

Control Descentralizado Neuro-Adaptable Indirecto con entrenamiento UKF aplicado a Robótica ^{*}

J. F. Guerra Cano ^{*} R. García-Hernández ^{*}
M. A. Llama Leal ^{*}

^{*} Tecnológico Nacional de México/I. T. de La Laguna
División de Estudios de Posgrado e Investigación, Torreón, México
(e-mail: [m.jfguerrac,rgarciah,mllama]@correo.itlalaguna.edu.mx)

Resumen: En este trabajo se presenta un controlador descentralizado para la solución al problema de seguimiento de trayectorias para un brazo robótico de siete grados de libertad (g.d.l.) por medio de un esquema de identificación y control neuronal. Utilizando redes neuronales recurrentes de alto orden (RHONN) discretas para aproximar las dinámicas del manipulador, se diseñan leyes de control que combinan el control por bloques y técnicas de modos deslizantes. El entrenamiento de la RHONN se realiza utilizando el filtro de Kalman *unscented* (UKF) y se compara con trabajos previos realizados con filtro de Kalman extendido (EKF).

Palabras clave: Filtros de Kalman, Control de robots, Redes neuronales recurrentes, Seguimiento de trayectoria, Manipuladores robóticos.

1. INTRODUCCIÓN

En las últimas décadas, la demanda tecnológica de la sociedad actual ha requerido nuevos enfoques a los problemas de control surgidos por la necesidad de crear sistemas cada vez más complejos y sofisticados. Las redes neuronales (RN), debido a su habilidad de aprendizaje y su capacidad de adaptarse, son una herramienta significativa en el modelado y control de este tipo de sistemas. El uso de las redes neuronales en la robótica, donde las necesidades de trabajar con sistemas no lineales de alto orden y encontrar modelos de control con un escaso conocimiento de la planta y de su medio ambiente, se aplica en la aproximación de la dinámica del sistema y, con base a este modelo, proponer una ley de control. Por esta razón, en el caso de robots manipuladores, es posible diseñar un control descentralizado (Siljak, 2011), compuesto por controladores independientes, basados en redes neuronales recurrentes que utilicen únicamente las variables locales para cada subsistema.

La teoría de control descentralizado para robots manipuladores ha sido previamente reportada en la literatura como en (Liu, 1999), el cual muestra la aplicación de un controlador descentralizado en manipuladores robóticos, y cada articulación con su respectivo eslabón se considera un subsistema. Por otro lado, en (Safaric y Rodic, 2000)

se reporta un control descentralizado para robots manipuladores utilizando redes neuronales “*feedforward*” para aproximar la dinámica de cada articulación de manera independiente.

Prosiguiendo con el diseño del controlador descentralizado, es necesario definir el tipo de entrenamiento para las redes neuronales recurrentes que identificarán las dinámicas de los subsistemas. El filtro de Kalman (KF) (Haykin, 2004) es una herramienta popular y efectiva como estimador de estados para sistemas lineales. En el caso de sistemas no lineales, existen bastantes variantes del KF para tratar la no linealidad del sistema; el filtro de Kalman extendido (EKF) (Kalman y Bucy, 1961) y el filtro de Kalman “*unscented*” (UKF) (Julier y Uhlmann, 1997) son los más utilizados. Existen trabajos y artículos donde se presentan el análisis, el estudio y la comparación entre los filtros EKF y UKF (Rhudy y Gu, 2013). Además, existen diversos trabajos sobre el uso de algoritmos de entrenamiento de redes neuronales basados en los filtros de Kalman no lineales. En (García-Hernández et al., 2009) y (García-Hernández et al., 2011), los autores presentan un identificador neuronal descentralizado discreto y un esquema de control desarrollado con RHONNs entrenadas con un algoritmo EKF implementado en tiempo real para un robot de 2 g.d.l. y la simulación en el brazo robótico Mitsubishi PA10-7CE de 7 g.d.l. respectivamente.

En (Guerra et al., 2019) se presenta un identificador neuronal descentralizado discreto y un esquema de control con distintos algoritmos de entrenamiento basados en

^{*} Los autores agradecen el apoyo financiero de los proyectos de investigación del Tecnológico Nacional de México/I. T. de La Laguna y del CONACYT 860723.

EKF y UKF para un robot de 2 g.d.l. En este trabajo se propone un esquema de control neuro-adaptable con entrenamiento UKF para el entrenamiento de las redes neuronales aplicado en el brazo robótico Mitsubishi PA10-7CE de 7 g.d.l. con el fin de comparar su desempeño con el entrenamiento EKF.

