

Se presenta el diseño un controlador neurodifuso que posibilita el uso de conocimiento de usuario experto con la adaptabilidad y capacidad de aprendizaje de las redes neuronales.



Control neurodifuso de un péndulo invertido rotacional

Neuro Fuzzy Control for an Inverted Rotational Pendulum

RESUMEN

En este artículo, se presenta el diseño de un controlador neurodifuso para el péndulo invertido rotacional, el cual integra las ventajas del conocimiento de un usuario experto, con las habilidades adaptativas de una red neuronal.

El sistema es diseñado para el control de los dos grados de libertad del péndulo, es decir, ángulo del péndulo y ángulo de la plataforma. Este péndulo invertido rotacional, al ser un sistema subactuado, está controlado únicamente por un motor DC a su vez conectado a un microcontrolador Arduino.

Para lograr el control, fue implementado un sistema neurodifuso, compuesto por dos módulos que usan las cuatro principales variables del sistema (los dos ángulos y sus variaciones), para luego ser éste entrenado usando aprendizaje por método híbrido. Para complementar el controlador, se unió este con un método de amoldamiento de energía para realizar el «swing-up» (balanceo vertical) del péndulo.

El controlador fue implementado y comparado con un regulador lineal cuadrático tradicional (LQR), demostrando tener un desempeño similar a este último. La ventaja principal radica en la fácil adaptación del controlador ante variaciones del sistema.

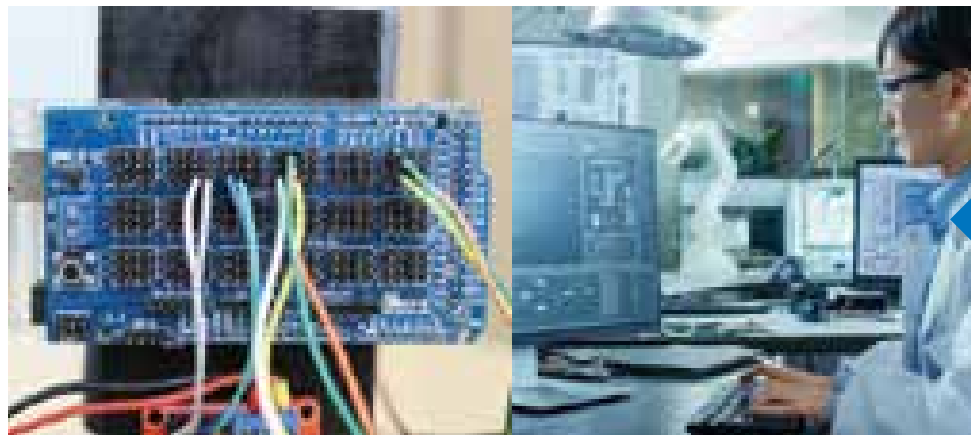
ABSTRACT

This article describes the design of a fuzzy controller for an inverted rotational pendulum, which adds the advantages of the knowledge of an experimented user and the adaptive capabilities of neural networks.

The system is designed for two degrees control of the rotational pendulum, i.e. angle of the pendulum and angle of the platform; this rotational inverted pendulum, being a sub-acting system, is controlled only by a DC motor which is in turn connected to an Arduino microcontroller.

To achieve the control, it was implemented a two-module neuro-fuzzy system using the 4 main variables of the system (two angles and their variations) and then trained using hybrid learning. To complement the controller, it was coupled with an energy shaping method to swing up the pendulum.

The controller was implemented and compared with a traditional linear quadratic regulator (LQR) controller showing similar performance. The main advantage of the new controller is its adaptation to system variations.



Palabras clave

Péndulo invertido rotacional, controlador neurodifuso, aprendizaje híbrido, LQR.

Key words

Rotational inverted pendulum, neuro-fuzzy controller, hybrid learning, LQR.

INTRODUCCIÓN

El péndulo invertido es un sistema demostrativo clásico para la implementación de técnicas de control avanzado y no-lineal que es apreciado en muchas aplicaciones de la vida real [2]. El sistema está compuesto por una base móvil que provee de energía cinética al péndulo y el péndulo en sí mismo. El péndulo Furuta o rotacional [3] es un sistema subactuado de dos grados de libertad donde la actuación es manejada por un brazo o plataforma que rota en la dirección horizontal en una manera análoga al clásico sistema carro-barra (*cart-pole*), aunque con una arquitectura más compacta.

