

Adaptive Neural Compensator for Robotic Systems Control

D. Gandolfo, F. Rossomando, C. Soria, and R. Carelli

Abstract—In the area of robotics systems, there are numerous applications where robots are expected to move rapidly from one place to another, or follow desired trajectories while maintaining good dynamic behavior. However, certain non-linearities, uncertainties in dynamics and external perturbations make the design of ideal controllers a complicated task in many situations. In this paper, we propose a control scheme that combines a nominal feedback controller with a classical PD and a robust adaptive compensator based on artificial neural networks. Using this control scheme, it is possible to obtain a fully tuned compensation parameters and a strong robustness with respect to uncertain dynamics and different non-linearities, as well as to keep the output tracking error bounded to values close to zero. In order to show the performance of the proposed technique, a SCARA (Selective Compliant Articulated Robot Arm) type robot with two degrees of freedom is considered in this case; but this control proposal can be applied to different systems with dynamic variations. The efficiency and performance of the control law is demonstrated through simulation results and the stability analysis is carried out using Lyapunov's theory.

Index Terms—Adaptive control, Artificial neural network, Identification, Robot manipulator.

I. INTRODUCCIÓN

LOS manipuladores robóticos industriales son sistemas dinámicos altamente no lineales con dinámicas e incertidumbres no modeladas, los cuales se utilizan comúnmente en tareas tales como la pulverización de pintura, la entrega de materiales en la fabricación y los sistemas de posicionamiento de piezas de precisión [1]. Para estas aplicaciones, se espera que se muevan rápidamente de un lugar a otro, o que sigan las trayectorias deseadas mientras mantienen un buen comportamiento dinámico. Sin embargo, las incertidumbres en la dinámica de los manipuladores robóticos hacen que el diseño de controladores ideales (para este tipo de sistemas) sea una tarea complicada en el área de control. Por lo tanto, es necesario tenerlas en cuenta en el diseño de control para un mejor desempeño del robot en aplicaciones de precisión, como puede ser el caso de robots quirúrgicos. En la literatura científica existen varias referencias a este tipo de problemática entre ellas las basadas en inteligencia artificial ([2]-[6]).

D. Gandolfo, Instituto de Automática – Conicet - Universidad Nacional de San Juan, Argentina, dgandolfo@inaut.unsj.edu.ar.

F. Rossomando, Instituto de Automática – Conicet - Universidad Nacional de San Juan, Argentina, frosoma@inaut.unsj.edu.ar.

C. Soria, Instituto de Automática – Conicet - Universidad Nacional de San Juan, Argentina, csoria@inaut.unsj.edu.ar.

R. Carelli, Instituto de Automática – Conicet - Universidad Nacional de San Juan, Argentina, rcarelli@inaut.unsj.edu.ar.

En libros Kelly et al. [7] y Lewis et al. [8] se proporciona una buena revisión de técnicas de control no lineal para su aplicación en manipuladores robóticos, las cuales incluyen un detallado análisis de estabilidad.

Trabajos recientes se basan en esquemas de control usando redes neuronales artificiales (ANN), como el control adaptable basado en redes neuronales sobre modelos discretos [9], [10] y el control por superficie deslizante neuro adaptable [11],[12], así como también control neuro adaptable con estimación de parámetros [13]. En los casos mencionados la técnica de control es aplicada sobre manipuladores robóticos.

Considerando técnicas de control basadas en Inteligencia Artificial (IA), se puede mencionar el trabajo [14], en el cual se plantea el control de seguimiento de trayectoria de un péndulo de rueda de inercia considerando las incertidumbres dinámicas; el control propuesto es un control adaptable basado en regresores y también propone un control basado en redes neuronales. En el desarrollo de este último controlador se utilizó un perceptrón multicapa (MLP), donde los autores demuestran la convergencia asintótica del error de seguimiento. El trabajo presenta experimentos en tiempo real donde se compara su funcionamiento con un PID clásico.

Otra aplicación usando MLP es mostrada en los trabajos [15] y [16], donde se presenta un controlador adaptable para un péndulo de Furuta, asimismo estos trabajos demuestran que la estabilidad de los controladores propuestos es finalmente uniformemente acotada. En ambos casos, resultados experimentales muestran un buen desempeño.

Entre las distintas aplicaciones de las técnicas de control sobre robots tipo SCARA podemos mencionar el trabajo [3], donde los autores utilizan un sistema adaptable de inferencia neuro-difuso (ANFIS). El cual se encuentra diseñado para el control dinámico del manipulador robótico de tres grados de libertad (D.O.F). Este sistema neuro-difuso permite una adaptación del controlador propuesto a través del ajuste de sus parámetros en función de las variaciones en el comportamiento del sistema. El trabajo presenta resultados de simulación con ANFIS de Matlab® que demuestran su factibilidad de aplicación en este tipo de robots manipuladores.

Basado en el modelo de robot SCARA, el trabajo propuesto por [17] muestra un controlador adaptable cuyo diseño utiliza redes neuronales de funciones de base radial (RBF), donde los únicos parámetros de ajuste son los pesos de la capa de salida de la red neuronal propuesta.

Otra técnica de control para este tipo de robots es propuesta por [4] donde se implementa un controlador difuso adaptable para manipuladores robóticos. El controlador difuso adaptable propuesto en el trabajo citado, utiliza una formulación de juego

de Markov basado en un sistema de inferencia difusa (FIS), el cual genera un control rápido y posee la ventaja que no requiere conocer la dinámica exacta del sistema robótico.

En [18], se propone un controlador computacional que consiste en un controlador basado en el Jacobiano del modelo del robot más un término adaptable que estima las fuerzas de gravedad generalizadas. La teoría de estabilidad de Lyapunov es utilizada por los autores para obtener la ley de control, donde las condiciones de ganancias aseguran la estabilidad asintótica de la técnica propuesta.

En este documento, proponemos un enfoque diferente para reducir el error de seguimiento. El controlador PD se encarga de la dinámica conocida del robot y la ANN compensa o reduce las posibles variaciones dinámicas. En forma similar que [19] donde este trabajo no presenta un análisis de estabilidad.

La propuesta original de este trabajo es el desarrollo de un de un compensador adaptable basado en redes neuronales del tipo RBF para la compensación del error producido por las variaciones dinámicas del robot manipulador. Así, la red neuronal ajusta sus parámetros para compensar la diferencia dinámica existente entre el controlador clásico (en este caso controlador PD) y el robot, y de este modo reduce el error de trayectoria. Además, la compensación propuesta ajusta en línea no solo los pesos de salida, sino también los centros y la apertura de la función gaussiana, las cuales son deducidas del análisis de estabilidad de la técnica de control propuesta. En la mayoría de los trabajos abordados en la literatura, el ajuste de estos dos últimos parámetros se realiza fuera de línea o en línea usando el algoritmo k-means de agrupamiento.