2. SISTEMA DESCENTRALIZADO DISCRETO

Considérese la siguiente clase de sistema no lineal en tiempo discreto, el cual se representa en la forma no lineal controlable por bloques (NBC) conformada por r bloques.

$$\begin{aligned} \chi_{i,k+1}^1 &= f_i^1(\chi_i^1) + B_i^1(\chi_i^1) \chi_i^2 + \Gamma_{i\ell}^1 \\ \chi_{i,k+1}^2 &= f_i^2(\chi_i^1, \chi_i^2) + B_i^2(\chi_i^1, \chi_i^2) \chi_i^3 + \Gamma_{i\ell}^2 \\ &\vdots \\ \chi_{i,k+1}^j &= f_i^j(\chi_i^1, \dots, \chi_i^j) + B_i^j(\chi_i^1, \dots, \chi_i^j) \chi_i^{j+1} + \Gamma_{i\ell}^j \\ &\vdots \\ \chi_{i,k+1}^r &= f_i^r(\chi_i) + B_i^r(\chi_i) u_i + \Gamma_{i\ell}^r \end{aligned} \quad (1)$$

donde $\chi_i \in \mathbb{R}^{n_i}$, $\chi_i^j \in \mathbb{R}^{n_{ij} \times 1}$, $i = 1, \dots, N$; $j = 1, \dots, r-1$; $\ell = 1, \dots, n_{ij}$, N es el número de subsistemas, $u_i \in \mathbb{R}^{m_i}$ es el vector de entradas. Suponiendo que f_i^j , B_i^j y $\Gamma_{i\ell}^j$ son funciones suaves y acotadas; además $f_i^j(0) = 0$ y $B_i^j(0) = 0$. $n_{i1} \leq n_{i2} \leq \dots \leq n_{ij} \leq m_i$ definen las diferentes estructuras de los subsistemas. Los términos de interconexión están dados por

$$\begin{aligned} \Gamma_{i\ell}^1 &= \sum_{\ell=1, \ell \neq i}^N \gamma_{i\ell}^1(\chi_\ell^1) \\ &\vdots \\ \Gamma_{i\ell}^r &= \sum_{\ell=1, \ell \neq i}^N \gamma_{i\ell}^r(\chi_\ell) \end{aligned} \quad (2)$$

donde $\chi_\ell = [\chi_\ell^1, \chi_\ell^2, \dots, \chi_\ell^j]^T$ representa el vector de estado del ℓ -ésimo subsistema con $1 \leq \ell \leq N$ y $\ell \neq i$.

2.1 Identificador neuronal descentralizado

La siguiente estructura descentralizada de RHONN discreta se propone para identificar a (1):

$$\begin{aligned} x_{i,k+1}^1 &= w_{i,k}^1 S(\chi_{i,k}^1) + w_{i,k}^1 \chi_{i,k}^2 \\ x_{i,k+1}^2 &= w_{i,k}^2 S(\chi_{i,k}^1, \chi_{i,k}^2) + w_{i,k}^2 \chi_{i,k}^3 \\ &\vdots \\ x_{i,k+1}^j &= w_{i,k}^j S(\chi_{i,k}^1, \chi_{i,k}^2, \dots, \chi_{i,k}^j) + w_{i,k}^j \chi_{i,k}^{j+1} \\ &\vdots \\ x_{i,k+1}^r &= w_{i,k}^r S(\chi_{i,k}^1, \dots, \chi_{i,k}^r) + w_{i,k}^r u_{i,k} \end{aligned} \quad (3)$$

donde $x_{i,k+1}^j = [x_i^1 \ x_i^2 \ \dots \ x_i^r]^T$ es el j -ésimo estado de la red neuronal con $i = 1, \dots, N$ y $j = 1, \dots, r-1$, $w_{i,k}^j$ son parámetros constantes, $rank(w_{i,k}^j) = n_{ij}$; $S(\bullet)$ es la función de activación y $u_{i,k}$ representa el vector de entradas.