El sistema presenta un desafío para la ejecución de un seguimiento debido a su naturaleza subactuada y a su dinámica compleja, es decir, presenta una dinámica mucho más compleja que la de un sistema *cart-pole* [2]. Sin embargo, este es un sistema compacto que hereda sus principales características del *cart-pole*. El interés de tales sistemas puede ser principalmente apreciado en la robótica para el control del movimiento en brazos y piernas [9].

En este artículo, se presenta el diseño del controlador neurodifuso que posibilita el uso de conocimiento de usuario experto con la adaptabilidad y capacidad de aprendizaje de las redes neuronales [5]. En comparación con otras técnicas de control utilizadas frecuentemente en este mecanismo, tales como los controladores PID [7] o el controlador por regímenes deslizantes [4], que pueden ser complicados de sintonizar para la estabilización de ambos ejes, el controlador neurodifuso es capaz de efectuar la tarea de estabilización incluso sin entrenamiento previo y para diferentes mecanismos, pues no requiere de un modelo del sistema para su implementación. Contrariamente a un controlador neuronal [6], este puede basarse además en la inferencia lógica. Por otra parte, la implementación de estructuras neurodifusas tradicionales, como el ANFIS [1], pueden resultar muy complejas.

Hemos integrado nuestro controlador con un algoritmo de amoldamiento de energía [9], y luego realizado un análisis comparativo con un regulador lineal cuadrático óptimo [8]. El objetivo de este procedimiento fue el de ilustrar la implementación pedagógica de controladores neuronales adaptativos en el laboratorio de control avanzado del Instituto Superior Tecnológico Tecsup.

El artículo está dividido en las siguientes secciones: primero, explicaremos el modelado del sistema y establecemos el algoritmo de amoldamiento de energía a usar para el balanceo vertical del péndulo a la posición vertical; luego, describiremos la metodología usada para implementar el sistema neurodifuso y el proceso de entrenamiento, así como los detalles de su diseño; también, mostraremos el sistema simulado e implementado, finalmente se presentan las conclusiones obtenidas.

FUNDAMENTOS

La estructura mecánica del péndulo rotacional es representada en la figura 1, el torque (τ) es aplicado al brazo horizontal. Este se moverá positivamente en la dirección horaria, generando un ángulo β con respecto al origen. El segundo brazo es el péndulo,

el cual rota libremente en la dirección vertical, el ángulo de este brazo con la vertical es denominado θ .

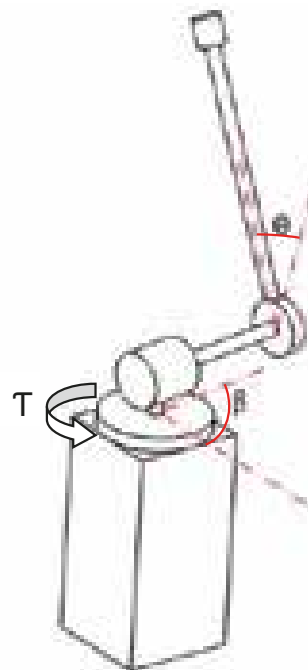


Figura 1. Estructura mecánica del péndulo Furuta.

Fuente: Elaboración propia.

Las variables de estado del sistema elegidos son el ángulo de la base (β) y el péndulo (θ) denominando así:

$$x = \begin{bmatrix} \beta \\ \theta \end{bmatrix}$$

La dinámica del sistema en forma de Euler-Lagrange es la siguiente:

$$D(x)\ddot{x} + H(\dot{x}, x) + G(x) = B(x)u \quad (1)$$

Donde las matrices son dadas como:

$$D = \begin{bmatrix} I_o + m_1(L_o^2 + l_1^2 \sin^2 \theta) & m_1 l_1 L_o \cos \theta \\ m_1 l_1 L_o \cos \theta & J_1 + m_1 l_1^2 \end{bmatrix}$$

$$H = \begin{bmatrix} m_1 l_1^2 \sin 2\theta \dot{\theta} \dot{\beta} - m_1 l_1 L_o \sin \theta \dot{\theta}^2 \\ -0.5 m_1 l_1^2 \sin 2\theta \dot{\beta}^2 \end{bmatrix}$$

$$G = \begin{bmatrix} 0 \\ -m_1 g l_1 \sin \theta \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Para realizar el balanceo vertical del péndulo, el sistema usa el algoritmo de amoldamiento de energía que ha sido propuesto en [9]. La energía total del sistema es:

$$E = E_k + E_p$$

Donde E_k la energía cinética y E_p la energía potencial. La primera es generada por la base y transmitida al péndulo, y se describe en la ecuación (2), mientras que la energía potencial es identificada únicamente por el ángulo del péndulo en (3).

