

Experimental comparison of model-free nonlinear controllers on a 4-degrees-of-freedom SCARA robot ^{*}

Cesar De La Paz ^{*} Alejandro Gutierrez-Giles ^{*}
Israel Cruz-Vega ^{*}

^{} Instituto Nacional de Astrofísica, Óptica y Electrónica, Luis Enrique Erro 1, San Andrés Cholula, Puebla, México, (e-mails: {cesar.delapaz, alejandro.giles, icruzv}@inaoep.mx)*

Abstract: This paper presents the construction of an experimental platform consisting of a 4-degrees-of-freedom SCARA robot. The proposed platform is designed to quickly test nonlinear controllers experimentally. The code is written in C++, which is both flexible and efficient, resulting in a 2 ms sample time for the closed-loop controller. Additionally, a model-free adaptive controller is proposed in this work, which uses the direct kinematics of the robot to construct a pseudo-regressor. The proposed approach demonstrated superior performance when compared experimentally with two other model-free controllers and a classical model-based adaptive controller.

Keywords: Robotic manipulators, control applications, SCARA robot, nonlinear control, PID control, adaptive control, artificial neural network, model-free control.

1. INTRODUCCIÓN

En este trabajo se abordan técnicas de control para robots manipuladores, que son sistemas dinámicos no lineales, para lograr un alto desempeño en tareas de seguimiento de trayectorias y regulación. Normalmente, el alto desempeño en estas tareas se logra mediante técnicas de control no lineales basadas en modelo, como lo es el control adaptable clásico. Sin embargo, en este trabajo se consideran alternativas que no requieren conocer un modelo matemático de la dinámica del robot ni su correspondiente parametrización lineal. Para la validación y el estudio comparativo de las técnicas de control, se considera la construcción de un prototipo de robot SCARA, con el fin de obtener una comparación experimental del desempeño de varios controladores.

Una de las técnicas predominantes en muchos sistemas de control automático es el control PID, el cual es un sistema de control en lazo cerrado donde la salida $y(t)$ se retroalimenta al sistema, comparándola con la señal de referencia $r(t)$ para producir la señal de error $e(t)$, que se deriva e integra para generar la señal de control.

Este tipo de control debe su gran popularidad a la simplicidad de su implementación, dado que no requiere un modelo matemático de la planta a controlar y ha sido usado ampliamente en muchas aplicaciones de robótica (Yu and Rosen, 2010). Sin embargo, el control PID solo tiene buen desempeño en sistemas que se comportan de manera lineal o cerca de la zona lineal (Critchlow, 1985), por lo que el desempeño no será suficiente para

sistemas no lineales (Pambudi and Pelawi, 2015; Elsrogy et al., 2013), específicamente para requisitos de operación de alta velocidad y precisión al emplear robots de tipo SCARA (Al-Khedher and Alshamasin, 2012).

Dentro de los métodos de inteligencia computacional que pretenden obtener un buen desempeño en control de sistemas no lineales sin requerir un modelo matemático se encuentran el control difuso, el control por modos deslizantes, el control por rechazo activo de perturbaciones y el control basado en redes neuronales. Por ejemplo, el control difuso se utiliza en Huang and Yang (1997), donde las ganancias del controlador PID son ajustadas en línea de acuerdo al promedio ponderado del sistema de inferencia difuso, mejorando un esquema por ganancias fijas que requieren pre-ajuste y no son adaptables al comportamiento no lineal del robot SCARA. También en Otkarina et al. (2019), se muestran las mejoras sobre el control PID usando un control difuso, mejorando la suavidad de la transición y la estabilidad del robot.

Por otra parte, también se han empleado las Redes Neuronales Artificiales (RNA) para mejorar la robustez y la precisión para el control de movimiento de robots. Las RNA tratan de imitar la capacidad de aprendizaje y de adaptabilidad de un cerebro humano, al tener una gran cantidad de elementos de procesamiento (neuronas), interconexiones y capas internas de estos elementos neuronales. Por ejemplo, en Al-Khedher and Alshamasin (2012), una red neuronal entrenada estima la posición y el error de movimiento para activamente ajustar las ganancias de un controlador PID y, así, tratar con el comportamiento no lineal del robot SCARA. En Visioli and Legnani (2002), una RNA es empleada para calcular los coeficientes no-

