1	No permitido.

El controlador propuesto se muestra en la figura 4. Se busca diseñar el controlador más eficiente, por lo cual se planea utilizar uno o más algoritmos de control inteligente del tipo neuronal, difuso, o una combinación de ambos. Para un diseño se plantea que una red neuronal puede generar los conjuntos de membresía del sistema, mientras que la parte difusa va a controlar el sistema.

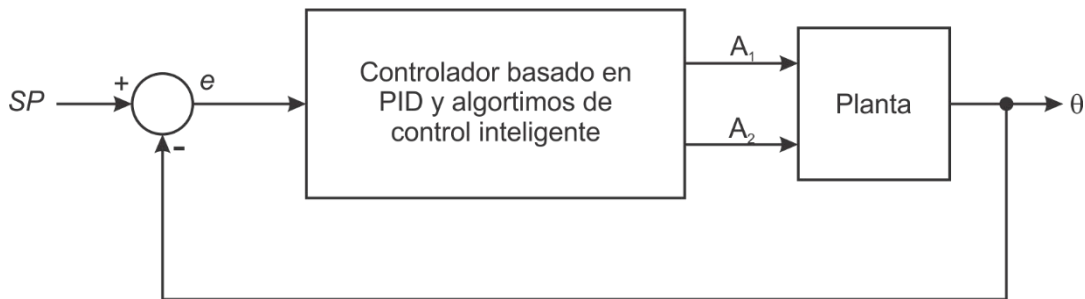


Figura 4. Esquema del controlador basado en PID y algoritmos inteligentes.

Cada grado de libertad está constituido por 1 electroválvula 5/3, 2 válvulas reguladoras de flujo de aire, y 1 encoder que indica la posición del grado correspondiente. El diagrama de conexión para 1 DOF se muestra en la figura 5.

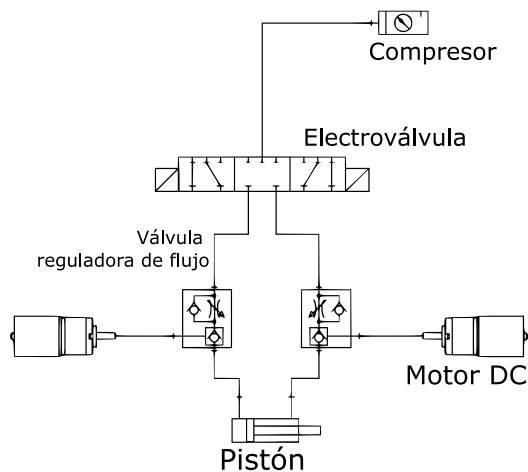


Figura 5. Esquema del sistema neumático para un grado de libertad.



En la figura 5 se muestra el controlador inteligente de forma más profunda para un robot neumático de 3 grados de libertad, la posición y orientación que se desea alcanzar son dadas por parte del usuario, el controlador inteligente determinara la posición de todos los motores DC, el cambio de posición de los motores cambiara la apertura de aire en las válvulas, y el cambio de presión generara movimiento en los pistones, hay que mencionar que el controlador inteligente tomara en cuenta las no linealidades del sistema buscando la manera más eficiente de posicionar el efector final en la posición deseada. Los motores DC son posicionados por controladores Proporcional Integral (PI), estos son los más eficientes para motores DC donde los parámetros son conocidos, se retroalimentará los valores de las coordenadas articulares al controlador inteligente para que este efectúe los cambios necesarios.

Figura 6. Esquema amplificado del controlador basado en PID y algoritmos inteligentes.

2.3 Sistema de monitoreo IoT

La parte esencial de este trabajo es el sistema de monitoreo basado en tecnología IoT, el cual tiene como principales características:

- Visualización en tiempo real.
- Servidor Web.
- Base de datos.
- Seguridad del sistema de control.

De esta manera, se busca que la información generada por el controlador y el robot sea accesada a través de Internet. De esta manera, la información puede ser recuperada en cualquier momento a distancia, y se puede realizar análisis del comportamiento del robot en cualquier instante y desde cualquier lugar.

3. Desarrollo del trabajo

El desarrollo del trabajo se divide especialmente en 3 partes: controlador, sistema de monitoreo IoT, e integración de sistemas. A continuación, se describe cada una de las partes mencionadas.