2.2 Ley de aprendizaje en línea

Se propone un algoritmo de entrenamiento basado en UKF descrito para cada i -ésima neurona por

$$\begin{aligned} K_{i,k}^j &= P_{i,k}^{jxy} (P_{i,k}^{jyy})^{-1} \\ w_{i,k+1}^j &= w_{i,k}^j + K_{i,k}^j e_{i,k}^j \\ P_{i,k+1}^j &= P_{i,k}^j - K_{i,k}^j P_{i,k}^{jyy} K_{i,k}^{jT} \end{aligned} \quad (4)$$

con

$$\begin{aligned} P_{i,k}^{jxy} &= \sum_{i=0}^{2L} \eta_i^c [\mathcal{X}_{i,k|k-1}^j - \hat{x}_{i,k}^{j-}] [\mathcal{Y}_{i,k|k-1}^j - \hat{y}_{i,k}^{j-}]^T \\ P_{i,k}^{jyy} &= R_{i,k}^j + \sum_{i=0}^{2L} \eta_i^{jc} [\mathcal{Y}_{i,k|k-1}^j - \hat{y}_{i,k}^{j-}] [\mathcal{Y}_{i,k|k-1}^j - \hat{y}_{i,k}^{j-}]^T \\ P_{i,k}^j &= Q_{i,k}^j + \sum_{i=0}^{2L} \eta_i^{jc} [\mathcal{X}_{i,k|k-1}^j - \hat{x}_{i,k}^{j-}] [\mathcal{X}_{i,k|k-1}^j - \hat{x}_{i,k}^{j-}]^T \\ e_{i,k}^j &= [\mathcal{X}_{i,k}^j - x_{i,k}^j] \end{aligned} \quad (5)$$

donde $e_{i,k}^j$ es el error de identificación, $P_{i,k}^j$ es la matriz de covarianza del error de predicción, $P_{i,k}^{jyy}$ es la matriz de covarianza de predicción de la salida, $P_{i,k}^{jxy}$ es la matriz de covarianza cruzada entre el estado y la salida; $w_{i,k}^j$ es el j -ésimo peso (estado) del i -ésimo subsistema, η_i^{jc} es un parámetro de diseño, $\mathcal{X}_{i,k}^j$ es el j -ésimo estado de la planta y $x_{i,k}^j$ el j -ésimo estado de la red neuronal. L es el número de estados, $K_{i,k}^j$ es la matriz de ganancias de Kalman, $Q_{i,k}^j$ es la matriz de covarianza de ruido en la medición, $R_{i,k}^j$ es la matriz de covarianza de ruido en los estados, $\hat{x}_{i,k}^{j-}$ es la media de la predicción del estado, $\hat{y}_{i,k}^{j-}$ es la media de la predicción de salida, $\mathcal{X}_{i,k|k-1}^j$ y $\mathcal{Y}_{i,k|k-1}^j$ son la propagación de cada punto sigma a través de la predicción y la observación respectivamente. La técnica central del algoritmo UKF es la generación de los puntos sigma, que es la principal característica de la transformación unscented (por sus siglas en inglés UT).

$$\mathcal{X}_{i,k-1}^j = [w_{i,k-1}^j \quad w_{i,k-1}^j \pm \sqrt{L + \lambda} \sqrt{P_{i,k-1}^j}] \quad (6)$$

donde $\mathcal{X}_{i,k-1}^j \in \mathbb{R}^{L \times 2L+1}$ es una matriz de puntos sigma, en la cual cada columna representa un punto sigma. Una vez que se han generado los puntos sigma, cada punto pasa a través de funciones no lineales y son usados para calcular

$$\begin{aligned} \mathcal{X}_{i,k|k-1}^j &= F(\mathcal{X}_{i,k-1}^j, u_{i,k-1}) \\ \mathcal{Y}_{i,k|k-1}^j &= H(\mathcal{X}_{i,k-1}^j, u_{i,k-1}) \\ \hat{x}_{i,k}^{j-} &= \sum_{i=0}^{2L} \eta_i^{jm} \mathcal{X}_{i,k|k-1}^j \\ \hat{y}_{i,k}^{j-} &= \sum_{i=0}^{2L} \eta_i^{jm} \mathcal{Y}_{i,k|k-1}^j \end{aligned} \quad (7)$$

donde F es una función no lineal de predicción del estado, H es una función no lineal de predicción de la salida; λ

es un parámetro de escalamiento, y los vectores de peso η_i^{jm} (promedio) y η_i^{jc} (covarianza) se definen por

$$\begin{aligned} \lambda &= \alpha^2(L + \kappa) - L \\ \eta_0^{jm} &= \lambda/(L + \lambda) \\ \eta_0^{jc} &= \lambda/(L + \lambda) + 1 - \alpha^2 + \beta \\ \eta_i^{jm} &= \eta_i^{jc} = \lambda/[2(L + \lambda)] \\ i &= 1, \dots, 2L \end{aligned}$$

donde α , β y κ son parámetros de propagación, distribución y ajuste para reducir los errores de predicción (Julier y Uhlmann, 1997; Haykin, 2004; Rhudy y Gu, 2013).