^{*} El primer autor agradece a Conahcyt por la beca CVU: 1128328

lineales de la ecuación de Euler-Lagrange de un robot SCARA. En el trabajo mencionado, se concluye que es posible obtener un modelo basado en una red neuronal de propagación hacia adelante (feedforward), basado sólo en información de las entradas y las salidas del robot. Una de las desventajas de estos algoritmos computacionales, es la falta de estructuras matemáticas convencionales para análisis matemático e interpretabilidad. Para algunos sistemas, se requieren una cantidad considerable de datos de entrenamiento, como en el caso de las redes neuronales profundas, o se necesitan mejores algoritmos para evitar sobre-entrenamiento o bajo-entrenamiento (*overfitting* y *underfitting*), o no se pueden lograr implementaciones en hardware en tiempo real debido a la cantidad de recursos computacionales necesarios (Camilleri and Prescott, 2017).

En los últimos años también se han desarrollado diversos algoritmos de control por modos deslizantes, que no requieren conocer un modelo matemático y han mostrado un buen desempeño en comparación con los algoritmos lineales. Sin embargo, existen otros problemas asociados con este tipo de controladores, como el chattering y la necesidad de una alta frecuencia de muestreo para obtener un buen desempeño.

Asimismo, en años recientes, se han utilizado otras técnicas, como el control por rechazo activo de perturbaciones (ADRC, por sus siglas en inglés). Algunos de los controladores por rechazo activo de perturbaciones sólo requieren conocer una parte del modelo, mientras que otras propuestas no requieren conocer un modelo matemático.

En este trabajo se realizará una comparación del desempeño de algunos algoritmos de control sin modelo propuestos en la literatura: el control PID y el control adaptable basado en RNA, junto con un nuevo controlador adaptable, basado en cinemática directa, que se propone en este estudio. Para realizar la comparación, también se presenta la construcción e implementación de un robot manipulador de cuatro grados de libertad del tipo SCARA. El resto del documento se organiza de la siguiente forma: en la Sección 2 se presentan algunos detalles sobre el diseño y puesta en funcionamiento del robot SCARA utilizado en la comparación experimental; en la Sección 3 se describen los controladores a comparar y el controlador propuesto; en la Sección 4 se presentan los resultados experimentales; y, por último, en la Sección 5 se dan algunas conclusiones y direcciones para trabajos futuros.

2. DESCRIPCIÓN DEL SISTEMA

2.1 Características del robot manipulador SCARA

En este trabajo se considera la implementación de un robot manipulador SCARA (siglas del inglés: *Selective Compliant Articulated Robot for Assembly*), visualizado en la Figura 1.

El robot SCARA se caracteriza por poseer cuatro grados de libertad (tres rotacionales y uno prismático), de acuerdo con el número de movimientos independientes (Cornejo

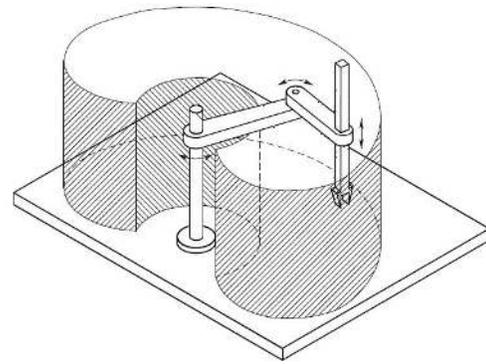


Figura 1. Robot manipulador SCARA y su espacio de trabajo (Siciliano et al., 2010).

et al., 2022). Para llevar a cabo experimentos con controladores no lineales, se diseñó una plataforma física como banco de pruebas. Las piezas para formar los eslabones del robot se imprimieron con filamento a base de ácido poliláctico (PLA, por sus siglas en inglés).

Como actuadores para las articulaciones se consideraron motores de corriente directa con cajas de engranaje reductor y encoders de cuadratura para medir la posición de las articulaciones. La comunicación con la computadora personal (PC) en la que se implementan los algoritmos de control se realizó mediante un esquema de microcontroladores basado en la propuesta de Flores and Gutierrez-Giles (2022). En este trabajo se presenta una mejora del sistema al implementarlo en un robot de 4 grados de libertad, en lugar de los 2 del trabajo mencionado. El diseño es de bajo costo y consta de la conexión de microcontroladores *PIC* de *Microchip* para la adquisición y transmisión de datos con la PC vía puerto USB. El diagrama esquemático de las conexiones del sistema se muestra en la Figura 2.