3.1 Sistema de control

Para el sistema de control se usa la tarjeta LAUNCHXL-F28377s [8], el cual contiene el hardware embebido necesario para este proyecto. Parte de este software son dos módulos eQEP para la lectura de dos encoders ópticos, y módulos PWM que sirven para el control de unos motores acoplados a la válvula reguladora de flujo. También tiene capacidad de manejar diversas interrupciones útiles en el proyecto.

La etapa de control es la más importante en este sistema y debe ser independiente a la etapa de monitoreo realizada por la Raspberry Pi, la ventaja de utilizar *Simulink* [9] es la rápida implementación del PID ya que requiere poco tiempo de programación y solo requiere llenar el formulario del bloque del PID, además se puede implementar controladores difusos y neuronales con mucha facilidad solo concentrándose en el diseño y no en la programación. En la figura 7 se puede observar el programa de control realizado en *Simulink*. Como se puede observar, la elaboración del controlador a bloques facilita considerablemente el diseño.

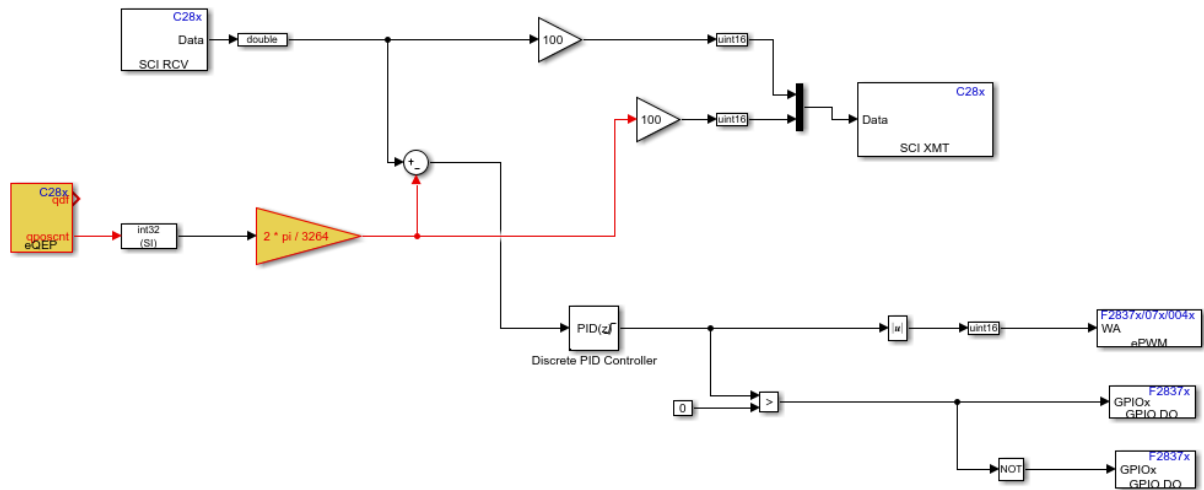


Figura 7. Control PID realizado en *Simulink*.

Una de las desventajas de utilizar *Simulink* es la falta de optimización que genera al utilizar los bloques y que hay una “caja negra” en gran parte del programa que desconocemos, en este proyecto se implementa también la realización del código en C con *Code Composer v9* [10], en la figura 8 se observa parte del código que sobrepasa ya las 800 líneas de código.

Se debe mencionar que la LAUNCHXL es un potente DSP pero su curva de aprendizaje es lenta, lo que vuelve lento el desarrollo del programa, con respecto a hacerlo con *Simulink* que es más rápido, las dos opciones presentan ventajas y desventajas en las cuales se observan en la tabla 1.