$$E_k = 0.5 \dot{x}^T D \dot{x} \quad (2)$$

$$E_p = m_1 g l_1 \cos \theta \quad (3)$$

La energía deseada del sistema en la posición debe ser energía potencial pura. Para ello, se define la diferencial de energía como:

$$\dot{E} = 0.5 \dot{x}^T D \dot{x} + m_1 g l_1 (\cos\theta - 1) \quad (4)$$

El comando por amoldamiento de energía [3] se extrae de la evaluación de la función de Lyapunov en (5) propuesta en [9].

$$V(\dot{x}, x) = \frac{k_E}{2} \dot{E}^2 + \frac{k_\beta}{2} \dot{\beta} + \frac{k_B}{2} \beta \quad (5)$$

La solución para (4) con la función de Lyapunov presentada antes permite encontrar la ley de control en (6):

$$u = \frac{k_w B^T D^{-1} (H + G) - k_o * \beta - k_d * \dot{\beta}}{k_e E + k_w B^T D^{-1} B} \quad (6)$$

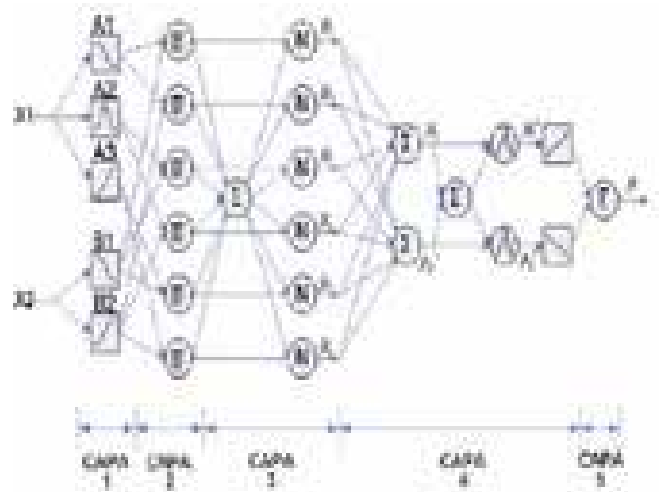


Figura 2. Estructura de ANFIS.
Fuente: Adaptado de [5].

METODOLOGÍA

Diseño del controlador neurodifuso

El controlador es compuesto por una serie de acciones coordinadas por un árbitro, las cuales son implementadas a través de redes neurodifusas basadas en la estructura ANFIS [5]. En esta sección, la estructura ANFIS es presentada en A; el diseño del sistema de control, en B; y el procedimiento de entrenamiento, en C.

A. Redes ANFIS

Un esquema de red neurodifusa es usado para implementar la lógica de los comandos. La estructura empleada es adaptada de [5] y es presentada en la figura 2. Se compone como sigue.

Capa 1: Neuronas implementando las funciones de pertenencia:

$$O_{1,i}^A = A_i(x_A); O_{1,i}^B = B_i(x_B) \quad (7)$$

Capa 2: Nodos producto, que calculan la componente para cada regla:

$$O_{2,i} = O_{1,n}^A \cdot O_{1,m}^B \quad (8)$$

Capa 3: Capa de normalización:

$$O_{3,i} = \frac{O_{2,i}}{\sum_{i=1}^N O_{2,i}} \quad (9)$$

Capa 4: Nodos adaptativos para generar la consecuencia de cada regla:

$$O_{4,i} = O_{3,i} \cdot f_n \quad (10)$$

Capa 5: Suma de señales de-difusificadas para obtener la salida de la red:

$$O_{5,i} = \sum_{i=1}^M O_{4,i} \quad (11)$$

B. Parte difusa del controlador

El controlador se diseñó para el control de la inclinación del péndulo y del brazo horizontal. Para el control de ambas dinámicas, necesitamos considerar todos los estados. Ello, infortunadamente, podría requerir muchas reglas debido a todas las posibles combinaciones entre las cuatro variables [1], haciendo así el diseño muy tedioso.