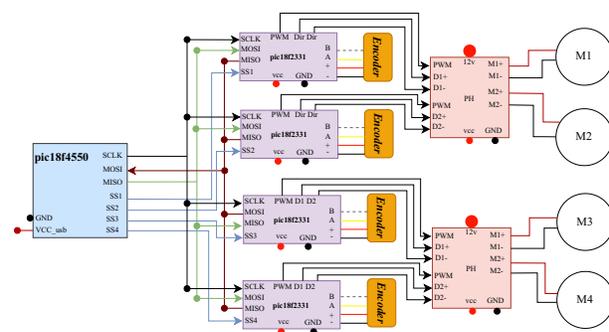


Figura 2. Esquema electrónico general para el robot manipulador SCARA de cuatro grados de libertad.

Debido a que la comunicación entre los microcontroladores se realizó mediante el protocolo *SPI* y se utilizaron los *Multimedia Timers* de *Microsoft Windows*, el periodo de muestreo obtenido para todo el lazo de control fue de 2 ms, lo que es apropiado para la implementación de la mayoría de los controladores no lineales existentes en la literatura. El robot SCARA de cuatro grados de libertad construido en este trabajo puede ser visualizado en la Figura 3.



Figura 3. Robot manipulador SCARA de cuatro grados de libertad.

2.2 Modelo dinámico del manipulador

El modelo dinámico de un robot se puede derivar utilizando principios físicos bien establecidos, como las leyes de la mecánica Newtoniana y Lagrangiana. Este enfoque implica la formulación de las ecuaciones de movimiento dinámico que describen el comportamiento de las distintas articulaciones del manipulador en función de los parámetros geométricos e inerciales de sus componentes, incluyendo las fuerzas externas que interactúan con éste Cortés (2020). Si se considera un robot manipulador de n grados de libertad, formado por eslabones rígidos conectados por articulaciones libres de elasticidad, se puede obtener el modelo Lagrangiano

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

donde $q \in \mathbb{R}^n$ son las coordenadas generalizadas, $M(q) \in \mathbb{R}^{n \times n}$ es la matriz de inercia, $C(q, \dot{q}) \in \mathbb{R}^n$ es el vector de fuerzas de Coriolis y fuerzas centrípetas, $f(\dot{q}) \in \mathbb{R}^n$ es el vector de fuerzas debidas a la fricción, $g(q) \in \mathbb{R}^n$ es el vector de fuerzas debidas a la de gravedad, y τ son los pares de entrada articulares.

2.3 Parámetros dinámicos

Una de las ventajas de construir un robot manipulador desde sus piezas básicas es que se pueden medir directamente algunos de los parámetros físicos que componen el modelo dinámico de la ecuación (1). Los parámetros medidos para el prototipo construido se muestran en la Tabla 1.

3. CONTROL NO LINEAL SIN MODELO

En esta sección se fundamentan los principios de los controladores mencionados en este trabajo, con el objetivo de realizar su comparación.

3.1 Control PID

Uno de los controladores más utilizados en la práctica (Åström and Hägglund, 2006) es el controlador

Tabla 1. Notación de los parámetros del robot manipulador.

Masas	Momentos de inercia
$m_1 = 0.93 \text{ kg}$	$p_r = 0.06$
$m_2 = 0.68 \text{ kg}$	$I_{z2} = m_2 \frac{(a_2 * a_2 + p_r^2)}{12}$
$m_3 = 0.54 \text{ kg}$	$I_{z3} = m_3 \frac{(a_3 * a_3 + p_r^2)}{12}$
$m_4 = 0.35 \text{ kg}$	$I_{z4} = \frac{m_4 p_r^2}{6}$
Parámetros Denavit - Hartenberg	Centros de masa
$a_2 = 0.2m$	$l_{c2} = \frac{a_2}{2}$
$a_3 = 0.145m$	$l_{c3} = \frac{a_3}{2}$
$a_4 = 0.05m$	
Definiciones de senos y cosenos	Constante gravitacional
$c_i = \cos(q_i)$	$g_0 = 9.8 \frac{m}{s^2}$
$s_i = \sin(q_i)$	
Coeficientes de fricción viscosa	
$cf_1, cf_2, cf_3, cf_4 = 0.01$	

Proporcional-Integral-Derivativo (PID), debido a su simplicidad y facilidad de implementación. Parte de su simplicidad radica en que no necesita un modelo matemático para utilizarse. Una ley de control PID para robots manipuladores está dada por

$$\tau = -k_p e - k_i \int e dt - k_d \frac{d(e)}{dt}, \quad (2)$$

donde $e = q - q_d$ es el error de posición, q_d son las posiciones deseadas en coordenadas articulares y q son las posiciones reales. Así mismo, k_p , k_i y k_d son las ganancias proporcional, integral y derivativa, respectivamente.