3.2 Sistema de monitoreo

Para el sistema de monitoreo se utiliza una tarjeta de desarrollo con procesador ARM Linux embebido, el cual brinda una ayuda excepcional en la instrumentación de experimentos, así como poder guardar avances y resultados de los proyectos donde sea indispensable el uso de sistemas de control. En este proyecto se propone la Raspberry Pi 4 [11] el cual posee una frecuencia de trabajo de 1.6Mhz y 4 Gb de RAM.

```

721 //Ecuación en diferencias.
722 //Ecuación integral.
723 if(fabs(e0) > epsilon){
724     yi0 = yi1 + (Ts * e0);
725 }
726
727
728 //Ecuación derivativa.
729 yd0 = (e0 - e1) / Ts;
730
731 //Resultado PID.
732 ypid = (Kp * e0) + (ki * yi0) + (Kd * yd0);
733
734 //Ajuste y corrección de la SALIDA Y(n)
735 //delimitada por los límites 0 y 5000.
736 //YN += ypid;
737 if(ypid > MAX){
738     ypid = MAX;
739 }
740 if(ypid < MIN){
741     ypid = MIN;
742 }
743 PWMAB = fabs(ypid);
744 if(sp > pos_mech + 0.1){
745     //disable_valve_2();
746     set_pwm_motor_1((unsigned int)PWMAB, zero);
747 }
748 else if(sp < pos_mech - 0.1){
749     set_pwm_motor_1(zero, (unsigned int)PWMAB);
750     enable_valve_1();
751 }
    
```

Figura 8. Control PID realizado en CCS V9.

Tabla 1. Ventajas y desventajas de plataformas de programación de la tarjeta LAUNCHXL

Programa	Ventajas	Desventajas
Simulink	Curva de aprendizaje rápida	Falta de eficiencia en Código
	Facilidad de implementar controladores inteligentes	Falta de conocimiento en el funcionamiento del Código
	Programación en bloques parecida a VHDL pero de forma visual y no abstracta	Depender de software de paga
CCS v9	Control total en la eficiencia y conocimiento del Código	Curva de aprendizaje lenta
	No dependencia de software de paga	Gran cantidad de líneas de Código para depurar

Para la creación del servidor web se utiliza el framework Django [12]. La página principal muestra un resumen del proyecto y sus alcances en las áreas de control inteligente, sistemas embebidos e internet de las cosas. Las otras páginas sirven para poder mostrar las gráficas y desplegar los valores que deseamos de la base de datos.

La elaboración de la base de datos se hace con MySQL [13], es importante mencionar que el uso de tecnologías libres es esencial para este proyecto, lo cual nos permite libremente usar este proyecto con fines de investigación, en la bases de datos se guardan cada cinco minutos las señales de control, si además se requiere de que se guarde estas señales en un momento específico, está la opción para hacer el guardado en cualquier momento que se desee.

3.3 Integración de los sistemas de monitoreo y control.

El controlador fue implementado en una tarjeta LAUNCHXL-F28377s de Texas Instruments, la cual se muestra en la figura 9. Esta tarjeta tiene un costo aproximado de \$100.00 USD.

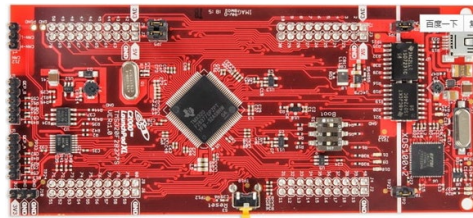


Figura 9. Tarjeta LAUNCHXL-F28377s.

Para el Sistema de Monitoreo, se usa una tarjeta Raspberry Pi 4, la cual se comunica con el sistema de control implementado en la tarjeta de la figura 9. La comunicación entre las tarjetas es mediante comunicación SPI. Esto se puede ver en el diagrama a bloques de la figura 9, donde se ilustra la integración del control y el monitoreo, con una comunicación SPI entre ambos sistemas.

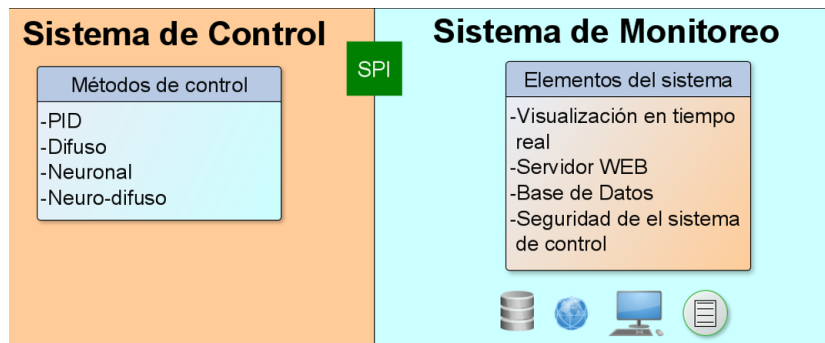


Figura 10. Sistema de monitoreo propuesto.