Para evitar ello, se realizó la separación del controlador en dos módulos. El primer módulo da la corrección para la dinámica de los ángulos (θ , β); mientras que el segundo módulo, para la dinámica de las velocidades ($\dot{\theta}$, $\dot{\beta}$). Cada uno es conformado por un sistema con tres funciones de pertenencia por entrada y nueve reglas de inferencia.

Comparado con el método clásico de considerar la forma PD en la inferencia difusa (e.g. θ y $\dot{\theta}$ como entradas), la estrategia usada permite acoplar mejor las acciones y por ello la fusión de ambos módulos puede realizarse con una simple adición (Figura 3). Se puede decir que el primer módulo genera la acción de corrección para el estado presente y el segundo módulo la acción para la predicción.

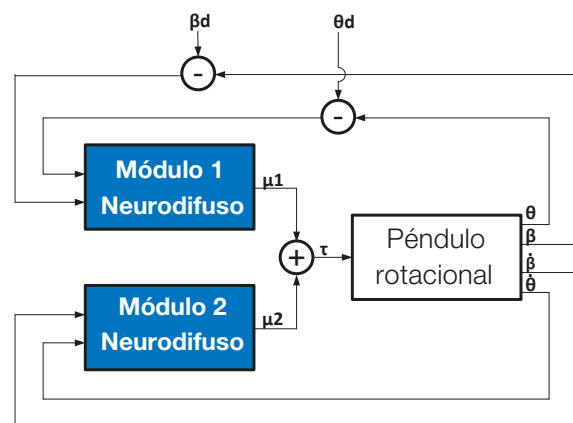


Figura 3. El diseño del controlador neurodifuso.
Fuente: Elaboración propia.

Se elaboraron las siguientes reglas de inferencia para el módulo 1:

- Si θ es NE y β es NE, entonces μ_1 es NE.
- Si θ es NE y β es CE, entonces μ_1 es NE.
- Si θ es NE y β es PO, entonces μ_1 es PO.
- Si θ es CE y β es NE, entonces μ_1 es NE.
- Si θ es CE y β es CE, entonces μ_1 es CE.
- Si θ es CE y β es PO, entonces μ_1 es PO.
- Si θ es PO y β es NE, entonces μ_1 es NE.
- Si θ es PO y β es CE, entonces μ_1 es PO.
- Si θ es PO y β es PO, entonces μ_1 es PO.

Y las siguientes reglas de inferencia para el módulo 2:

- Si $\dot{\theta}$ es NE y $\dot{\beta}$ es NE, entonces μ_2 es NE.
- Si $\dot{\theta}$ es NE y $\dot{\beta}$ es CE, entonces μ_2 es NE.
- Si $\dot{\theta}$ es NE y $\dot{\beta}$ es PO, entonces μ_2 es PO.
- Si $\dot{\theta}$ es CE y $\dot{\beta}$ es NE, entonces μ_2 es CE.
- Si $\dot{\theta}$ es CE y $\dot{\beta}$ es CE, entonces μ_2 es CE.
- Si $\dot{\theta}$ es CE y $\dot{\beta}$ es PO, entonces μ_2 es CE.
- Si $\dot{\theta}$ es PO y $\dot{\beta}$ es NE, entonces μ_2 es NE.
- Si $\dot{\theta}$ es PO y $\dot{\beta}$ es CE, entonces μ_2 es PO.
- Si $\dot{\theta}$ es PO y $\dot{\beta}$ es PO, entonces μ_2 es PO.

Donde los variables lingüísticas son las siguientes: Negativo (NE), Cero (CE) y Positivo (PO).

C. Entrenamiento

El entrenamiento de la red es realizado usando el algoritmo de propagación dinámica en retroceso (*Dynamic back-propagation* o DBP). Por ello, las siguientes se hacen las adaptaciones en las capas 1 y 4 de la red.