3.2 Control adaptable basado en RNA

En este trabajo se considera el algoritmo de control adaptable basado en RNA propuesto en He et al. (2015). Se eligió este algoritmo dado que no utiliza un modelo matemático del robot para ser implementado. Para este algoritmo se considera la estructura neuronal que permite la estimación de los efectos no lineales, que se muestra en la Figura 4.

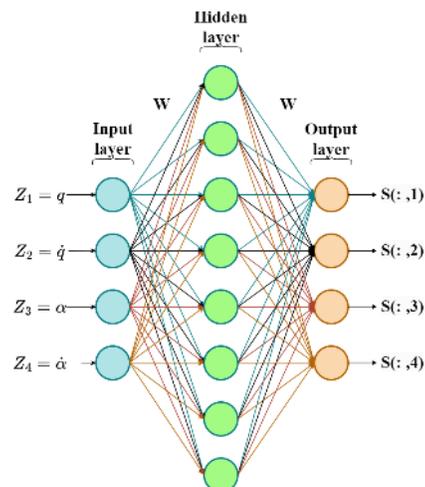


Figura 4. RNA que estima las no linealidades.

Esta red se caracteriza por tener cuatro neuronas externas que definen las entradas, correspondientes a los términos de variables de movimiento reales y estimadas. En la parte

interna de la red se concentran las funciones de activación Gaussianas, con 8 neuronas ocultas.

Adicionalmente, en el artículo original se utiliza la red neuronal para compensar la zona muerta en los actuadores. La idea de utilizar este tipo de técnica es poder aproximar las no linealidades del sistema a partir de estimaciones del movimiento del robot.

El esquema consta de un vector de pesos \mathbf{W} y de las variables del algoritmo clásico de control adaptable (Slotine and Li, 1987) definidas como

$$\alpha_1 = \dot{q}_d - \mathbf{k}_1(\mathbf{q} - \mathbf{q}_d) \quad (3)$$

$$\dot{\alpha}_1 = \ddot{q}_d - \mathbf{k}_1(\dot{\mathbf{q}} - \dot{q}_d) \quad (4)$$

$$\mathbf{e}_2 = \dot{\mathbf{q}} - \alpha_1 = \dot{\mathbf{q}} - \dot{q}_d + \mathbf{k}_1(\mathbf{q} - \mathbf{q}_d), \quad (5)$$

donde \mathbf{k}_1 es una matriz de ganancias constantes. α y $\dot{\alpha}$ son las velocidades y aceleraciones estimadas, respectivamente.

Por otra parte, de acuerdo con la Figura 4, se obtiene la capa de entrada Z . Por ejemplo, para el caso particular de un robot de 4 grados de libertad, se tiene

$$Z = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \\ \dot{q}_1 & \dot{q}_2 & \dot{q}_3 & \dot{q}_4 \\ \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \dot{\alpha}_{11} & \dot{\alpha}_{12} & \dot{\alpha}_{13} & \dot{\alpha}_{14} \end{bmatrix}, \quad (6)$$

donde α_{ij} denota la j -ésima componente del vector α_i . De forma similar, para calcular el vector de entrada Z_τ se tiene

$$Z_\tau = \begin{bmatrix} \int \dot{\tau}_1 dt & \int \dot{\tau}_2 dt & \int \dot{\tau}_3 dt & \int \dot{\tau}_4 dt \\ q_1 & q_2 & q_3 & q_4 \\ \dot{q}_1 & \dot{q}_2 & \dot{q}_3 & \dot{q}_4 \\ \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \end{bmatrix}. \quad (7)$$