3. DISEÑO DEL CONTROLADOR

Para realizar el seguimiento de la trayectoria del primer bloque en (1), es necesario definir el error de seguimiento como

$$\mathbf{z}_{i,k}^1 = x_{i,k}^1 - x_{id,k}^1 \quad (8)$$

donde $x_{id,k}^1$ es la i -ésima señal de la trayectoria deseada y $x_{i,k}^1$ es el i -ésimo estado de la red neuronal, con $i = 1, \dots, N$ subsistemas.

A partir de (8) se determina el siguiente valor como

$$\mathbf{z}_{i,k+1}^1 = w_{i,k}^1 S(\chi_{i,k}^1) + w_i'^1 \chi_{i,k}^2 - x_{id,k+1}^1 \quad (9)$$

la ecuación (9) es vista como un bloque con el estado $\mathbf{z}_{i,k}^1$ y el estado $\chi_{i,k}^2$ es considerado como una entrada de pseudo-control, donde es posible imponer una dinámica deseada. Esta ecuación se puede resolver con la anticipación de la dinámica deseada para este bloque como

$$\begin{aligned} \mathbf{z}_{i,k+1}^1 &= w_{i,k}^1 S(\chi_{i,k}^1) + w_i'^1 \chi_{i,k}^2 - x_{id,k+1}^1 \\ &= k_i^1 \mathbf{z}_{i,k}^1 \end{aligned} \quad (10)$$

donde $|k_i^1| < 1$, con el fin de garantizar la estabilidad de (10). $x_{id,k}^2$ se calcula como

$$x_{id,k}^2 = \frac{1}{w_i'^1} [-w_{i,k}^1 S(\chi_{i,k}^1) + \chi_{id,k+1}^1 + k_i^1 \mathbf{z}_{i,k}^1]$$

hay que tener en cuenta que el valor calculado del estado $x_{id,k}^2$ no es su valor real; en lugar de ello, representa el comportamiento deseado para $\chi_{i,k}^2$.

De la misma manera que para el primer bloque, se puede definir una segunda variable en las nuevas coordenadas como:

$$\mathbf{z}_{i,k}^2 = x_{i,k}^2 - x_{id,k}^2 \quad (11)$$

el siguiente valor para $\mathbf{z}_{i,k}^2$ produce

$$\mathbf{z}_{i,k+1}^2 = x_{i,k+1}^2 - x_{id,k+1}^2 \quad (12)$$

donde se impone la dinámica deseada para este bloque como

$$\begin{aligned} \mathbf{z}_{i,k+1}^2 &= w_{i,k}^2 S(\chi_{i,k}^1, \chi_{i,k}^2) + w_i'^2 \chi_{i,k}^3 - x_{id,k+1}^2 \\ &= k_i^2 \mathbf{z}_{i,k}^2 \end{aligned} \quad (13)$$

con $|k_i^2| < 1$. De forma iterativa, se toman estos pasos, en donde en el último paso la variable deseada conocida es $x_{id,k}^r$, y la última nueva variable se define como

$$\mathbf{z}_{i,k}^r = x_{i,k}^r - x_{id,k}^r$$

finalmente, el siguiente valor se obtiene como

$$\mathbf{z}_{i,k+1}^r = w_{i,k}^r S(\chi_{i,k}^1, \dots, \chi_{i,k}^r) + w_i'^r u_{i,k} - x_{id,k+1}^r \quad (14)$$

El sistema (3) se puede representar en las nuevas variables $\mathbf{z} = [z_i^{1T} \ z_i^{2T} \ \dots \ z_i^{rT}]$ como

$$\mathbf{z}_{i,k+1}^{r-1} = k_i^{r-1} \mathbf{z}_{i,k}^{r-1} + w_i'^{r-1} \mathbf{z}_{i,k}^r \quad (15)$$

$$\mathbf{z}_{i,k+1}^r = w_{i,k}^r S(\chi_{i,k}^1, \dots, \chi_{i,k}^r) + w_i'^r u_{i,k} - x_{id,k+1}^r$$