El algoritmo DBP es gobernado por el cálculo de la variación en la función de costo usando la ecuación (12) para el caso de los pesos v , donde la recursividad se complementa usando la ecuación (13) como valor inicial.

$$\frac{dJ}{dv} = \frac{dx_i du}{du dv} + \left(\frac{dx_i du}{du dx} + \frac{dx_i}{dx_{i-1}} \right) \bar{dx}_{i-1} \quad (12)$$

$$\frac{\bar{dx}_1}{dv} = \frac{dx_1 du}{du dv} \quad (13)$$

Para el caso de los pesos en la capa 2 de nuestra red, la ecuación DBP es obtenida usando las derivadas dadas en la ecuación (14).

$$\frac{du}{dv} = \frac{du}{dO_5} \frac{dO_5}{dO_4} \frac{dO_4}{dO_3} \frac{dO_3}{dO_2} \frac{dO_2}{dv} \quad (14)$$

$$\frac{dx_{i-1}}{dv} = \frac{dx_{i-1}}{dO_5} \frac{dO_5}{dO_4} \frac{dO_4}{dO_3} \frac{dO_3}{dO_2} \frac{dO_2}{dv}$$

La adaptación en la capa 5 es más simple, pero nosotros preferimos el método híbrido de entrenamiento, debido a su mayor rapidez de convergencia [5]. El algoritmo de mínimos cuadrados es, por tanto, configurado según (15).

$$dw = (A^T A)^{-1} A^T b \quad (15)$$

Donde A es la matriz que relaciona w con la salida en la capa 5 y b son los errores retropropagados hasta esta misma capa (por cada iteración).

El entrenamiento se puede realizar para ambos módulos al mismo tiempo, usando el mismo procedimiento anterior, pero

con las matrices aumentadas correspondientemente para la consideración de ambos sistemas.

RESULTADOS

Para validar el diseño, una simulación fue realizada con los siguientes valores: $m_1 = 0.1$ kg, $L_0 = 0.05$ m., $l_1 = 0.1$ m, $l_0 = 6 \times 10^{-5}$ kg.m², $J_1 = 4 \times 10^{-5}$ kg.m². La posición inicial para estos ensayos es $\theta = 2.5\pi/3$ y $\beta = -\pi/2$. Para iniciar el proceso, la base de reglas del sistema de inferencia y una configuración básica son configuradas tal que establezcan el sistema. La respuesta a esta configuración inicial se ilustra en la figura 4. El sistema converge al ángulo deseado, aunque presenta una oscilación apreciable en la base del ángulo.

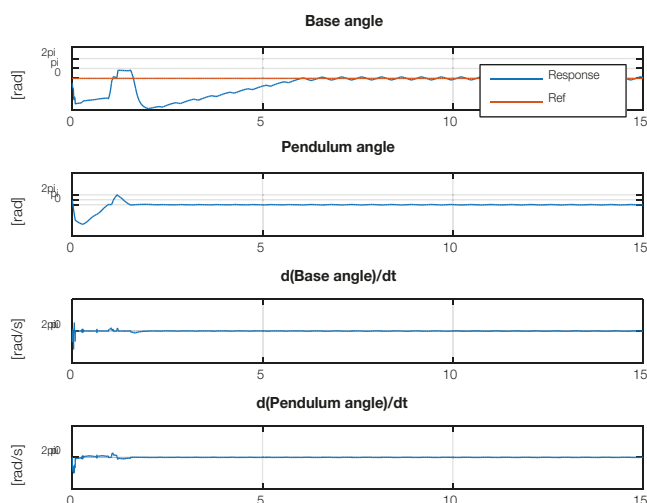


Figura 4. Simulación por el controlador neuro difuso sin entrenamiento. Fuente: Elaboración propia.

Una mejora adicional es conseguida usando el proceso de entrenamiento híbrido para la red. La respuesta para este caso se muestra en la figura 5. Esta última respuesta muestra la mejora de la estabilidad y rapidez del sistema. Además, en la figura 6 se presenta el desempeño ante variaciones en la masa, test que no puede ser replicado con LQR, pues requiere un rediseño de sus ganancias.

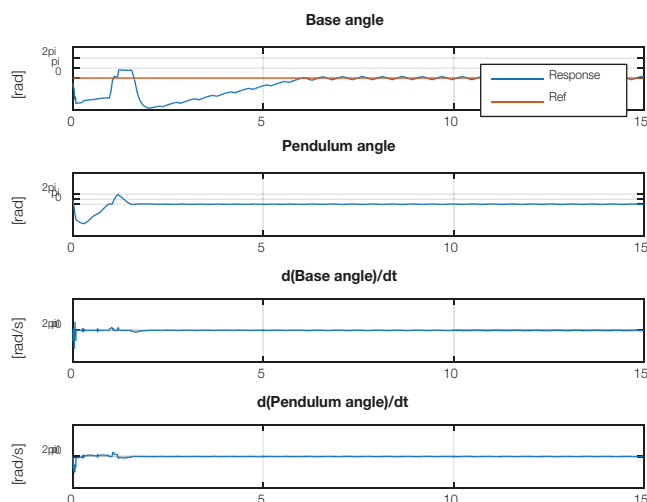


Figura 5. Simulación por el controlador neuro difuso con entrenamiento. Fuente: Elaboración propia.