La función de base radial de la red neuronal se presenta en la ecuación

$$S_i(Z_i) = \exp\left[-\frac{(Z_i - \mu_i)^T(Z_i - \mu_i)}{\eta_i^2}\right], \quad (8)$$

para $i = 1, \dots, n$. Del mismo modo, para la salida de la red neuronal utilizada para la estimación de la zona muerta se obtiene

$$S_{\tau n}(Z_{\tau n}) = \exp\left[-\frac{(Z_{\tau n} - \mu_{\tau n})^T(Z_{\tau n} - \mu_{\tau n})}{\eta_n^2}\right]. \quad (9)$$

Por último, la ley de control se calcula mediante

$$\boldsymbol{\tau} = \mathbf{k}_2 \mathbf{e}_2 + (\mathbf{W}^T \mathbf{S} + \mathbf{W}_\tau^T \mathbf{S}_\tau)^T, \quad (10)$$

donde \mathbf{k}_2 es una matriz de ganancias, \mathbf{S} y \mathbf{S}_τ son las salidas de la red neuronal, para estimar las no linealidades del sistema.

La actualización de los pesos de la red neuronal \mathbf{W} se obtiene mediante

$$\dot{\mathbf{W}} = -\gamma(\mathbf{S} \mathbf{e}_2 + \boldsymbol{\sigma} \mathbf{W}). \quad (11)$$

De la misma forma, para \mathbf{W}_τ

$$\dot{\mathbf{W}}_\tau = -\gamma_\tau(\mathbf{S}_\tau \mathbf{e}_2 + \boldsymbol{\sigma}_\tau \mathbf{W}_\tau). \quad (12)$$

donde γ es el vector de ganancias de adaptación y σ es la ganancia para la actualización de los pesos de la red neuronal artificial.

3.3 Control adaptable basado en cinemática directa

En esta sección se describe el controlador que se propone en este trabajo. El controlador propuesto no necesita conocer el modelo dinámico del robot para ser implementado, solo la cinemática directa que puede ser obtenida mediante el algoritmo de Denavit-Hartenberg (Arteaga et al., 2022, p. 43). El algoritmo consta de un regresor \mathbf{Y} y un vector de parámetros estimados $\hat{\boldsymbol{\theta}}$ y utiliza las variables de Slotine-Li, las mismas que se utilizan el control adaptable clásico y el control adaptable mediante redes neuronales descrito en la Sección 3.2, i.e. α_1 y $\dot{\alpha}_1$ definidas en (3) y (4), respectivamente. El regresor propuesto se construye de la siguiente forma

$$\mathbf{Y}(\alpha_1, \dot{\alpha}_1, \mathbf{q}) = [\boldsymbol{\beta}_1^T \otimes \mathbf{I}_{n \times n} \boldsymbol{\beta}_2^T \otimes \mathbf{I}_{n \times n} \mathbf{Y}_g(\mathbf{q})], \quad (13)$$

donde \otimes denota el producto de Kronecker, $\mathbf{I}_{n \times n}$ es la matriz identidad de $n \times n$, $\boldsymbol{\beta}_1$ y $\boldsymbol{\beta}_2$ son vectores cuyas i -ésimas componentes están definidas por $\beta_{1i} = \tanh(\alpha_{1i})$ y $\beta_{2i} = \tanh(\dot{\alpha}_{1i})$, respectivamente, y la matriz $\mathbf{Y}_g(\mathbf{q})$ se construye como se indica en el siguiente algoritmo:

1. Se considera que se dispone de los ejes articulares ${}^0\mathbf{z}_0, \dots, {}^0\mathbf{z}_n$ y los orígenes de los sistemas coordenados ${}^0\mathbf{o}_0, \dots, {}^0\mathbf{o}_n$ de la cinemática directa.
2. Se define un vector unitario $\bar{\mathbf{g}}$, cuya dirección es la de crecimiento de la energía potencial (contraria a la aceleración de la gravedad) en coordenadas de la base del robot.
3. Para calcular la i -ésima columna de \mathbf{Y}_g :
 - a) Si la articulación i es prismática, crear un vector de dimensión n cuyo i -ésimo elemento se calcula como $\bar{\mathbf{g}}^T {}^0\mathbf{z}_{i-1}$ y todos los demás elementos son cero.
 - b) Si la articulación i es de revolución, crear un vector de dimensión n cuyo j -ésimo componente se calcula mediante:
 - 1) Si $j \leq i$, se calcula $(\bar{\mathbf{g}} \times {}^0\mathbf{z}_{j-1})^T ({}^0\mathbf{o}_i - {}^0\mathbf{o}_{j-1})$.
 - 2) Si $j > i$, el j -ésimo componente es cero.