Cabe aclarar que una ventaja interesante de este enfoque de control es la flexibilidad de poder ser utilizado en paralelo con un cualquier controlador existente (no solo un PD) para disminuir el error de seguimiento, así como también puede utilizarse para otros sistemas robóticos.

El documento está organizado de la siguiente manera: la Sección II presenta una descripción general del sistema y muestra la representación matemática del modelo de robot manipulador tipo SCARA completo junto con el controlador PD. La compensación neuronal dinámica se trata en la Sección III, así como en el análisis de error correspondiente. La sección IV presenta resultados mediante simulaciones computacionales para mostrar el rendimiento del controlador adaptable. Finalmente, las conclusiones se dan en la Sección V.

II. FORMULACIÓN DEL PROBLEMA

A. Sistema Robótico y Control PD

La ecuación dinámica en el espacio de articulación para un manipulador de robot con n grados de libertad se puede escribir como en [7] y [8]:

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

donde:

\mathbf{q} : es el vector ($n \times 1$) de coordenadas generalizadas (posiciones articulares),

$\mathbf{M}(\mathbf{q})$: matriz ($n \times n$) generalmente denominada matriz de masa del manipulador que contiene las funciones de energía cinética.

$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$: matriz ($n \times n$) que representa los torques que surgen de las fuerzas centrífugas y Coriolis.

$\mathbf{f}(\dot{\mathbf{q}})$: ($n \times 1$) representa los efectos de fricción de Coulomb cuando el manipulador se está moviendo en su espacio de trabajo.

\mathbf{T} : Vector de torques de acción de control de articulaciones ($n \times 1$)

Las matrices de parámetros estimados del robot Bosch SR-800 son las siguientes:

$$\hat{\mathbf{M}}(\mathbf{q}) = \begin{bmatrix} 1.7277 + 0.1908\cos(q_1) & 0.0918 + 0.0954\cos(q_1) \\ 0.3340 + 0.3418\cos(q_1) & 0.9184 \end{bmatrix}$$

$$\hat{\mathbf{c}}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} 31.8192 - 0.0954\sin(q_1)(\dot{q}_1) & -0.0954\sin(q_1)(\dot{q}_0 + \dot{q}_1) \\ 0.3418\sin(q_1)(\dot{q}_0) & 12.5783 \end{bmatrix}$$

$$\hat{\mathbf{f}}(\dot{\mathbf{q}}) = \begin{bmatrix} 1.0256\text{sign}(\dot{q}_0) \\ 1.7842\text{sign}(\dot{q}_1) \end{bmatrix}$$

La matriz $\hat{\mathbf{M}}(\mathbf{q})$ será utilizada en la Sección III para el desarrollo de la técnica de control, es fácil observar que aplicando el criterio de Sylvester es definida positiva, por lo cual posee inversa. En el libro [7] indica las propiedades de la matriz de masas del robot SCARA en la que demuestra que las mencionadas matrices son definidas positivas.

La ecuación (1) puede ser representada por:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{T} \quad (2)$$

donde $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{f}(\dot{\mathbf{q}})$ y definiendo \mathbf{q}_d como el vector de posición articular deseado, entonces el error de seguimiento se define:

$$\mathbf{e} = \mathbf{q}_d - \mathbf{q} \quad (3)$$

Derivando (3) resulta:

$$\begin{cases} \dot{\mathbf{e}} = \dot{\mathbf{q}}_d - \dot{\mathbf{q}} \\ \ddot{\mathbf{e}} = \ddot{\mathbf{q}}_d - \ddot{\mathbf{q}} \end{cases} \quad (4)$$

Resolviendo ahora para $\ddot{\mathbf{q}}$ en (2) y sustituyendo en (4), se puede obtener la siguiente ecuación:

$$\ddot{\mathbf{e}} = \ddot{\mathbf{q}}_d + \mathbf{M}^{-1}(\mathbf{T} - \mathbf{N}) \quad (5)$$

De la ecuación anterior se puede definir una acción de control simple:

$$\mathbf{u} = \ddot{\mathbf{q}}_d + \mathbf{M}^{-1}(\mathbf{T} - \mathbf{N}) \quad (6)$$

La transformación de la linealización por realimentación (6) puede invertirse para producir:

$$\mathbf{T} = \mathbf{M}(\ddot{\mathbf{q}}_d - \mathbf{u}) + \mathbf{N} \quad (7)$$

Para obtener un control PD, la señal \mathbf{u} debe definirse como:

$$\mathbf{u} = -\mathbf{K}_v \dot{\mathbf{e}} - \mathbf{K}_p \mathbf{e} \quad (8)$$

donde \mathbf{K}_v y \mathbf{K}_p son matrices de orden $n \times n$ de ganancia diagonal constantes, (2) se conoce como el método de control manipulador de torque calculado y la Fig. 1 muestra el diagrama de bloques de este esquema de control. Se supone que cualquier trayectoria deseada del manipulador se conoce y se puede expresar como funciones de tiempo de posiciones, velocidades y aceleraciones conjuntas, \mathbf{q}_d , $\dot{\mathbf{q}}_d$ y $\ddot{\mathbf{q}}_d$.

Luego, la acción general de control del brazo del robot queda definida por:

$$\mathbf{T} = \mathbf{M}(\ddot{\mathbf{q}}_d + \mathbf{K}_v \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e}) + \mathbf{N} \quad (9)$$

La ley de control (9) se selecciona porque en la situación ideal de perfecto conocimiento de los valores de los parámetros, la dinámica de error de lazo cerrado de las uniones está desacoplada en la forma:

$$\ddot{\mathbf{e}} + \mathbf{K}_v \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} = 0 \quad (10)$$

Pero este diseño se basa en la condición de que todos los parámetros sean conocidos.

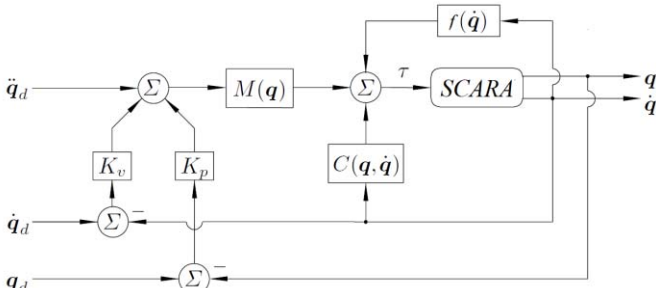


Fig. 1. Estructura de control PD.

