

# Controlador PD con compensación Neuro-Adaptable aplicado a la dinámica de un RMR Omnidireccional<sup>\*</sup>

S. López<sup>\*</sup> Miguel A. Llama<sup>\*</sup> Ramón García-Hernández<sup>\*</sup>

<sup>\*</sup> *Tecnológico Nacional de México/I. T. La Laguna, Torreón 27000, Mexico, (e-mail: {m.slopezh, mllama, rgarciah}@correo.itlalaguna.edu.mx).*

**Resumen:** El esquema presentado en este artículo es un controlador neuro-adaptable para solucionar el problema de seguimiento de trayectoria de un robot móvil con ruedas mecanum omnidireccionales suponiendo incertidumbres paramétricas. Los pesos de la red neuronal artificial se actualizan en línea utilizando filtrado del error y leyes adaptables. El controlador neuro-adaptable se valida en simulación y se compara con un PID clásico obteniendo mejores resultados para el esquema propuesto.

*Palabras clave:* Robot móvil Nexus, control neuronal, control adaptable, robot móvil omnidireccional, control en seguimiento, actualización de pesos en línea, ruedas mecanum.

## 1. INTRODUCCIÓN

Gracias a los avances tecnológicos el interés en los vehículos autónomos se ha incrementado, y a su vez ha aumentado el interés en el desarrollo y validación de leyes de control para los mismos. La maniobrabilidad de los robots móviles con ruedas (por sus siglas RMR) dependerá de las ruedas y motores usados. Una propuesta interesante son los móviles omnidireccionales con cuatro ruedas mecanum (Hendzel et al., 2017), los cuales se estudian en el presente trabajo.

En seguida se enlistan trabajos relacionados al control de RMR omnidireccionales. En (Ovalle et al., 2019) se busca resolver el problema de seguimiento de trayectorias de un RMR omnidireccional Nexus con ruedas mecanum. Se utiliza un observador de alto orden por modos deslizantes además de diferentes técnicas de control robusto basadas en modos deslizantes, todo esto se valida de manera experimental. En (Lin y Shih, 2013) se presenta el modelo de un RMR omnidireccional con ruedas mecanum; a este modelo se le aplica un control adaptable basado en la técnica de *Backstepping*, se presentan resultados en simulación. En (Alakshendra y Chiddarwar, 2017b) se pretende controlar la posición y orientación de un RMR con ruedas mecanum con un cilindro balanceándose en la parte superior del robot, se permuta entre dos controladores robustos por modos deslizantes de segundo orden. En (Alakshendra y Chiddarwar, 2017a) se presenta un control robusto adaptable por modos deslizantes de segundo

orden aplicado a un RMR con ruedas mecanum, además se valida experimentalmente. Otros trabajos relacionados se encuentran en (Lu et al., 2018), (Chen et al., 2018) y (Borisov et al., 2015).

Por su parte al hablar de control neuronal se debe hacer distinción entre identificación en lazo abierto y control por realimentación en lazo cerrado; este último brinda la posibilidad de implementar algoritmos de aprendizaje en línea que no requieren de ajustes preliminares fuera de línea (Lewis et al., 1998), (Lewis y Ge, 2006), (Lewis, 1999) y (Dawson et al., 2003). A continuación se mencionan algunos trabajos que se refieren al control neuronal por realimentación como aquellos que no requieren etapa de aprendizaje fuera de línea. En (Lewis, 1999) se expone una estructura para el diseño de controladores usando redes neuronales y lógica difusa; se presentan arquitecturas de control por realimentación y algoritmos de sintonización de pesos para asegurar la estabilidad en lazo cerrado y el acotamiento de los pesos. En (Kwan et al., 1995) se propone un controlador neuronal de compensación deseada basada en leyes adaptables para conseguir el control robusto de robots manipuladores de eslabones rígidos; se incluye un estudio de simulación comparativa con diferentes controladores robustos y adaptables. En (Montoya-Cháirez et al., 2019) se implementan diferentes controladores para seguimiento de trayectorias aplicado a un giroscopio de dos grados de libertad subactuado. Entre los controladores analizados se tiene un controlador por realimentación de estados, un algoritmo neuro-adaptable, un esquema adaptable por modelo del regresor, un control PD clásico y un controlador PID-PID en cascada; las propuestas se evalúan de manera experimental. En (Moreno-