Implementando un control por modos deslizantes, se tiene que, cuando la entrada de control está limitada por u_{0i}

$$|u_{i,k}| \leq u_{0i} \quad (16)$$

por lo que se diseña una superficie de deslizamiento y una ley de control, que lleve los estados hacia tal superficie. Para tal efecto, la superficie se selecciona como $S_{D_{i,k}} = \mathbf{z}_{i,k}^r = 0$; entonces, el sistema (16) se reescribe de la siguiente manera:

$$S_{D_{i,k+1}} = w_{i,k}^r S(\chi_{i,k}^1, \dots, \chi_{i,k}^r) + w_i'^r u_{i,k} - x_{id,k+1}^r \quad (17)$$

con la superficie deslizante definida, el siguiente paso es encontrar una ley de control que tome a consideración el acotamiento de (17), por lo tanto, el control $u_{i,k}$ se selecciona (Utkin et al., 2009) de la siguiente manera

$$u_{i,k} = \begin{cases} u_{eq_{i,k}} & \text{for } \|u_{eq_{i,k}}\| \leq u_{0i} \\ u_{0i} \frac{u_{eq_{i,k}}}{\|u_{eq_{i,k}}\|} & \text{for } \|u_{eq_{i,k}}\| > u_{0i} \end{cases} \quad (18)$$

donde el control equivalente $u_{eq_{i,k}}$ se obtiene a partir de $S_{D_{i,k+1}} = 0$ como

$$u_{eq_{i,k}} = \frac{1}{w_i'^r} [-w_{i,k}^r S(\chi_{i,k}^1, \dots, \chi_{i,k}^r) + x_{id,k+1}^r] \quad (19)$$

El esquema completo de identificación y control propuesto para el sistema se muestra en la Figura 1.

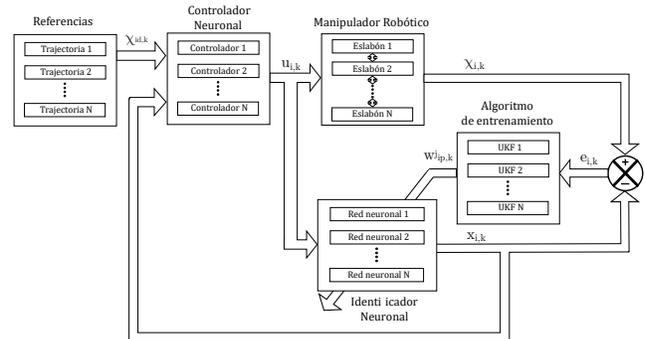


Figura 1. Esquema de identificación y control neuronal con entrenamiento UKF

4. ROBOT MITSUBISHI PA10-7CE

4.1 Descripción del robot

El robot manipulador industrial Mistubishi PA10-7CE es una de las versiones del “Brazo Inteligente Portátil de Propósito General (Portable General-Purpose Intelligent Arm PA10)” de arquitectura abierta desarrollado por Mitsubishi Heavy Industries (MHI). Esta versión se encuentra disponible en el Laboratorio de Mecatrónica y Control del Instituto Tecnológico de la Laguna, y está compuesto por siete articulaciones conectadas a través de eslabones tal y como se ilustra en la Figura 2.

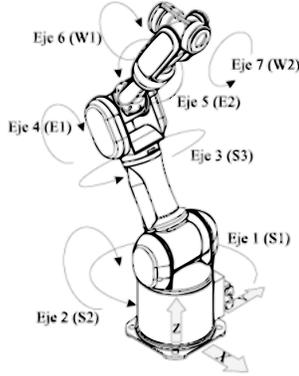


Figura 2. Estructura del manipulador PA10-7CE

En la Tabla 1 se muestran las consignas de par máximo que se pueden aplicar a cada una de las articulaciones

Tabla 1. Pares máximos de las articulaciones

Articulación	Pares Max.
S1	$\pm 232 \text{ Nm}$
S2	$\pm 232 \text{ Nm}$
S3	$\pm 100 \text{ Nm}$
E1	$\pm 100 \text{ Nm}$
E2	$\pm 14.5 \text{ Nm}$
W1	$\pm 14.5 \text{ Nm}$
W2	$\pm 14.5 \text{ Nm}$