Tanto la tarjeta Raspberry Pi como la LAUNCHXL-F28377s tienen embebido puertos de comunicación SPI, la cual es necesaria para realizar la comunicación. La información que se transmite entre ambos sistemas son las variables de control: valor de referencia, error, posición del robot neumático, entre otros. El número de señales a enviar depende de las que el usuario programe.

Para las gráficas de las señales se utiliza la librería matplotlib [14] de Python la cual permite crear graficas similares a Matlab, estas graficas son guardadas en una dirección especial en la tarjeta Raspberry, donde después el servidor WEB las toma para ser mostradas por un navegador de Internet. Las señales principales que se muestran son las de la señal del controlador y el error. Los datos

obtenidos para las gráficas son de un controlador PID [15], pero además se puede implementar controladores difusos, neuronales o una combinación de estos.

4. Resultados

En la figura 11 podemos observar la página principal de nuestro servidor WEB, el cual contiene una plantilla que nos permite navegar por todo el servidor web. En este proyecto se habilita y se trabaja sobre las pestañas de pantalla principal, sensores y base de datos. Se pretende tener un trabajo sólido y ampliable de forma que se pueda expandir para manejar y controlar más sistemas. Por esta razón se puede considerar esta parte como la versión 1.0 de nuestro servidor WEB.

[HOME](#) [SENSORES](#) [BASE DE DATOS](#) [CREADORES DEL PROYECTO](#) [ACERCA DE](#)

POSICIONAMIENTO DE UN ROBOT DE 3 GRADOS DE LIBERTAD CON ACTUADORES NEUMÁTICOS

Universidad Autónoma de Querétaro
Estamos a día: Dec. 3, 2019, 6:21 p.m.

Control Inteligente

Control de posición de una valvula de la base del robot del proyecto de tesis, esto es, controlar un grado de libertad del robot a través de logica difusa o redes neuronales. Se utiliza una tarjeta LAUNCHXL de Texas Instruments Programada en simulink esta tarjeta facilitara la implementación del controlador.

Grado de libertad a controlar

Internet of things

Figura 11. Página principal del servidor web. [referencia]

En la página principal solo se habla de las características del proyecto, sus alcances y como se compone. También es posible desplegar los valores obtenidos por los sensores. En la segunda pestaña, se muestra la página de una manera similar a la figura 12, en la cual se muestra el nombre de la Universidad Autónoma de Querétaro, seguido de tres botones que nos permiten crear graficas de posición y del error del motor. Para que esto funcione, se debe establecer la comunicación con la tarjeta LAUNCHXL, presionando el botón “comenzar comunicación con la LAUNCHXL”, el cual abre el puerto de comunicación serial y guarda los datos en un archivo con extensión txt, que posteriormente se usa para mostrar las gráficas respectivas.

Debajo de los botones se muestran los valores de las señales de interés, las cuales son iniciadas con un valor de 0.0. Para ver los valores reales, es necesario establecer la comunicación serial entre el sistema de control y el de monitoreo. Una vez iniciada la transmisión, se modifica un poco la visualización, como se muestra en la figura 13. los primeros dos botones son para actualizar los datos que se van recopilando los sensores al momento de presionar el motor al presionarlo se actualizan los valores, el segundo botón es para cerrar la comunicación serial.



[HOME](#) [SENSORES](#) [BASE DE DATOS](#) [CREADORES DEL PROYECTO](#) [ACERCA DE](#)

POSICIONAMIENTO DE UN ROBOT DE 3 GRADOS DE LIBERTAD CON ACTUADORES NEUMÁTICOS

Universidad Autónoma de Querétaro

Las señales obtenidas de los sensores son:

Posición del motor 0.0

Setpoint del motor 0.0

Error de la señal 0.0

Servidor web para acceder a los parametros del proyecto via remota, se utiliza raspberry pi, python con django para la realización de este proyecto, la etapa de control se elabora en un DSP de la marca Texas Instruments.

Figura 12. Página para el control y visualización de datos de los sensores.