<sup>\*</sup> Los autores agradecen el apoyo financiero de los proyectos de investigación del Tecnológico Nacional de México/I. T. La Laguna y del CONACYT 711820.

Valenzuela et al., 2016) se presenta un control neuro-adaptable para resolver el problema de control de un péndulo de Furuta, también se hace una comparación con otros esquemas de manera experimental. Otros trabajos relacionados se encuentran en (Sun et al., 2011), (Wang et al., 2009), (Hernandez et al., 2011), (Selmic y Lewis, 2000) y (Ha y Han, 2004).

En el presente trabajo se propone implementar un esquema neuro-adaptable que compense los términos desconocidos del sistema. Los pesos de la red neuronal se actualizan en línea utilizando filtrado del error y leyes adaptables; para complementar este compensador se añade un PD clásico. Los parámetros dinámicos son considerados desconocidos pero constantes. Este esquema se puede considerar una versión neuronal del esquema llamado "Controlador adaptable de Slotine and Li" (Slotine y Li, 1987), (Slotine y Li, 1991), (Dawson et al., 2003) y (Kelly et al., 2006). Este trabajo está organizado de la siguiente manera. En la sección 2 se presenta el modelo dinámico de un RMR omnidireccional. En la sección 3 se presenta el controlador neuro-adaptable propuesto. En la sección 4 se muestran los resultados en simulación de la aplicación del esquema neuro-adaptable al robot móvil Nexus perturbado, además de una comparación con un PID clásico. En la sección 5 se exponen las conclusiones.

## 2. MODELO DINÁMICO DEL RMR NEXUS

La forma más común de resolver el problema de control de un RMR omnidireccional es a través de su modelo cinemático (Viboonchaicheep et al., 2003), (Tsai et al., 2011) y (Siegwart et al., 2011), lo anterior provoca imprecisiones al aumentar la velocidad de movimiento del robot. En esta sección se describe el modelo dinámico del RMR Nexus.

### 2.1 Modelo dinámico del RMR Nexus

El modelo dinámico de posición y orientación de un robot móvil omnidireccional con respecto al marco inercial está dado por (Campion et al., 1996):

$$\mathbf{R}^T(\theta)\mathbf{M}_R\ddot{\mathbf{q}} + \mathbf{E}^T\mathbf{I}\dot{\boldsymbol{\varphi}} = \mathbf{E}^T\boldsymbol{\tau}_\varphi \quad (1)$$

donde  $\mathbf{q} = [x \ y \ \theta]^T$  representa las variables de configuración en el espacio de trabajo,  $\boldsymbol{\varphi} = [\varphi_1 \ \varphi_2 \ \varphi_3 \ \varphi_4]^T$  representa la posición angular de cada rueda,

$$\boldsymbol{\tau}_\varphi = [\tau_{\varphi_1} \ \tau_{\varphi_2} \ \tau_{\varphi_3} \ \tau_{\varphi_4}]^T$$

es el vector de par aplicado a cada rueda,

$$\mathbf{M}_R = \text{Diag}\{m_1 + 4m_2, m_1 + 4m_2, 4m_2(l_1^2 + l_2^2) + J_1 + 4J_3\}$$

es la matriz de inercia del robot,  $\mathbf{I} = \text{Diag}\{J_2, J_2, J_2, J_2\}$  es la matriz de inercia de las cuatro ruedas,

$$\mathbf{E} = \frac{1}{r} \begin{bmatrix} 1 & 1 & l_1 + l_2 \\ 1 & -1 & -(l_1 + l_2) \\ 1 & 1 & -(l_1 + l_2) \\ 1 & -1 & l_1 + l_2 \end{bmatrix},$$