B. Dinámica del Error con Ley de Control Aproximada

En algunos casos, $\mathbf{M}(\mathbf{q})$ no se conoce exactamente (por ejemplo, masa de carga útil desconocida) o $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}})$ no se conoce exactamente (por ejemplo, los términos de fricción desconocidos). Entonces $\hat{\mathbf{M}}(\mathbf{q})$ y $\hat{\mathbf{N}}(\mathbf{q}, \dot{\mathbf{q}})$ podrían ser la mejor estimación que tenemos para estos términos. Por otro lado, podríamos simplemente evitar calcular $\mathbf{M}(\mathbf{q})$ y $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}})$ en cada tiempo de muestreo, o el período de muestra podría ser demasiado corto para permitir este cálculo con el hardware disponible.

En este caso, para calcular la acción de control (9) se utiliza el modelo dinámico no lineal de un manipulador basado en parámetros estimados:

$$\hat{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} + \hat{\mathbf{N}}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{T} \quad (11)$$

Donde las cantidades $\hat{\mathbf{M}}(\mathbf{q})$ y $\hat{\mathbf{N}}(\mathbf{q}, \dot{\mathbf{q}})$ son estimaciones de $\mathbf{M}(\mathbf{q})$ y $\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}})$. Ahora, usando $\hat{\mathbf{M}}(\mathbf{q})$ y $\hat{\mathbf{N}}(\mathbf{q}, \dot{\mathbf{q}})$ obtendremos la dinámica de error si el controlador de torque aproximado (9) se aplica al brazo robótico SCARA (2).

$$\mathbf{T} = \hat{\mathbf{M}}(\ddot{\mathbf{q}}_d + \mathbf{K}_v \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e}) + \hat{\mathbf{N}} \quad (12)$$

Substituyendo \mathbf{T} en la ecuación de la dinámica del robot (2) permite:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \hat{\mathbf{M}}(\ddot{\mathbf{q}}_d + \mathbf{K}_v \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e}) + \hat{\mathbf{N}} \quad (13)$$

Reordenando (13)

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \hat{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}}_d = \hat{\mathbf{M}}(\mathbf{K}_v \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e}) + \hat{\mathbf{N}} - \mathbf{N} \quad (14)$$

Sumando y sustrayendo el término $\hat{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}}$, (14) puede ser reescrita como:

$$\hat{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{e}} = \hat{\mathbf{M}}(-\mathbf{K}_v \dot{\mathbf{e}} - \mathbf{K}_p \mathbf{e}) + \hat{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} + \tilde{\mathbf{N}} \quad (15)$$

Donde $\tilde{\mathbf{M}}(\mathbf{q}) = \mathbf{M}(\mathbf{q}) - \hat{\mathbf{M}}(\mathbf{q})$ y $\tilde{\mathbf{N}} = \mathbf{N} - \hat{\mathbf{N}}$

y el error de control se expresa como:

$$\ddot{\mathbf{e}} = (-\mathbf{K}_v \dot{\mathbf{e}} - \mathbf{K}_p \mathbf{e}) + \hat{\mathbf{M}}^{-1}(\tilde{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} + \tilde{\mathbf{N}}) \quad (16)$$

El término $\hat{\mathbf{M}}^{-1}(\tilde{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} + \tilde{\mathbf{N}})$ representa las matrices de variaciones de parámetros para el modelo dinámico utilizado en la ley de control de PD. Por lo tanto, es necesario reducir esta variación agregando una compensación adaptable para minimizar el error de seguimiento.

III. COMPENSADOR ADAPTABLE NEURONAL

A. Análisis de Estabilidad y Ajuste de Parámetros

En esta sección se diseñará una técnica de compensación basada en redes neuronales RBF, cuyo objetivo será ajustar las acciones de control (Torques) para compensar las variaciones dinámicas del robot, sin modificar los parámetros del controlador PD. En la Fig. 2 se muestra la estructura de control propuesta. Para evaluar esta estrategia, se utilizará el modelo identificado con $\hat{\mathbf{M}}(\mathbf{q})$ y $\hat{\mathbf{N}}(\mathbf{q}, \dot{\mathbf{q}})$ sobre el cual se diseñará la ley de control con compensación neuronal.

El torque de compensación \mathbf{T}_C será la señal aditiva al torque calculado por el controlador PD. Este torque se puede aproximar mediante un RBF-NN con aprendizaje en línea y configurando el vector regresor ζ de la RBF-NN como $\zeta = [q, \dot{q}, \ddot{q}]^T$, se propone:

$$\begin{aligned} \mathbf{T}_C &= \sum_{k=1}^m w_k^{*T} \exp\left(-\frac{1}{\rho_k^*} \|\zeta - c_k^*\|^2\right) + \boldsymbol{\varepsilon}_h \\ &= \mathbf{w}^{*T} \boldsymbol{\xi}^*(\zeta, c^*, \rho^*) + \boldsymbol{\varepsilon}_h \end{aligned} \quad (17)$$

donde \mathbf{w}^* es la matriz $m \times n$ (m indica el número de neuronas ocultas y n es el número de las entradas de control) y $\boldsymbol{\xi}$ es $m \times 1$. El término $\boldsymbol{\varepsilon}_h$ es el mínimo error para los parámetros óptimos de la red neuronal. Cabe aclarar que la notación de norma utilizada en todas las ecuaciones de este documento, corresponde a la norma euclídea.