Universidad Autónoma de Querétaro

Las señales obtenidas de los sensores son:

Posición del motor 10.06

Setpoint del motor 30.0

Error de la señal 19.94

Setpoint

Figura 13. Visualización del panel de adquisición de datos.

Debajo de las señales se encuentra un botón para el cambio del valor de referencia (setpoint) y un botón para enviar el dato por puerto serial. El setpoint puede ser actualizado tantas veces como se requiera. Al cerrar la comunicación serial se muestran los datos adquiridos, de manera que se puede graficar tanto posición como error de la señal, como se muestra en la figura 14. Una base de datos se implementó al final que guarda la información cada cierto tiempo programable además el usuario tiene la opción de guardar los datos haciendo clic en el botón de guardar, la base de datos fue implementada utilizando MariaDB que es una versión libre parecida a MySQL[16].

En la fotografía de la figura 15 se muestra un prototipo del sistema, se pretende implementarlo finalmente en el proyecto de tesis donde se apoya de controladores inteligentes para el control de un robot con actuadores neumáticos, por lo tanto, se pretende que este proyecto vaya en crecimiento, adaptándose a las necesidades y requerimientos necesarios de este proyecto.

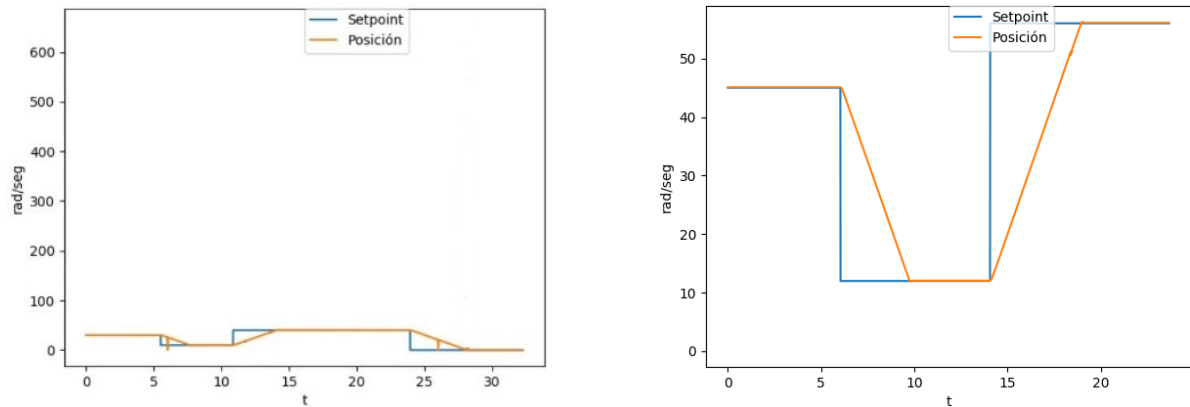


Figura 14. Grafica de la posición del motor.

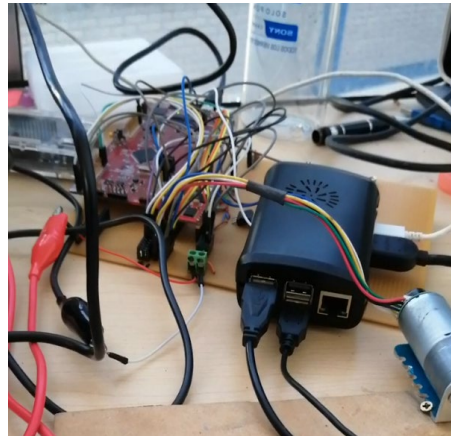


Figura 15. Prototipo del sistema de monitoreo.

5. Conclusiones

El servidor web funcionó satisfactoriamente, realizando graficas del controlador PID efectuado en este proyecto, independientemente de los resultados del controlador, puede servir como ajuste y sintonización de estos desde otra computadora o portátil además se pueden implementar y añadir mas señales de control, la realización de base de datos fue también satisfactoria añadiendo los valores de posición o velocidad en los tiempos deseados lo cual puede servir para el correcto funcionamiento del controlador en largos tiempos de prueba.

Se propone el uso del Microcontrolador LAUNCHXL ya que este permite implementar casi cualquier tipo de controlador de manera fácil y rápida desde Simulink de Matlab, pero puede ser reemplazado por cualquier otro componente semejante desde un Arduino hasta una FPGA.