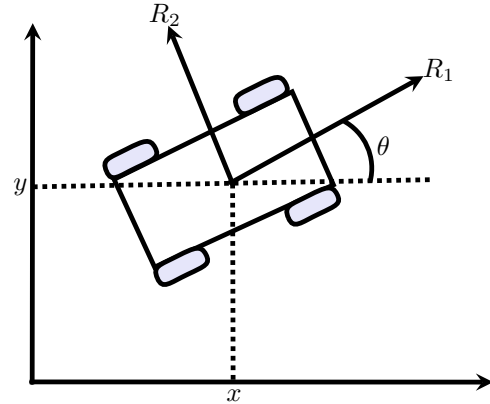


Figura 1. Diagrama de un robot móvil con ruedas.

representa la matriz jacobiana del sistema,  $m_1$  es la masa del cuerpo,  $m_2$  es la masa de cada rueda,  $J_1$  es la inercia del cuerpo,  $J_2$  es la inercia de las ruedas sobre el eje del motor,  $J_3$  es la inercia de las ruedas perpendicular al eje del motor,  $l_1$  es la distancia del centro geométrico al eje de cada una de las ruedas sobre el eje  $R_1$ ,  $l_2$  es la distancia del centro geométrico al eje de cada una de las ruedas sobre el eje  $R_2$ ,  $r$  es el radio de las ruedas, y

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

es la matriz de rotación para el movimiento planar.

Considerando la relación  $\dot{\boldsymbol{\varphi}} = \mathbf{E}\mathbf{R}(\theta)^T\dot{\mathbf{q}}$ , es posible mostrar que la dinámica (1) está dada por

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}(\dot{\theta})\dot{\mathbf{q}} = \boldsymbol{\tau} + \boldsymbol{\delta}, \quad (2)$$

donde  $\boldsymbol{\tau} = [\tau_1 \ \tau_2 \ \tau_3]^T$  es el vector de fuerzas generalizadas,  $\boldsymbol{\delta} = [\delta_1 \ \delta_2 \ \delta_3]^T$  es un vector que contiene fuerzas de perturbación para cada variable de configuración,  $\mathbf{M} \in \mathbb{R}^{3 \times 3}$  es la matriz de inercia y  $\mathbf{C}(\dot{\theta}) \in \mathbb{R}^{3 \times 3}$  es la matriz antisimétrica de fuerzas centrífugas y de Coriolis; estas últimas están dadas de la siguiente manera

$$\mathbf{M} = \mathbf{M}_R + \mathbf{I}\mathbf{E}^T\mathbf{E},$$

$$\mathbf{C}(\dot{\theta}) = \frac{4}{r}\mathbf{I}\dot{\theta}\mathbf{B},$$

donde

$$\mathbf{B} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Se debe observar que  $\mathbf{M}$  es una matriz diagonal constante. La ecuación

$$\boldsymbol{\tau} = \mathbf{R}(\theta)\mathbf{E}^T\boldsymbol{\tau}_\varphi$$

relaciona las fuerzas generalizadas con los pares aplicados a los actuadores. Ya que la matriz  $\mathbf{E}$  no es invertible es necesario auxiliarse de su pseudoinversa.

### 3. DISEÑO DEL CONTROLADOR

En esta sección se plantea un esquema de control para seguimiento de trayectorias, el controlador propuesto tiene la forma de un controlador PD más una compensación neuro-adaptable. El diseño del controlador parte de la suposición de que el par deseado puede ser aproximado por una red neuronal, entonces pueden estimarse en línea los pesos de la red neuronal mediante las leyes de adaptación adecuadas.