Cada función gaussiana está determinada por los centros c y los anchos ρ ; para simplicidad ξ se usa en lugar de $\zeta(\zeta, c, \rho)$ a continuación:

$$\mathbf{T}_c = \mathbf{w}^{*T} \xi + \varepsilon_h \quad (18)$$

Añadiendo el par de compensación, entonces (13) puede ser expresado por:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \hat{\mathbf{M}}(\ddot{\mathbf{q}}_d + \mathbf{K}_v \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e}) + \hat{\mathbf{N}} - \mathbf{T}_c \quad (19)$$

Ahora, definiendo el error de vector de peso $\tilde{\mathbf{w}}$ como la diferencia entre \mathbf{w}^* (óptimo) y \mathbf{w} (instantáneo), y de la misma forma para $\tilde{\xi} = \xi^* - \xi$, (19) puede escribirse como:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) = \hat{\mathbf{M}}(\ddot{\mathbf{q}}_d + \mathbf{K}_v \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e}) + \dots + \hat{\mathbf{N}} - \mathbf{w}^T \xi - \tilde{\mathbf{w}}^T \xi - \mathbf{w}^T \tilde{\xi} - \tilde{\mathbf{w}}^T \tilde{\xi} - \varepsilon_h \quad (20)$$

Reordenando (20), resulta:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} - \hat{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}}_d = \hat{\mathbf{M}}(\mathbf{K}_v \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e}) + (\hat{\mathbf{N}} - \mathbf{N}) - \dots - \mathbf{w}^T \tilde{\xi} - (\tilde{\mathbf{w}}^T \xi + \mathbf{w}^T \tilde{\xi} + \tilde{\mathbf{w}}^T \tilde{\xi} + \varepsilon_h) \quad (21)$$

donde los términos $\tilde{\mathbf{w}}^T \xi + \mathbf{w}^T \tilde{\xi}$ representan el error de aprendizaje neuronal y el término $\tilde{\mathbf{w}}^T \tilde{\xi}$ (producto de dos errores) es considerado como parte de ε_h , ahora sumando el término $(\hat{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} - \hat{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}}_d)$, (20) puede escribirse:

$$\hat{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{e}} = \hat{\mathbf{M}}(-\mathbf{K}_v \dot{\mathbf{e}} - \mathbf{K}_p \mathbf{e}) + \tilde{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} + \tilde{\mathbf{N}} - \dots - \mathbf{w}^T \tilde{\xi} - (\tilde{\mathbf{w}}^T \xi + \mathbf{w}^T \tilde{\xi} + \varepsilon_h) \quad (22)$$

Haciendo que $(\tilde{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} + \tilde{\mathbf{N}}) = \mathbf{w}^T \tilde{\xi}$ entonces (22) es expresada:

$$\hat{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{e}} = \hat{\mathbf{M}}(-\mathbf{K}_v \dot{\mathbf{e}} - \mathbf{K}_p \mathbf{e}) - (\tilde{\mathbf{w}}^T \xi + \mathbf{w}^T \tilde{\xi} + \varepsilon_h) \quad (23)$$

y considerando (23), se puede escribir:

$$\ddot{\mathbf{e}} + \mathbf{K}_v \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} + \hat{\mathbf{M}}^{-1}(\tilde{\mathbf{w}}^T \xi + \mathbf{w}^T \tilde{\xi} + \varepsilon_h) = 0 \quad (24)$$

Usando una aproximación para la función $\tilde{\xi} = \tilde{\xi}^*(\zeta, \mathbf{c}^*, \boldsymbol{\rho}^*) - \hat{\xi}(\zeta, \hat{\mathbf{c}}, \hat{\boldsymbol{\rho}})$. Para lidiar con ξ^* , se aplica la expansión de Taylor de $\tilde{\xi}$ en los puntos $\mathbf{c}^* = \hat{\mathbf{c}}$ y $\boldsymbol{\rho}^* = \hat{\boldsymbol{\rho}}$.

$$\tilde{\xi}^*(\zeta, \mathbf{c}^*, \boldsymbol{\rho}^*) = \hat{\xi}(\zeta, \hat{\mathbf{c}}, \hat{\boldsymbol{\rho}}) + \boldsymbol{\theta}^T \tilde{\mathbf{c}} + \boldsymbol{\Lambda}^T \tilde{\boldsymbol{\eta}} + \mathbf{O}(\zeta, \tilde{\mathbf{c}}, \tilde{\boldsymbol{\rho}}) \quad (25)$$

donde $\mathbf{O}(\cdot)$ indica los argumentos de orden superior en una expansión de series de Taylor, y $\boldsymbol{\theta}$ y $\boldsymbol{\Lambda}$ son derivadas de $\tilde{\xi}^*(\zeta, \mathbf{c}^*, \boldsymbol{\rho}^*)$ con respecto a \mathbf{c}^* y a $\boldsymbol{\rho}^*$ en el punto $(\hat{\mathbf{c}}, \hat{\boldsymbol{\rho}})$. que pueden ser expresadas como $\boldsymbol{\theta}^T = [\partial \tilde{\xi}(\zeta, \mathbf{c}^*, \boldsymbol{\rho}^*) / \partial \mathbf{c}^*]$ y $\boldsymbol{\Lambda}^T = [\partial \tilde{\xi}(\zeta, \mathbf{c}^*, \boldsymbol{\rho}^*) / \partial \boldsymbol{\rho}^*]$ con $\mathbf{c}^* = \hat{\mathbf{c}}$ y $\boldsymbol{\rho}^* = \hat{\boldsymbol{\rho}}$.

Ahora de (25) pasando $\hat{\xi}$ al otro miembro se obtiene:

$$\tilde{\xi} = \boldsymbol{\theta}^T \tilde{\mathbf{c}} + \boldsymbol{\Lambda}^T \tilde{\boldsymbol{\eta}} + \mathbf{O}(\zeta, \tilde{\mathbf{c}}, \tilde{\boldsymbol{\rho}}) \quad (26)$$

De (26) el término de orden superior $\mathbf{O}(\cdot)$ es acotado por:

$$\|\mathbf{O}(\zeta, \tilde{\mathbf{c}}, \tilde{\boldsymbol{\rho}})\| = \|\tilde{\xi}(\cdot) - \boldsymbol{\theta}^T \tilde{\mathbf{c}} - \boldsymbol{\Lambda}^T \tilde{\boldsymbol{\rho}}\| \leq \dots \leq \|\tilde{\xi}(\cdot)\| + \|\boldsymbol{\theta}^T \tilde{\mathbf{c}}\| + \|\boldsymbol{\Lambda}^T \tilde{\boldsymbol{\rho}}\| \leq b_1 + b_2 \|\tilde{\mathbf{c}}\| + b_3 \|\tilde{\boldsymbol{\rho}}\| \leq \mathbf{O}_{Max} \quad (27)$$

donde b_1 , b_2 , y b_3 son constantes debido al hecho de que las funciones neuronales (RBF) y sus derivadas están limitadas por b_1 , b_2 , y b_3 . Sustituyendo (26) en (24), se obtiene:

$$\ddot{\mathbf{e}} + \mathbf{K}_v \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} + \dots + \hat{\mathbf{M}}^{-1}(\tilde{\mathbf{w}}^T \xi + \mathbf{w}^T \boldsymbol{\theta}^T \tilde{\mathbf{c}} + \mathbf{w}^T \boldsymbol{\Lambda}^T \tilde{\boldsymbol{\rho}} + \mathbf{w}^T \mathbf{O}(\cdot) + \varepsilon_h) = 0 \quad (28)$$