Se pretende seguir avanzando con el proyecto, adaptando el servidor web con las necesidades del usuario, añadiendo más señales de control y sus respectivas señales características tales como error, setpoint y energía suministrada al sistema. Además, se pretende sustituir la LAUNCHXL por una FPGA, para lo cual se puede implementar todo en una sola tarjeta de desarrollo como la tarjeta ZYBO



de XILINX la cual tiene embebida el procesador con Linux embebido y la FPGA donde el hardware se reduciría considerablemente.

Trabajo a futuro

Por último, hay que mencionar que este proyecto está diseñado para poder crecer e implementar nuevas tecnologías y herramientas de hardware y software, como la elaboración de reportes de las variables de control para ser incluidas en *Latex*. También realizar un monitoreo en tiempo real con una aplicación *WEB*, etc.

Referencias

- [1] Quishpe Rosero, Daniel Andrés. Estudio de la neumática y sus aplicaciones en diferentes campos de la industria. Trabajo de titulación previo a la obtención del Título de Tecnólogo Superior en Electromecánica. Carrera de Electromecánica. Quito: UCE. 75 p. 2018.
- [2] Guo D., Kang Z. Chamber layout design optimization of soft pneumatic robots. *Smart Materials and Structures*. Ed. IOP Science, Vol. 29, número 2, Online ISSN: 1361-665X, Print ISSN: 0964-1726. 2020.
- [3] Kaplish, A. Modeling and System Identification of Pneumatic Robot (Doctoral dissertation).
- [4] T. Tagami, T. Miyazaki, T. Kawase, T. Kanno and K. Kawashima, "Pressure Control of a Pneumatic Artificial Muscle Including Pneumatic Circuit Model," in *IEEE Access*, vol. 8, pp. 60526-60538, 2020, doi: 10.1109/ACCESS.2020.2983602.
- [5] Gama, F. O., Martins, J. K. D. C., de Miranda, T. F., Tomé, W. M. D. F., Silva, S. N., & Fernandes, M. A. Control of airflow in ventilation systems using embedded systems on microcontrollers. *Microsystem Technologies*, 25(11), 4067-4076. 2019.
- [6] Fernando F. Kiyama, José E. Vargas; Modelo Termo-Mecánico Para Un Manipulador Tipo Dieléctrico; *Información Tecnológica*; volumen 15; número 5; páginas 23 a 31; 2004; ISSN 0716-8756.
- [7] Kiyama F., Vargas E.; Modelo Integral del Accionamiento Neumático para un Robot Flexible Industrial; *CRODE-CIDESI*; 2º Congreso Nacional de Mecatrónica, 2003.
- [8] Instruments, T. (2015). *LAUNCHXL-F28377S Overview User's Guide*. Dallas.
- [9] Klee, H., & Allen, R. (2016). *Simulation of dynamic systems with MATLAB and Simulink*. Crc Press.
- [10] Instruments, T. (2018). *Code composer studio (CCS) integrated development environment (IDE)*.
- [11] Molloy, D. (2016). *Exploring Raspberry Pi*. Exploring Raspberry Pi. John Wiley & Sons.
- [12] Natri, S. (2018). *Django-The Easy Way: A step-by-step guide on building Django websites*. Independently published.
- [13] Bell, C. (2016). *MySQL for the Internet of Things*. Springer.
- [14] Saha, A. (2015). *Doing Math With Python*. Network Security.
- [15] Ogata, K. (2010). *Engenharia de Controle Moderno*. Control Engineering.
- [16] West, A. W., & Prettyman, S. (2018). *Practical PHP 7, MySQL 8, and MariaDB Website Databases: A Simplified Approach to Developing Database-Driven Websites*. Apress.