#### 3.1 Planteamiento del problema

Considérese el modelo dinámico de posición y orientación de un RMR omnidireccional con ruedas mecanum y actuadores ideales (2) donde  $\mathbf{M} \in \mathbb{R}^{3 \times 3}$  es la matriz de inercia,  $\mathbf{C}(\dot{\theta})\dot{\mathbf{q}} \in \mathbb{R}^3$  es el vector de fuerzas centrífugas y de Coriolis y  $\boldsymbol{\tau} \in \mathbb{R}^3$  es el vector de fuerzas externas y pares aplicados a los actuadores. Los vectores  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$ ,  $\ddot{\mathbf{q}} \in \mathbb{R}^3$  expresan la posición, velocidad y aceleración respectivamente.

Para conseguir el seguimiento de trayectorias se definen los siguientes errores

$$\tilde{\mathbf{q}} = \mathbf{q}_d - \mathbf{q}, \quad \dot{\tilde{\mathbf{q}}} = \dot{\mathbf{q}}_d - \dot{\mathbf{q}}.$$

Es necesario considerar el siguiente filtrado del error

$$\mathbf{r} = \dot{\tilde{\mathbf{q}}} + \boldsymbol{\Lambda}\tilde{\mathbf{q}}$$

donde  $\boldsymbol{\Lambda} \in \mathbb{R}^{3 \times 3}$  es una matriz simétrica definida positiva (o de manera más general  $-\boldsymbol{\Lambda}$  es Hurwitz). Se expresa el sistema (2) en función de  $\mathbf{r} \in \mathbb{R}^3$  considerando que

$$\dot{\mathbf{q}} = -\mathbf{r} + \dot{\mathbf{q}}_d + \boldsymbol{\Lambda}\tilde{\mathbf{q}}, \quad \ddot{\mathbf{q}} = -\dot{\mathbf{r}} + \ddot{\mathbf{q}}_d + \boldsymbol{\Lambda}\dot{\tilde{\mathbf{q}}},$$

entonces resulta

$$\mathbf{M}\dot{\mathbf{r}} = -\mathbf{C}(\dot{\theta})\mathbf{r} + \mathbf{M}(\ddot{\mathbf{q}}_d + \boldsymbol{\Lambda}\dot{\tilde{\mathbf{q}}}) + \mathbf{C}(\dot{\theta})(\dot{\mathbf{q}}_d + \boldsymbol{\Lambda}\tilde{\mathbf{q}}) - \boldsymbol{\tau} - \boldsymbol{\delta}.$$

Ahora se define

$$\mathbf{u} = \ddot{\mathbf{q}}_d + \boldsymbol{\Lambda}\dot{\tilde{\mathbf{q}}}, \quad \mathbf{v} = \dot{\mathbf{q}}_d + \boldsymbol{\Lambda}\tilde{\mathbf{q}} \quad \text{y}$$

$$\mathbf{f} = \mathbf{W}^T \boldsymbol{\Phi}(\mathbf{x}) + \boldsymbol{\varepsilon} = -\mathbf{C}(\dot{\theta})\mathbf{r} + \mathbf{M}\mathbf{u} + \mathbf{C}(\dot{\theta})\mathbf{v} - \boldsymbol{\delta},$$

siendo  $\boldsymbol{\varepsilon} \in \mathbb{R}^3$  el error de aproximación, lo que resulta en el sistema expresado en función del filtrado del error

$$\mathbf{M}\dot{\mathbf{r}} = \mathbf{W}^T \boldsymbol{\Phi}(\mathbf{x}) + \boldsymbol{\varepsilon} - \boldsymbol{\tau},$$

donde  $\boldsymbol{\Phi}(\mathbf{x}) \in \mathbb{R}^L$  es el vector de funciones de activación de la red neuronal,  $\mathbf{W} \in \mathbb{R}^{L \times 3}$  es la matriz de pesos desconocidos del sistema (en este caso no hay una correspondencia con los parámetros de la dinámica) y  $\mathbf{x} \in \mathbb{R}^L$  es el vector de entradas de la red neuronal el cual se elige de manera holística, en este caso se propone

$$\mathbf{x} \equiv \begin{bmatrix} \tilde{\mathbf{q}}^T & \dot{\tilde{\mathbf{q}}}^T & \mathbf{q}_d^T & \dot{\mathbf{q}}_d^T & \ddot{\mathbf{q}}_d^T & 1 \end{bmatrix}^T, \quad (3)$$

lo que resulta en  $L = 16$ .