Definiendo a la suma $\mathbf{w}^T \mathbf{O}(\cdot) + \varepsilon_h = \varepsilon_H$ donde se supone que esta suma está delimitada por:

$$\|\varepsilon_H\| \leq \varepsilon_{Max} \quad (29)$$

El término ε_{Max} es una constante y ε_H representa el error de aproximación de la compensación neuronal. Utilizando la teoría de aproximación neuronal, el error de aproximación inherente ε_h puede reducirse arbitrariamente aumentando el número de m [20], entonces es razonable suponer que ε_H esté limitado por una constante ε_{Max} ; de esta forma (28) puede reescribirse como:

$$\ddot{\mathbf{e}} + \mathbf{K}_v \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e} + \dots + \hat{\mathbf{M}}^{-1}(\tilde{\mathbf{w}}^T \xi + \mathbf{w}^T \boldsymbol{\theta}^T \tilde{\mathbf{c}} + \mathbf{w}^T \boldsymbol{\Lambda}^T \tilde{\boldsymbol{\rho}} + \varepsilon_H) = 0 \quad (30)$$

Con las matrices de ganancia seleccionadas para el funcionamiento del sistema en lazo cerrado, (30) puede reescribirse en el espacio de estados de la forma siguiente:

$$\dot{\mathbf{E}} = \mathbf{A}\mathbf{E} - \mathbf{B}\hat{\mathbf{M}}^{-1}(\tilde{\mathbf{w}}^T \xi + \mathbf{w}^T \boldsymbol{\theta}^T \tilde{\mathbf{c}} + \mathbf{w}^T \boldsymbol{\Lambda}^T \tilde{\boldsymbol{\rho}} + \varepsilon_H) \quad (31)$$

donde:

$$\mathbf{E} = \begin{bmatrix} \mathbf{e} \\ \dot{\mathbf{e}} \end{bmatrix}, \mathbf{A} = \begin{bmatrix} 0 & \mathbf{I} \\ -\mathbf{K}_p & -\mathbf{K}_v \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ \mathbf{I} \end{bmatrix} \quad (32)$$

La ecuación (31) será utilizada para demostrar la convergencia de la técnica propuesta en el análisis de estabilidad en la subsección siguiente.

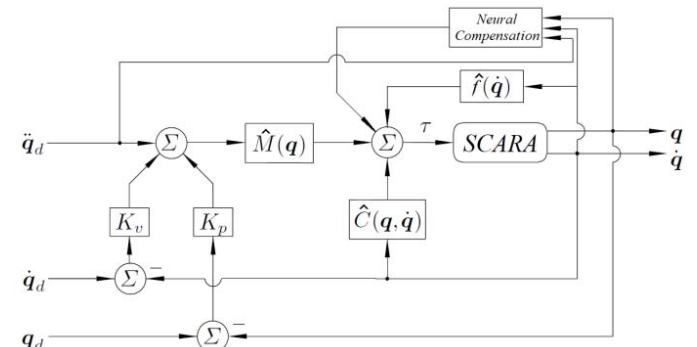


Fig. 2. PD con compensación neuronal

B. Análisis de Estabilidad y Ajuste de Parámetros

Para obtener la ley de ajuste estable, se elige la siguiente función candidata de Lyapunov:

$$V = \frac{1}{2} \mathbf{E}^T \mathbf{P} \mathbf{E} + \frac{1}{2} \text{tr} \left(\frac{\tilde{\mathbf{w}}^T \tilde{\mathbf{w}}}{\alpha} \right) + \dots + \frac{1}{2} \text{tr} \left(\frac{\tilde{\mathbf{c}}^T \tilde{\mathbf{c}}}{\beta} \right) + \frac{1}{2} \text{tr} \left(\frac{\tilde{\boldsymbol{\rho}}^T \tilde{\boldsymbol{\rho}}}{\gamma} \right) \quad (33)$$

donde \mathbf{P} es una matriz diagonal definida positiva $2n \times 2n$, el término tr es el operador traza y α, β, γ son constantes no negativas mayores que cero. La derivada de la función de Lyapunov está dada por:

$$\dot{V} = \frac{1}{2} (\mathbf{E}^T \mathbf{P} \dot{\mathbf{E}} + \dot{\mathbf{E}}^T \mathbf{P} \mathbf{E}) + \text{tr} \left(\frac{\dot{\tilde{\mathbf{w}}}^T \tilde{\mathbf{w}}}{\alpha} \right) + \dots + \text{tr} \left(\frac{\dot{\tilde{\mathbf{c}}}^T \tilde{\mathbf{c}}}{\beta} \right) + \text{tr} \left(\frac{\dot{\tilde{\boldsymbol{\rho}}}^T \tilde{\boldsymbol{\rho}}}{\gamma} \right) \quad (34)$$

Sustituyendo la (33) en (34), resulta:

$$\dot{V} = \frac{1}{2} \mathbf{E}^T \mathbf{P} \mathbf{A} \mathbf{E} + \frac{1}{2} \mathbf{E}^T \mathbf{A}^T \mathbf{P} \mathbf{E} + \mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \tilde{\mathbf{w}}^T \xi + \dots + \mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \mathbf{w}^T \boldsymbol{\theta}^T \tilde{\mathbf{c}} + \mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \mathbf{w}^T \boldsymbol{\Lambda}^T \tilde{\boldsymbol{\rho}} + \mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \boldsymbol{\varepsilon}_H + \dots + \text{tr} \left(\frac{\dot{\tilde{\mathbf{w}}}^T \tilde{\mathbf{w}}}{\alpha} \right) + \text{tr} \left(\frac{\dot{\tilde{\mathbf{c}}}^T \tilde{\mathbf{c}}}{\beta} \right) + \text{tr} \left(\frac{\dot{\tilde{\boldsymbol{\rho}}}^T \tilde{\boldsymbol{\rho}}}{\gamma} \right) \quad (35)$$

Conociendo que $-2\mathbf{Q} = (\mathbf{P}\mathbf{A} + \mathbf{A}^T\mathbf{P})$, siendo \mathbf{A} una matriz Hurwitz, la solución de la ecuación de Lyapunov puede ser encontrada y siempre tiene una solución única.