4.2 Objetivo de Control

El objetivo es seguir una señal de referencia, diseñando una ley de control que combina control por bloques y técnicas de modos deslizantes. El error de seguimiento se define como

$$\mathbf{z}_{i,k}^1 = x_{i,k}^1 - x_{id,k}^1 \quad (20)$$

donde $x_{id,k}^1$ es la señal de la trayectoria deseada. Usando (3) e imponiendo la dinámica deseada para $\mathbf{z}_{i,k}^1$ se tiene que

$$\begin{aligned} \mathbf{z}_{i,k+1}^1 &= w_{i1,k}^1 S(\chi_{i,k}^1) + w_i'^2 \chi_{i,k}^2 - x_{id,k+1}^1 \\ &= k_i^1 \mathbf{z}_{i,k+1}^1 \end{aligned} \quad (21)$$

donde el valor deseado $x_{id,k+1}^1$ para $\chi_{i,k}^2$ se calcula a partir de (22) como

$$x_{id,k+1}^2 = \frac{1}{w_i'} [-w_{i1,k}^1 S(\chi_{i,k}^1) + \chi_{id,k+1}^1 + k_i^1 \mathbf{z}_{i,k}^1] \quad (22)$$

en el siguiente paso, se necesita definir una nueva variable como

$$\mathbf{z}_{i,k}^2 = x_{i,k}^2 - x_{id,k}^2 \quad (23)$$

el siguiente valor se obtiene como

$$\begin{aligned} \mathbf{z}_{i,k+1}^2 &= w_{i2,k}^2 S(\chi_{i,k}^1) + w_{i2,k}^2 S(\chi_{i,k}^2) + w_i'^2 \chi_{i,k}^3 - x_{id,k+1}^2 \\ &= k_i^2 \mathbf{z}_{i,k}^2 \end{aligned} \quad (24)$$

la superficie de deslizamiento se elige como $S_{D_i,k} = z_{i,k}^2 = 0$. La ley de control está dada por

$$u_{i,k} = \begin{cases} u_{eqi,k} & \text{for } \|u_{eqi,k}\| \leq \tau_i^{\max} \\ u_{0i} \frac{u_{eqi,k}}{\|u_{eqi,k}\|} & \text{for } \|u_{eqi,k}\| > \tau_i^{\max} \end{cases} \quad (25)$$

donde el $u_{eqi,k}$ se calcula a partir de $S_{D_i,k+1} = 0$ como

$$u_{eqi,k} = \frac{1}{w_i'} [-w_{i1,k}^2 S(\chi_{i,k}^1) + w_{i2,k}^2 S(\chi_{i,k}^2) + x_{id,k+1}^2] \quad (26)$$

y las entradas de control están acotadas por $\pm \tau_i^{\max}$.

5. RESULTADOS EN SIMULACIÓN

Para la simulación del controlador descentralizado neuronal por bloques (por sus siglas en inglés DNBC), se eligieron las trayectorias discretas propuestas en (García-Hernández et al., 2011)

$$\begin{aligned} x_{1d,k}^1 &= c_1(1 - e^{d_1 k T^3}) \sin(\omega_1 k T) [^\circ] \\ x_{2d,k}^1 &= c_2(1 - e^{d_2 k T^3}) \sin(\omega_2 k T) [^\circ] \\ x_{3d,k}^1 &= c_3(1 - e^{d_3 k T^3}) \sin(\omega_3 k T) [^\circ] \\ x_{4d,k}^1 &= c_4(1 - e^{d_4 k T^3}) \sin(\omega_4 k T) [^\circ] \\ x_{5d,k}^1 &= c_5(1 - e^{d_5 k T^3}) \sin(\omega_5 k T) [^\circ] \\ x_{6d,k}^1 &= c_6(1 - e^{d_6 k T^3}) \sin(\omega_6 k T) [^\circ] \\ x_{7d,k}^1 &= c_7(1 - e^{d_7 k T^3}) \sin(\omega_7 k T) [^\circ] \end{aligned} \quad (27)$$

donde los parámetros c_i , d_i y ω_i para las trayectorias deseadas de cada articulación se encuentran en la Tabla 2. El periodo de muestreo es de $T = 1 \text{ ms}$.

Tabla 2. Parámetros de trayectorias deseadas

Eslabón	c_i	d_i	ω_i
1	90	-0.001	16.3292°/s
2	60	-0.001	24.9236°/s
3	90	-0.01	31.7991°/s
4	60	-0.01	36.9557°/s
5	90	-0.01	19.7670°/s
6	60	-0.01	35.2369°/s
7	90	-0.01	26.6425°/s

Las trayectorias seleccionadas en (27) permiten evaluar el rendimiento en presencia de señales periódicas relativamente rápidas para las cuales las no linealidades de la dinámica del robot son realmente importantes y presentan

un término que crece sin problemas para mantener el robot en un estado de operación sin saturar los actuadores.