Biografía de Autores

Marco Antonio Cabrera Rufino. Es Ingeniero en Electrónica egresado de la Universidad Autónoma Metropolitana en la unidad Azcapotzalco, Ciudad de México, como parte de requisito de egreso realizó su proyecto terminal con título "Medición de franjas de interferencia para la medición de desplazamientos nanométricos" en el Instituto de Ciencias Aplicadas y Tecnología (antes CCADET) de la UNAM, campus CU, donde se definió su interés a seguir desarrollándose como futuro investigador,



después, durante algunos años trabajó entre Querétaro y Guanajuato en varias integradoras que permitió desarrollarse y crecer más como ingeniero, realizó proyectos para empresas como Neo Aluminio, Cordaflex, Pierburg Rheinmetall Automotive (anteriormente KSPG), entre otras. Actualmente realiza la maestría en ciencias en instrumentación y control en la Universidad Autónoma de Querétaro, Campus CU, con el tema de tesis “posicionamiento de un robot de 3 grados de libertad con actuadores neumáticos”, a cargo del Dr. Juan Manuel Ramos Arreguín.

Juan Manuel Ramos Arreguín. Tiene Doctorado en Ciencias y Tecnología con Especialidad en Mecatrónica en el Centro de Ingeniería y Desarrollo Industrial. Con Maestría en Ingeniería Eléctrica con opción en Instrumentación y Sistemas Digitales en la Universidad de Guanajuato. La ingeniería en Comunicaciones y Electrónica en la Universidad de Guanajuato. Fue Presidente de la Asociación Mexicana de Mecatrónica en el periodo 2013 a 2016. Pertenece al SNI en nivel I. Cuenta con reconocimiento al perfil PRODEP. A partir de 2017 es Senior Member en el IEEE. Profesor de tiempo completo en la Universidad Autónoma de Querétaro, en la Facultad de Ingeniería.

Edgar Alejandro Rivas Araiza. Profesor-Investigador de la Universidad Autónoma de Querétaro desde 2010 adscrito a la facultad de ingeniería, es miembro del SNI nivel I, cuenta con perfil deseable PRODEP y forma parte del cuerpo académico consolidado "Sistemas Embebidos y Aplicaciones", sus actividades incluyen docencia a nivel licenciatura y posgrado, ha dirigido y colaborado en distintos proyectos de investigación con financiamiento externo como FOMIX, PEI y FESE. Su área de interés incluyen visión por computadora, sistemas embebidos, automatización, procesamiento de señales y control automático. Cuenta con publicaciones en congresos nacionales e internacionales, en revistas arbitradas e indizadas, colaborador en capítulos de libro, además ha dirigido trabajos de tesis a nivel licenciatura, maestría y doctorado. Miembro de la IEEE desde 2004.

Efren Gorrostieta Hurtado. Doctor en ingeniería con especialidad en mecatrónica, Maestro en ciencias y tecnología con especialidad en automatización y control. Licenciado en ingeniería electrónica con la especialidad de Control y Biomédica en el instituto tecnológico de estudios superiores de occidente. Se ha desempeñado como profesor investigador en distintas instituciones de educación superior de México, como en el Instituto tecnológico Nacional en las unidades de Morelia y Querétaro, Universidad la Salle, Instituto tecnológico de estudios superiores de occidente. Actualmente, es Profesor Investigador de la Universidad Autónoma de Querétaro donde también se desempeña como profesor en la Maestría de Inteligencia Artificial. Ha sido editor de tres libros de Robótica y revisor de varias revistas del área. Cuenta con varias publicaciones a nivel nacional e internacional en revistas arbitradas y congresos especializados. También se ha desempeñado como jefe del área de posgrado, coordinador de carrera, presidente de la asociación mexicana de mecatrónica y sección Querétaro de IEEE. Las áreas de trabajo son el desarrollo de sistemas mecatrónicos, sistemas de aprendizaje máquina, robots caminantes y robots móviles.

Jesús Carlos Pedraza Ortega. Jesús Carlos Pedraza Ortega realizó sus estudios de Maestría en la FIMEE, Universidad de Guanajuato. Obtuvo el Doctorado en Ingeniería Mecánica con especialidad en Robótica - Sistemas de Reconstrucción 3D en la University of Tsukuba en Japón, donde trabajó con el desarrollo de un sistema monocular de reconstrucción 3D utilizando. Como docente, ha impartido diferentes cursos en los tres niveles de estudios (Licenciatura, Maestría y Doctorado) desde 1997, actualmente en la Universidad Autónoma de Querétaro. Es Senior Member por la IEEE y es miembro de la Academia Mexicana de Ciencias. Sus líneas de investigación son sistemas de reconstrucción 3D, inteligencia artificial aplicada a sistemas de visión, entre otros.