$$\dot{V} = -\frac{1}{2} \mathbf{E}^T \mathbf{Q} \mathbf{E} + \mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \tilde{\mathbf{w}}^T \xi + \dots + \mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \mathbf{w}^T \boldsymbol{\theta}^T \tilde{\mathbf{c}} + \mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \mathbf{w}^T \boldsymbol{\Lambda}^T \tilde{\boldsymbol{\rho}} + \mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \boldsymbol{\varepsilon}_H + \dots + \text{tr} \left(\frac{\dot{\tilde{\mathbf{w}}}^T \tilde{\mathbf{w}}}{\alpha} \right) + \text{tr} \left(\frac{\dot{\tilde{\mathbf{c}}}^T \tilde{\mathbf{c}}}{\beta} \right) + \text{tr} \left(\frac{\dot{\tilde{\boldsymbol{\rho}}}^T \tilde{\boldsymbol{\rho}}}{\gamma} \right) \quad (36)$$

Aplicando el operador traza al segundo, tercero y cuarto término de (36) queda expresada como:

$$\dot{V} = -\frac{1}{2} \mathbf{E}^T \mathbf{Q} \mathbf{E} + \text{tr} (\mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \tilde{\mathbf{w}}^T \xi) + \dots + \text{tr} (\mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \mathbf{w}^T \boldsymbol{\theta}^T \tilde{\mathbf{c}}) + \text{tr} (\mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \mathbf{w}^T \boldsymbol{\Lambda}^T \tilde{\boldsymbol{\rho}}) + \dots + \mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \boldsymbol{\varepsilon}_H + \text{tr} \left(\frac{\dot{\tilde{\mathbf{w}}}^T \tilde{\mathbf{w}}}{\alpha} \right) + \text{tr} \left(\frac{\dot{\tilde{\mathbf{c}}}^T \tilde{\mathbf{c}}}{\beta} \right) + \text{tr} \left(\frac{\dot{\tilde{\boldsymbol{\rho}}}^T \tilde{\boldsymbol{\rho}}}{\gamma} \right) \quad (37)$$

Reordenando (37) y aplicando la propiedad asociativa del operador traza obtenemos:

$$\dot{V} = -\mathbf{E}^T \mathbf{Q} \mathbf{E} + \text{tr} \left(\mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \tilde{\mathbf{w}}^T \xi + \frac{\tilde{\mathbf{w}}^T \dot{\tilde{\mathbf{w}}}}{\alpha} \right) + \text{tr} \left(\mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \mathbf{w}^T \boldsymbol{\theta}^T \tilde{\mathbf{c}} + \frac{\tilde{\mathbf{c}}^T \dot{\tilde{\mathbf{c}}}}{\beta} \right) + \dots + \text{tr} \left(\mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \mathbf{w}^T \boldsymbol{\Lambda}^T \tilde{\boldsymbol{\rho}} + \frac{\tilde{\boldsymbol{\rho}}^T \dot{\tilde{\boldsymbol{\rho}}}}{\gamma} \right) + \dots + \mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \boldsymbol{\varepsilon}_H \quad (38)$$

Aplicando la propiedad cíclica de traza en el segundo término de (38) y reordenando se obtiene:

$$\dot{V} = -\mathbf{E}^T \mathbf{Q} \mathbf{E} + \text{tr} \left[\tilde{\mathbf{w}}^T \left(\xi \mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} + \frac{\dot{\tilde{\mathbf{w}}}}{\alpha} \right) \right] + \text{tr} \left[\left(\mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \mathbf{w}^T \boldsymbol{\theta}^T + \frac{\dot{\tilde{\mathbf{c}}}}{\beta} \right) \tilde{\mathbf{c}} \right] + \dots + \text{tr} \left[\left(\mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \mathbf{w}^T \boldsymbol{\Lambda}^T + \frac{\dot{\tilde{\boldsymbol{\rho}}}}{\gamma} \right) \tilde{\boldsymbol{\rho}} \right] + \dots + \mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \boldsymbol{\varepsilon}_H \quad (39)$$

Igualando a cero los términos entre paréntesis de (39) y despejando $\dot{\tilde{\mathbf{w}}}, \dot{\tilde{\mathbf{c}}}, \dot{\tilde{\boldsymbol{\rho}}}$, obtenemos cada una de las leyes de ajuste de los parámetros del compensador neuronal:

$$\dot{\tilde{\mathbf{w}}} = -\alpha \xi \mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \quad (40)$$

$$\dot{\tilde{\mathbf{c}}}^T = -\beta \mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \mathbf{w}^T \boldsymbol{\theta}^T \quad (41)$$

$$\dot{\tilde{\boldsymbol{\rho}}}^T = -\gamma \mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \mathbf{w}^T \boldsymbol{\Lambda}^T \quad (42)$$

Reemplazando cada una de las leyes de ajuste en (39) queda expresada como:

$$\dot{V} = -\mathbf{E}^T \mathbf{Q} \mathbf{E} + \mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \boldsymbol{\varepsilon}_H \quad (43)$$

Entonces \dot{V} se puede demostrar que es negativa de acuerdo con (43),

$$\dot{V} \leq -\lambda(\mathbf{Q}) \|\mathbf{E}\|^2 \frac{\lambda(\mathbf{P}) \|\mathbf{B}\|_{\text{MaxMax}}}{\|\hat{\mathbf{M}}\| \|\mathbf{E}\|_{\text{min}}} \quad (44)$$

cumpliendo con $\dot{V} < 0$ cuando:

$$\|\mathbf{E}\| \geq \frac{2\lambda_{\text{max}}(\mathbf{P}) \|\mathbf{B}\| \varepsilon_{\text{Max}}}{\lambda_{\text{min}}(\mathbf{Q}) \|\hat{\mathbf{M}}\|} = \psi_e \quad (45)$$

Ahora considerando $\dot{\tilde{\mathbf{w}}} = \dot{\mathbf{w}}^* - \dot{\mathbf{w}}, \dot{\tilde{\mathbf{c}}} = \dot{\mathbf{c}}^* - \dot{\mathbf{c}}, \dot{\tilde{\boldsymbol{\rho}}} = \dot{\boldsymbol{\rho}}^* - \dot{\boldsymbol{\rho}}$ y $\dot{\mathbf{w}}^* = 0, \dot{\mathbf{c}}^* = 0, \dot{\boldsymbol{\rho}}^* = 0$ las reglas de ajuste son:

$$\dot{\mathbf{w}} = \alpha \xi \mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \quad (46)$$

$$\dot{\mathbf{c}}^T = \beta \mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \mathbf{w}^T \boldsymbol{\theta}^T \quad (47)$$

$$\dot{\boldsymbol{\rho}}^T = \gamma \mathbf{E}^T \mathbf{P} \mathbf{B} \hat{\mathbf{M}}^{-1} \mathbf{w}^T \boldsymbol{\Lambda}^T \quad (48)$$

La ecuación (45) implica que las normas de errores de control están limitadas en última instancia por el valor:

$$\psi_e = \frac{2\lambda_{\text{max}}(\mathbf{P}) \|\mathbf{B}\| \varepsilon_{\text{Max}}}{\lambda_{\text{min}}(\mathbf{Q}) \|\hat{\mathbf{M}}\|} \quad (49)$$

Este es un resultado práctico que permite afirmar que el error de control está limitado en términos del error de aproximación NN.