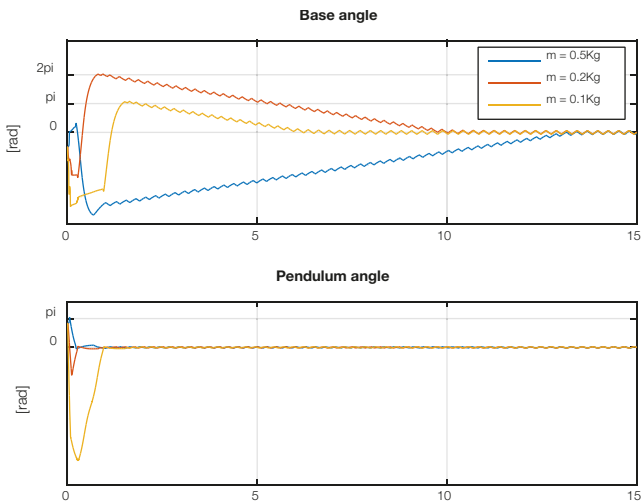


Figura 6. Simulación por el controlador neuro difuso con diferentes masas del péndulo.
Fuente: Elaboración propia.

Para validar el trabajo, se realizó la implementación usando un péndulo invertido rotacional (Figura 7). El actuador es un motor DC que opera la plataforma rotativa. Se dispone también de dos *encoders* para la medición de los ángulos, los algoritmos se implementaron en un microcontrolador Arduino, donde se programaron el LQR [8] y el controlador propuesto, el cual fue entrenado previamente fuera de línea. Los resultados del sistema para la estabilización se muestran en la figura 8.

Además, una comparación adicional para el caso de seguimiento es presentado en la figura 9, donde se usa la señal de test $\sin(t) - 0.2 \sin(4t)$. Cuando el controlador neurodifuso es empleado, se tiene una dinámica más variante, pero una mayor velocidad de convergencia.



Figura 7. Péndulo invertido rotacional.
Fuente: Elaboración propia.

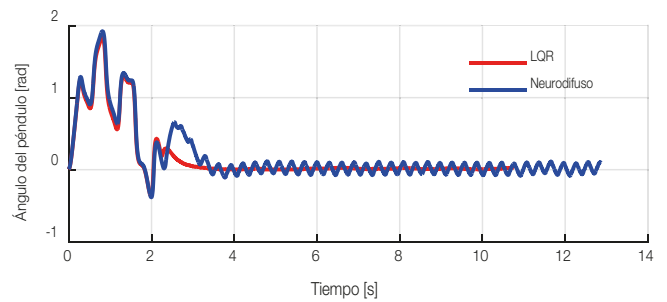
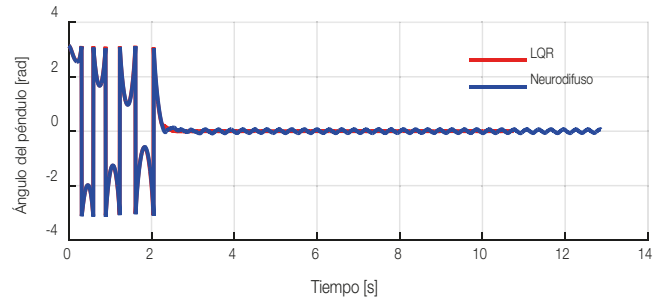


Figura 8. Balanceo vertical del péndulo.
Fuente: Elaboración propia.