Las condiciones iniciales para la planta son iguales a cero para todas las articulaciones; en el caso del identificador neuronal las condiciones iniciales son 10° para todas las articulaciones. Los resultados de la simulación se muestran en las Figuras 3 a 9. De acuerdo con estas figuras los errores de seguimiento para todas las articulaciones presentan un buen comportamiento y permanecen limitados como se muestra en la Figura 11. Los pares aplicados se muestran en la Figura 10. Se observa fácilmente que las señales de control siempre se encuentran dentro de los límites de los actuadores $\pm\tau_1^{max}$ a $\pm\tau_7^{max}$, respectivamente.

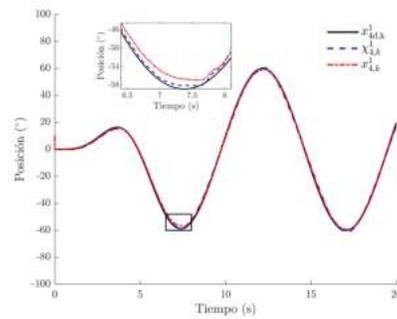


Figura 6. Identificación y seguimiento eslabón 4

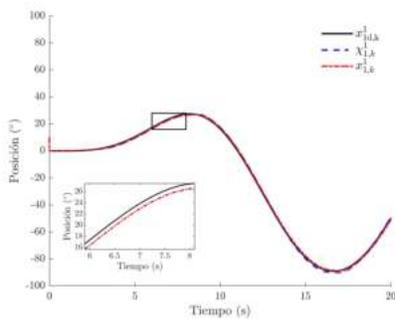


Figura 3. Identificación y seguimiento eslabón 1

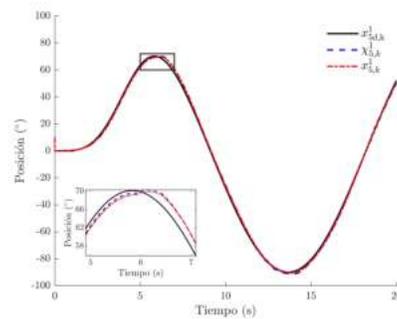


Figura 7. Identificación y seguimiento eslabón 5

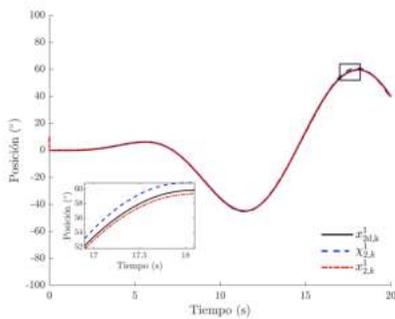


Figura 4. Identificación y seguimiento eslabón 2

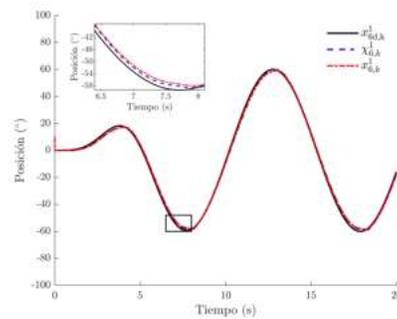


Figura 8. Identificación y seguimiento eslabón 6

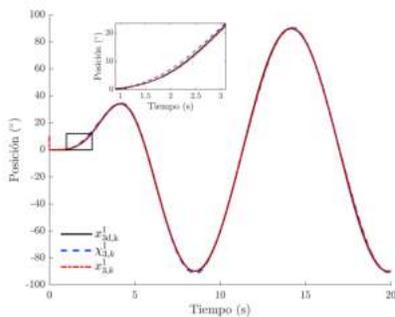


Figura 5. Identificación y seguimiento eslabón 3

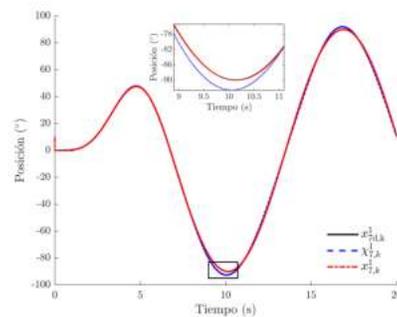


Figura 9. Identificación y seguimiento eslabón 7

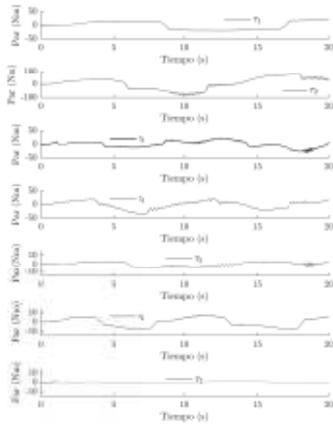


Figura 10. Pares aplicados eslabones 1-7

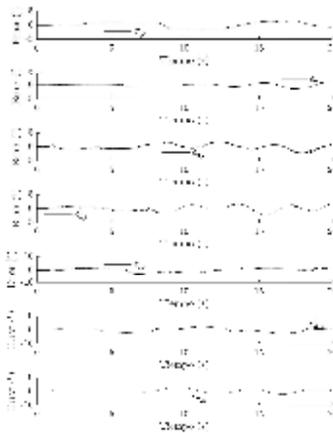


Figura 11. Errores de seguimiento eslabones 1-7

Los filtros de Kalman brindan una manera de estimar el vector de estado utilizando una ganancia de observador óptima para minimizar la media del error cuadrático (MSE por sus siglas en inglés) esperado de la señal estimada. Este hecho permite analizar y comparar el desempeño del algoritmo propuesto UKF con el algoritmo EKF reportado en (García-Hernández et al., 2011). El error de seguimiento $e_{i,k}^1$ es calculado como

$$MSE[e_{i,k}^1] = \sqrt{\frac{1}{t} \sum_{k=0}^n \|e_{i,k}^1\|^2 T} \quad (28)$$

donde $e_{i,k}^1 = x_{id,k}^1 - x_{i,k}^1$, $x_{i,k}^1$ es la salida del sistema, T es el periodo de muestreo y t es el tiempo de simulación. Los resultados se muestran en la Tabla 3.

6. CONCLUSIONES

Los resultados de simulación obtenidos son alentadores. Además, como se muestra en la Tabla 3, el entrenamiento UKF muestra un mejor rendimiento con respecto al entrenamiento EKF y es más fácil de implementar en