IV. RESULTADOS DE SIMULACIÓN

Para demostrar la viabilidad de la técnica propuesta se trabajó con el modelo identificado de un robot SCARA Bosch SR-800, El controlador propuesto se implementó en conjunto con el modelo identificado del robot. El objetivo es que el robot SCARA describa una trayectoria en forma de 8. Este tipo de trayectoria se elige debido a que resulta exigente para los sistemas robóticos y permite evaluar mejor el desempeño de los sistemas de control propuestos para dichos sistemas. En este caso, el controlador PD fue sacado de sintonía para que de esta forma se pueda representar la situación en que el robot es cargado produciendo una variación de parámetros en la dinámica del mismo. Este hecho produce que el controlador PD no se encuentre en sintonía con la dinámica nominal del robot, por lo tanto, la red neuronal deberá compensar esa diferencia de parámetros tratando de disminuir el error existente entre la trayectoria deseada y la trayectoria seguida por el robot. En las siguientes figuras se puede observar los resultados obtenidos. La Fig.4 muestra la trayectoria realizada con controlares PD y PID, además se superpone la trayectoria obtenida con compensación neuronal, donde puede verse un buen desempeño en cuanto al seguimiento de la trayectoria solicitada. La Fig.5 muestra la trayectoria de referencia deseada (q_{1ref} , q_{2ref}) y la real de cada unión articular (q_1 y q_2) utilizando la compensación neuronal, comparada con otras técnicas de control clásicas. Por otro lado, la Fig.6 muestra diferentes acciones de control, donde la línea continua representa combinación de la acción de control del controlador PD sumada con la compensación neuronal y la línea de trazo representa la acción de control de la compensación neuronal únicamente.



Fig. 3. Robot SCARA Bosch SR-800.

La Fig. 7 muestra los errores cuadráticos medios obtenidos con tres propuestas de control, donde se puede apreciar que el control PD con compensación neuronal presenta una reducción del 20% del error en promedio con respecto al controlador PD. En comparación con el control PID, este presenta mejoras con respecto al PD y en algunos tramos de la trayectoria el error inclusive llega a ser menor que el PD con compensación neuronal, pero el error promedio en toda la trayectoria de la técnica de compensación es el menor de todas las propuestas.

Esta técnica permite obtener un control adaptable sin intervenir sobre los ajustes del controlador estático. Además, es

simple de implementar, y no necesita un entrenamiento previo pudiendo ser colocada en paralelo junto con el controlador clásico PD previamente existente.

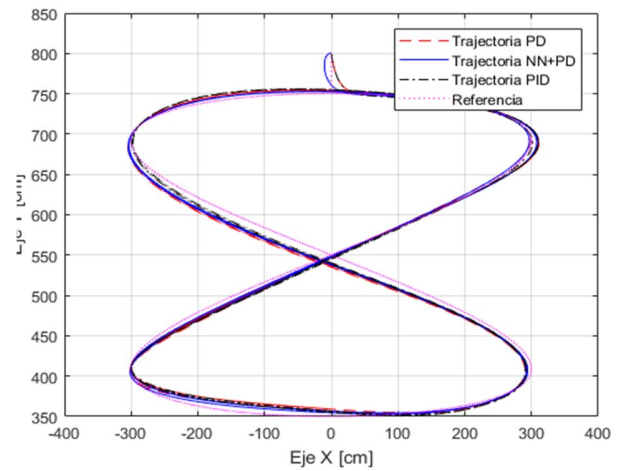


Fig. 4. Trayectoria de referencia y seguida por el Robot con el controlador PD, PID y PD con compensación Neuronal.

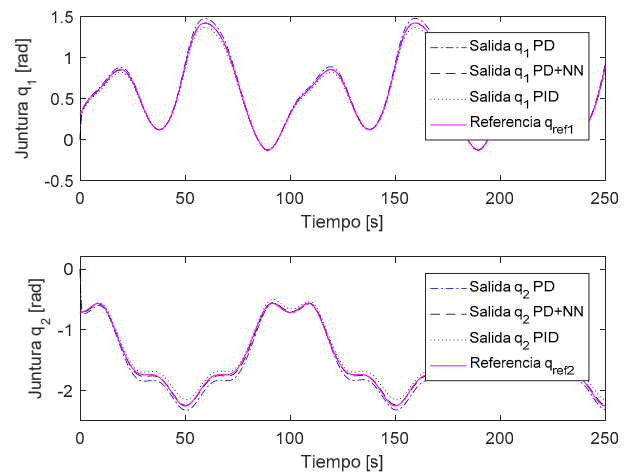


Fig. 5. Trayectoria deseada y real de cada unión articular al emplear los controladores PD, PID y PD con compensación neuronal.

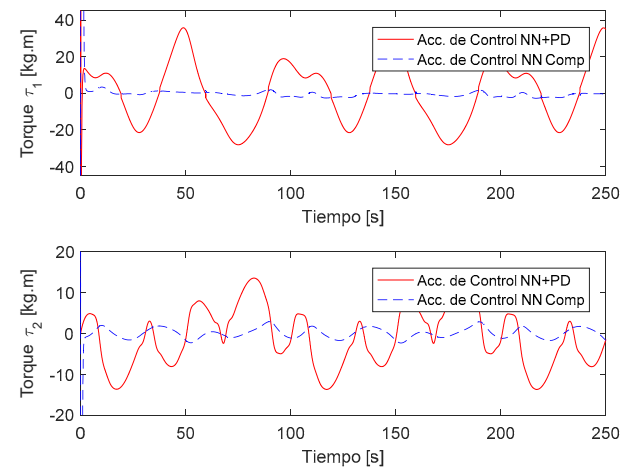


Fig. 6. Acciones de control, PD con compensación neuronal (línea continua) y señal de compensación neuronal (línea de trazo).