Comentario 1. Este algoritmo se simplifica de manera significativa dadas las propiedades del producto punto y el producto cruz de los vectores $\bar{\mathbf{g}}$ y ${}^0\mathbf{z}_{i-1}$. Por ejemplo, para articulaciones prismáticas, si el eje de la articulación siempre es perpendicular a la dirección de la gravedad, la columna correspondiente es cero. Si el eje de la articulación i tiene la dirección de la gravedad o contraria a ésta, se tendrá un vector con la i -ésima componente igual a ± 1 . Para articulaciones de revolución, si el eje de giro de la articulación siempre tiene la misma dirección que la gravedad (o totalmente opuesta), entonces la columna es cero. Aunque en general estas simplificaciones no se cumplen y se debe seguir el algoritmo anterior, para

muchos robots manipuladores existentes sí se cumplen en una o varias articulaciones.

Comentario 2. En lugar de hacer unitario al vector \bar{g} , se puede hacer que éste tenga norma igual a la aceleración de la gravedad g_0 para mejorar el desempeño del controlador, manteniendo la misma dirección.

El funcionamiento del esquema propuesto, se basa en el cambio de variable

$$s = \dot{q} - \alpha_1 \quad (14)$$

con el que se obtiene el modelo equivalente a (1), cuando se considera sólo fricción viscosa

$$M(q)\dot{s} + C(q, \dot{q})s + Ds = \tau - M(q)\dot{\alpha}_1 - C(q, \dot{q})\alpha_1 - D\alpha_1 - g(q), \quad (15)$$

donde $D \in \mathbb{R}^{n \times n}$ es una matriz diagonal de coeficientes de fricción viscosa. La suposición en la que se basa el funcionamiento del esquema es que exista un vector $\theta(t)$ tal que la siguiente aproximación sea válida (omitendo los argumentos por simplicidad)

$$M\dot{\alpha}_1 + C\alpha_1 + D\alpha_1 + g = Y\theta(t) + r(t) \quad (16)$$

al menos localmente, donde $r(t)$ es un término residual que se asume acotado. Si esta aproximación es válida, se puede proponer la ley de control

$$\tau = -k_2s + Y(\alpha_1, \dot{\alpha}_1, q)\hat{\theta} \quad (17)$$

y una ley de adaptación

$$\dot{\hat{\theta}} = -\Gamma Y^T s. \quad (18)$$

Se puede demostrar que el controlador propuesto es estable localmente, bajo las suposiciones mencionadas, utilizando la función de Lyapunov $V = s^T M(q)s$.

4. EXPERIMENTOS

Se realizaron experimentos para el seguimiento de trayectorias senoidales, utilizando las técnicas de control PID, control adaptable clásico, control adaptable basado en cinemática directa y control adaptable basado en RNA.

$$\begin{bmatrix} q_{d1} = 0.08 \sin(0.05t) \\ q_{d2} = 25 \sin(0.1t) \\ q_{d3} = 30 \sin(0.5t) \\ q_{d4} = 35 \sin(0.7t) \end{bmatrix} \quad (19)$$

Se emplearon las mismas ganancias proporcionales y derivativas para todos los controladores. Las ganancias propuestas para el control PID fueron: $k_p = \text{diag}(20, 10, 10, 10)$, $k_d = \text{diag}(0.8, 2.0, 0.9, 1)$ y $k_i = \text{diag}(2.2, 1.3, 1.6, 1.3)$, mismas que se emplearon en los controladores restantes: $k_2 = k_d$ y $k_1 = \lambda = \frac{k_p}{k_d} = \text{diag}(60, 5, 11.11, 100)$. Así mismo, para los controles adaptables: clásico, basado en RNA y basado en cinemática directa, se emplearon las ganancias $\gamma = \text{diag}(0.002, 0.00001, 0.00001, 0.00001, 0.00001,$

$0.00001, 0.00001, 0.1, 0.1, 0.1)$, $\gamma = 0.55$, $\gamma = \text{diag}(30, 30, 0.01, 10, 0.001, 30, 0.1, 0.1, 0.01, 0.1, 30, 0.1, 0.01, 10, 0.1, 1.0, 0.001, 0.1, 0.01, 0.01, 0.01, 30, 0.1, 0.1, 0.01, 1.0, 20, 0.1, 0.01, 1, 0.1, 1, 0.0005)$, respectivamente.