los manipuladores robóticos. Es importante destacar que el esquema de control propuesto no requiere conocer los parámetros de la planta ni las perturbaciones externas. Como trabajo futuro se plantea realizar los experimentos en tiempo real.

Tabla 3. MSE del error de seguimiento para el brazo robótico PA10-7CE

Eslabón	Algoritmo de entrenamiento	
	UKF	EKF
1	0.9239°	0.9787°
2	0.5246°	0.7290°
3	0.9143°	1.1950°
4	0.9327°	1.1690°
5	1.3090°	1.3590°
6	1.3730°	1.3590°
7	1.2130°	1.2630°

REFERENCIAS

- García-Hernández, R., Sanchez, E.N., Santibáñez, V., Llama, M.A., y Bayro-Corrochano, E. (2009). Real-time decentralized neural block controller for a robot manipulator. En *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 906–911.
- García-Hernández, R., Sanchez, E.N., Santibáñez, V., y Ruz-Hernandez, J. (2011). Decentralized neural block control for an industrial PA10-7CE robot arm. En *2011 International Joint Conference on Neural Networks*, 2787–2794.
- Guerra, J., Garcia-Hernandez, R., y Llama, M. (2019). Decentralized neural block control for a robot manipulator based in UKF training. En *2019 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, 88–94.
- Haykin, S. (2004). *Kalman filtering and neural networks*, volume 47. John Wiley and Sons.
- Julier, S.J. y Uhlmann, J.K. (1997). A new extension of the Kalman filter to nonlinear systems. *Signal processing, sensor fusion, and target recognition VI*, 3068(International Society for Optics and Photonics.), 182–194.
- Kalman, R.E. y Bucy, R.S. (1961). New results in linear filtering and prediction theory. *Journal of Basic Engineering*, 83(1), 95–108.
- Liu, M. (1999). Decentralized control of robot manipulators: nonlinear and adaptive approaches. *IEEE Transactions on Automatic Control*, 44(2), 357–363.
- Rhudy, M. y Gu, Y. (2013). Understanding nonlinear Kalman filters, part II: An implementation guide. *Interactive Robotics Letters*.
- Safaric, R. y Rodic, J. (2000). Decentralized neural-network sliding-mode robot controller. En *Industrial Electronics Society, 2000. IECON 2000. 26th Annual Conference of the IEEE*, volume 2, 906–911.
- Siljak, D.D. (2011). *Decentralized control of complex systems*. Courier Corporation.
- Utkin, V., Guldner, J., y Shi, J. (2009). *Sliding mode control in electro-mechanical systems*. CRC press.