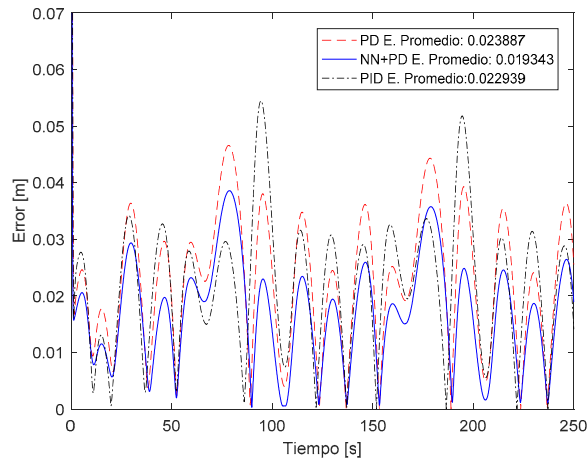


Fig. 7. Comparación del error cuadrático medio entre el controlador PD, PID y PD con compensación neuronal.

V. CONCLUSIONES

En este trabajo, se propuso un compensador adaptable basado en red neuronal del tipo de función de base radial (RBF) que tiene la ventaja de ser muy flexible, ya que puede ser usado en paralelo con cualquier controlador estático previamente definido (PD, PID, etc.) y en diferentes sistemas robóticos. La ley de aprendizaje se obtiene mediante la aplicación del método de Lyapunov, obteniéndose un algoritmo de ajuste de parámetros neuronales estable y completamente sintonizado; es decir que se ajustan tanto los pesos de salida de la red neuronal como los centros y ancho de las funciones gaussianas. Los resultados obtenidos, demuestran que se puede mejorar en forma significativa (20%) la aproximación a la trayectoria deseada en situaciones de variaciones dinámicas del robot manipulador, aportando así robustez al sistema de control.

REFERENCIAS

- [1] C. C. de Wit, B. Siciliano and G. Bastin, In *Theory of robot control*, Londres, Inglaterra, Springer Science & Business Media. 2012.
- [2] M. H. Korayem, R. Shiri and S. R. Nekoo, "Sliding mode control design with adaptable gains for robotic manipulators," In *3rd RSI International Conference on Robotics and Mechatronics (ICROM)*, Tehran, 2015, pp. 037-042.
- [3] D. Prabu, S. Kumar and R. Prasad, "Dynamic Control of Three-Link SCARA Manipulator using Adaptive Neuro Fuzzy Inference System," In *IEEE International Conference on Networking, Sensing and Control*, Sanya, 2008, pp. 1609-1614.
- [4] R. Sharma and M. Gopal, "A Markov Game-Adaptive Fuzzy Controller for Robot Manipulators," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 1, pp. 171-186, 2008.
- [5] S. Li, L. Jin, and M. A. Mirza, In *Kinematic Control of Redundant Robot Arms Using Neural Networks: A Theoretical Study*. Wiley-IEEE Press. West Sussex, Inglaterra, 2019.
- [6] A. Ohri, and S. Salim, "Neural Network based control of Robot Manipulator," *World Scientific News*, pp.14-21, 2019.
- [7] R. Kelly, V. Santibáñez and A. Loria, In *Control of robot manipulators in joint space*, Leipzig, Alemania, Springer Science & Business Media, 2006.
- [8] F. L. Lewis, D. M. Dawson and C. T. Abdallah, In *Robot manipulator control: theory and practice*, Nueva York, USA, Mersel Dekker, Inc., 2003.
- [9] F. G. Rossomando and C. M. Soria, "Adaptive neural sliding mode control in discrete time for a SCARA robot arm," *IEEE Latin America Transactions*, vol. 14, no. 6, pp. 2556-2564, 2016.

- [10] F. G. Rossomando and C. M. Soria, "Discrete-time sliding mode neuro-adaptive controller for SCARA robot arm". *Neural Computing and Applications*, Vol. 28, no. 12, pp.3837-3850, 2017.
- [11] F. G. Rossomando, C. M. Soria, E. O. Freire and R. O. Carelli, "Sliding Mode Neuro-Adaptive Controller Designed in Discrete Time for Mobile Robots," *Mechatronic Systems and Control*, Vol. 46, no. 2, pp. 55-63, 2018.
- [12] V. T. Yen, W. Y. Nan and P. Van Cuong, "Robust Adaptive Sliding Mode Neural Networks Control for Industrial Robot manipulators," *International Journal of Control, Automation and Systems*, Vol. 17, no. 3, pp. 783-792, 2019.
- [13] E. O. Freire, F. G. Rossomando, C. M. Soria, and R. O. Carelli, "Self-tuning of a Neuro-Adaptive PID Controller for a SCARA Robot Based on Neural Network," *IEEE Latin America Transactions*, Vol. 16, no.5, pp. 1364-1374, 2018.
- [14] J. Moreno-Valenzuela, C. Aguilar-Avelar, S. Puga-Guzmán, and V. Santibáñez, "Two adaptive control strategies for trajectory tracking of the inertia wheel pendulum: neural networks vis à vis model regressor," *Intelligent Automation & Soft Computing*, Vol. 23, no. 1, pp. 63-73, 2017.
- [15] S. A. Puga-Guzmán, C. Aguilar-Avelar, J. Moreno-Valenzuela and V. Santibáñez, "Tracking of periodic oscillations in an underactuated system via adaptive neural networks," *Journal of Low Frequency Noise, Vibration and Active Control*, Vol. 37, no. 1, pp. 128-143, 2018.
- [16] J. Moreno-Valenzuela, C. Aguilar-Avelar, S. A. Puga-Guzmán and V. Santibáñez, "Adaptive neural network control for the trajectory tracking of the Furuta pendulum," *IEEE transactions on cybernetics*, Vol. 46, no. 12, pp. 3439-3452, 2016.
- [17] F. Luan, J. Na, Y. Huang and G. Gao, "Adaptive Neural Network Control for Robotic Manipulators with Guaranteed Finite-Time Convergence," *Neurocomputing*, Vol. 337, pp. 153-164, 2019.
- [18] M.S. Galicki, "An Adaptive Regulator of Robotic Manipulators in the Task Space," *IEEE Transactions on Automatic Control*, Vol. 53, no. 4, pp. 1058-1062, 2008.
- [19] L. Jong Shin, "A Compensation Control Method Using Neural Network for Mechanical Deflection Error in SCARA Robot with Random Payload," *Jou. of Korean Soc. of Mechanical Technology*, Vol. 13, no. 3 2011.
- [20] S. Haykin, In *Neural networks: A Comprehensive Foundation*, 1st ed., Prentice Hall PTR Upper Saddle River, NJ, USA, 1994.