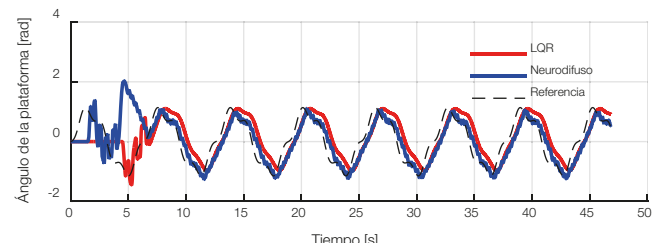
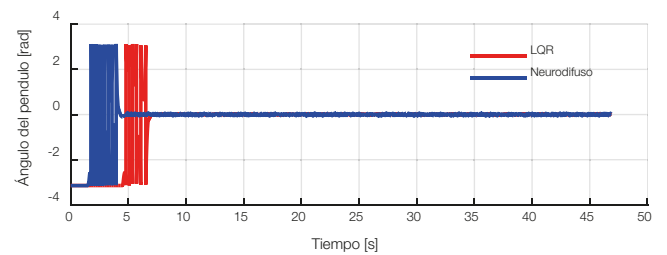


Figura 9. Ensayo de seguimiento con ambos controladores, rojo: LQR, azul: red neurodifusa.
Fuente: Elaboración propia.

CONCLUSIONES

- El controlador neurodifuso fue implementado y demostrado satisfactoriamente en el péndulo invertido rotacional. Pudimos comprobar el buen desempeño de este controlador, pues puede ser comparado a la implementación de un LQR en términos de rapidez y estado transitorio.
- El controlador diseñado tiene como principales ventajas su fácil diseño, ya que se basa en reglas lógicas simples. Su capacidad adaptativa es mayor, pues puede ser entrenado, y mantiene su aplicabilidad aun ante variaciones grandes en los parámetros del sistema, como la masa del péndulo sin la necesidad de un rediseño.

REFERENCIAS

- [1] Agrawal, R., y Mitra, R. (2013). Adaptive Neuro Fuzzy Inference Structure Controller for Rotary Inverted Pendulum. *Proceedings of International Conference on Advances in Computing*. Springer: New Delhi, 1163-1170.
- [2] Åström, K., y Furuta, K. (1996). Swinging up a pendulum by energy control. *IFAC Proceedings volumes*. 29(1), 1919-1924.
- [3] Furuta, K., Yamakita, M., y Kobayashi, S. (1992). Swing-up control of inverted pendulum using pseudo-state feedback. *Journal of Systems and Control Engineering*. 206(49), 263–269.
- [4] Grossimon, P., Barbieri, E., y Drakunov, S. (1996). Sliding mode control of an inverted pendulum. *Proceedings of 28th Southeastern Symposium on System Theory*, Baton Rouge, Los Angeles, 248-252.
- [5] Jang, J. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems Man and Cybernetics*. 23, 665-685.
- [6] Moreno-Valenzuela, J., Aguilar-Avelar, C., Puga-Guzmán, S., y Santibáñez, V. (2016) Adaptive Neural Network Control for the Trajectory Tracking of the Furuta Pendulum. *IEEE Transactions on Cybernetics*. 46(12), 3439-3452.
- [7] Rani, M., Selamat, H., Zamzuri, H., Ahmad, F. (2011). PID controller optimization for a rotational inverted pendulum using genetic algorithm. *Fourth International Conference on Modeling, Simulation and Applied Optimization*, Kuala Lumpur, 1-6.

[8] Siva, M., Dasu, B., y Ramesh, G. (1995). Design of LQR based stabilizer for rotary inverted pendulum system. *IJCTA*. 9 (29), 21-29.

[9] Tanaka, S., Xin, X., y Yamasaki, T. (2001). New results of energy based swing-up control for a rotational pendulum. *SICE Journal of control, measurement, and system integration*. 4 (6), 394-400.

ACERCA DE LOS AUTORES

Luis Enciso Salas

Graduado con honores como bachiller en Ingeniería Electrónica. Es magíster en Control y Automatización en la Pontificia Universidad Católica del Perú y en Sistemas Espaciales en ISAE-SUPAERO (Francia). Con experiencia en la industria de la automatización y desarrollando actividades en docencia e investigación.

@ lenciso@pucp.pe

Miguel Angel Chávez Luna

Profesional en Electrónica y Sistemas Computarizados por Tecsup. Bachiller en Ingeniería Industrial por la Universidad Antonio Ruiz de Montoya. Ha realizado especializaciones en Adquisición de Datos y Control por Computadora en EE. UU., Ingeniería Mecatrónica en México y Brasil, y Sistemas de Automatización en Alemania. Posee certificación *Train the Trainer* dada por Bosch Rexroth de Alemania en Sistemas de Automatización e Industria 4.0. Miembro de IEEE en la sección de Robótica y Automatización. Profesor a tiempo completo en Tecsup, dicta cursos de Automatización Industrial y Sistemas Mecatrónicos.

@ mchavez@tecsup.edu.pe