Los experimentos se ejecutaron durante 15 y 20 segundos, obteniendo los resultados que se muestran en las Figuras 5-8, se especifica que se efectuó un ajuste para optimizar la visualización.

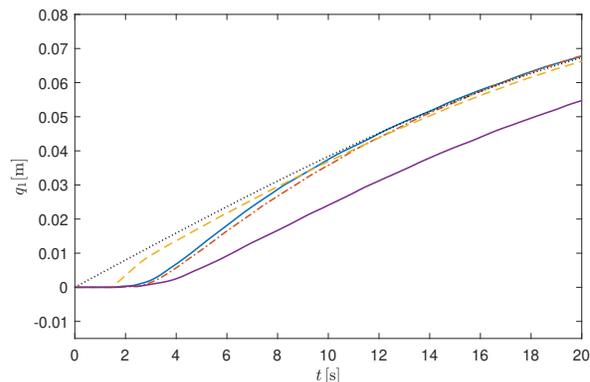


Figura 5. Posición q_1 : PID(-); adaptable basado en RNA(-); adaptable(-.-); adaptable basado en cinemática directa(-.-.-); referencia(...).

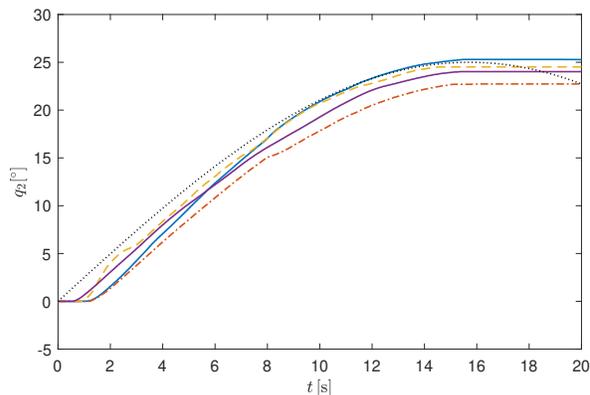


Figura 6. Posición q_2 : PID(-); adaptable basado en RNA(-); adaptable(-.-); adaptable basado en cinemática directa(-.-.-); referencia(...).

A partir de los resultados experimentales obtenidos para cada uno de los algoritmos, se determinó el error cuadrático medio (RMSE, por sus siglas en inglés: *Root Mean Square Error*) para obtener un análisis numérico del rendimiento de los controladores, visualizado en la Tabla 2.

Tabla 2. Comparación de los errores cuadráticos medios.

Root Mean Square Error (RMSE)				
	PID	Adaptable clásico	Adaptable basado en RNA	Adaptable basado en cinemática directa
q_1	0.0035	0.0043	0.0105	0.0011
q_2	1.9762	2.3497	2.6458	1.9146
q_3	3.6720	1.4676	3.5603	0.9633
q_4	3.1909	0.2495	2.9967	0.4007

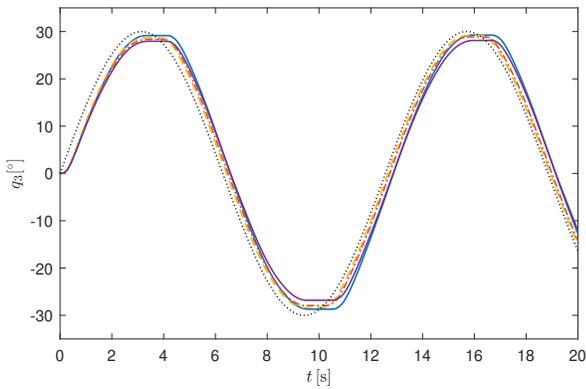


Figura 7. Posición q_3 : PID(—); adaptable basado en RNA(---); adaptable(.-.); adaptable basado en cinemática directa(- - -); referencia(...).

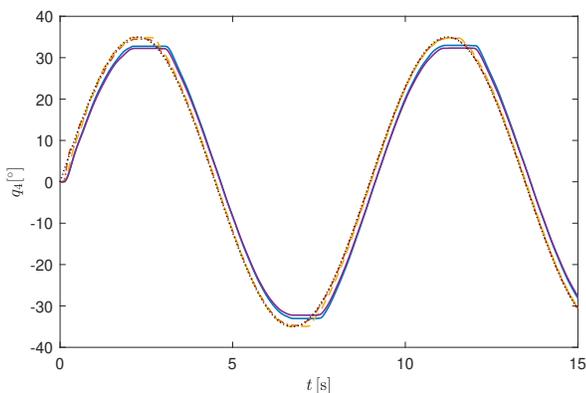


Figura 8. Posición q_4 : PID(—); adaptable basado en RNA(---); adaptable(.-.); adaptable basado en cinemática directa(- - -); referencia(...).

5. CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se presentó la construcción de un robot de bajo costo que permite la evaluación de algoritmos de control no lineales de manera flexible. Asimismo, se propuso un control adaptable sin utilizar el modelo dinámico del robot ni la correspondiente parametrización lineal, únicamente basado en la cinemática directa, que normalmente es más fácil de obtener. Dicho controlador tuvo un mejor desempeño que otros controladores libres de modelo, e incluso que el control adaptable clásico basado en modelo.

Como trabajo futuro, se realizará la demostración de estabilidad para definir la región en la que es válida y cómo deben de seleccionarse las ganancias del controlador. También se trabajará en la compensación de otros efectos dinámicos no considerados, como la fricción seca o la zona muerta.

REFERENCIAS

Al-Khedher, M.A. and Alshamasin, M.S. (2012). SCARA robot control using neural networks. In *2012 4th International Conference on Intelligent and Advanced Systems (ICIAS2012)*, volume 1, 126–130. IEEE.

Arteaga, M.A., Gutiérrez-Giles, A., and Pliego-Jiménez, J. (2022). *Local stability and ultimate boundedness in the control of robot manipulators*. Springer.

Åström, K.J. and Hägglund, T. (2006). PID control. *IEEE Control Systems Magazine*, 1066.

Camilleri, D. and Prescott, T. (2017). Analysing the limitations of deep learning for developmental robotics. In *Biomimetic and Biohybrid Systems: 6th International Conference, Living Machines 2017, Stanford, CA, USA, July 26–28, 2017, Proceedings 6*, 86–94. Springer.

Cornejo, J., Cruz, V., Carrillo, F., Cerda, R., and Penadillo, E.R.S. (2022). Mechatronics design and kinematic simulation of SCARA robot to improve safety and time processing of Covid-19 rapid test. In *2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT)*, 1–6. IEEE.

Cortés, F.R. (2020). *Robótica: control de robots manipuladores*. Marcombo.

Critchlow, A.J. (1985). Introduction to robotics.

Elsrogy, W.M., Fkirin, M., and Hassan, M.M. (2013). Speed control of DC motor using PID controller based on artificial intelligence techniques. In *2013 International Conference on Control, Decision and Information Technologies (CoDIT)*, 196–201. IEEE.

Flores, J.C. and Gutiérrez-Giles, A. (2022). Design and construction of a low-cost robotic system for teaching and research. In *Memorias del 2022 Congreso Nacional de Control Automático*. Asociación de México de Control Automático (AMCA).

He, W., David, A.O., Yin, Z., and Sun, C. (2015). Neural network control of a robotic manipulator with input deadzone and output constraint. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(6), 759–770.

Huang, S.J. and Yang, D.N. (1997). Fuzzy logic controller for a SCARA robot with synchronous network. *International Journal of Computer Applications in Technology*, 10(1-2), 15–26.

Oktarina, Y., Septiarini, F., Dewi, T., Risma, P., and Nawawi, M. (2019). Fuzzy-PID controller design of 4 DoF industrial arm robot manipulator. *Computer Engineering and Applications Journal*, 8(2), 123–136.

Pambudi, W.S. and Pelawi, J.P.B. (2015). Simulasi folding machine dengan PID, P, PI, PD and fuzzy PD. *Jurnal Sains dan Informatika*, 1(1), 25–33.

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2010). *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. Springer London.

Slotine, J.J.E. and Li, W. (1987). On the adaptive control of robot manipulators. *The international journal of robotics research*, 6(3), 49–59.

Visioli, A. and Legnani, G. (2002). On the trajectory tracking control of industrial SCARA robot manipulators. *IEEE transactions on industrial electronics*, 49(1), 224–232.

Yu, W. and Rosen, J. (2010). A novel linear PID controller for an upper limb exoskeleton. In *49th IEEE Conference on Decision and Control (CDC)*, 3548–3553. IEEE.