En (Lewis et al., 1998) se propone la siguiente ley de control neuro-adaptable (ver figura 2)

$$\boldsymbol{\tau} = \widehat{\mathbf{W}}^T \boldsymbol{\Phi}(\mathbf{x}) + \mathbf{K}_v \mathbf{r}, \quad (4)$$

$$\dot{\widehat{\mathbf{W}}} = \boldsymbol{\Gamma} \boldsymbol{\Phi}(\mathbf{x}) \mathbf{r}^T, \quad (5)$$

donde  $\boldsymbol{\Gamma} \in \mathbb{R}^{3 \times 3}$  es la matriz de ganancias ajustables, que generalmente se elige como una matriz diagonal con elementos positivos, y  $\mathbf{K}_v \in \mathbb{R}^{3 \times 3}$  es una matriz simétrica definida positiva ajustable denominada ganancia de velocidad. Definiendo  $\boldsymbol{\Lambda} = \mathbf{K}_v^{-1} \mathbf{K}_p$ , donde  $\mathbf{K}_p \in \mathbb{R}^{3 \times 3}$  es una matriz simétrica definida positiva ajustable denominada ganancia de posición, esto lleva a que el término  $\mathbf{K}_v \mathbf{r}$  sea meramente un PD. Este controlador neuro-adaptable produce una matriz de pesos estimados  $\widehat{\mathbf{W}}$  para la matriz de pesos desconocidos  $\mathbf{W}$  usando (5), así el controlador tiene su propia dinámica. El estimado de la función no lineal  $\mathbf{f} \in \mathbb{R}^3$  está dada por la siguiente red neuronal

$$\hat{\mathbf{f}} = \widehat{\mathbf{W}}^T \boldsymbol{\Phi}(\mathbf{x}),$$

además suponiendo que la matriz de pesos desconocidos  $\mathbf{W}$  es constante y definiendo

$$\tilde{\mathbf{f}} = \mathbf{f} - \hat{\mathbf{f}} = \mathbf{W}^T \boldsymbol{\Phi}(\mathbf{x}) + \boldsymbol{\varepsilon} - \widehat{\mathbf{W}}^T \boldsymbol{\Phi}(\mathbf{x}) = \widetilde{\mathbf{W}}^T \boldsymbol{\Phi}(\mathbf{x}) + \boldsymbol{\varepsilon},$$

con el error de estimación de pesos definido como  $\widetilde{\mathbf{W}} = \mathbf{W} - \widehat{\mathbf{W}}$  y considerando el error de aproximación  $\boldsymbol{\varepsilon} = \mathbf{0}$ . Por tanto, usando la ley de control propuesta (4-5) resulta la siguiente dinámica en lazo cerrado

$$\mathbf{M}\dot{\mathbf{r}} = -\mathbf{K}_v \mathbf{r} + \widetilde{\mathbf{W}}^T \boldsymbol{\Phi}(\mathbf{x}). \quad (6)$$

La ecuación de lazo cerrado formado por las ecuaciones (5) y (6) puede reescribirse como

$$\frac{d}{dt} \begin{bmatrix} \tilde{\mathbf{q}} \\ \dot{\tilde{\mathbf{q}}} \end{bmatrix} = \begin{bmatrix} \dot{\tilde{\mathbf{q}}} \\ \mathbf{M}^{-1} \left[ \widetilde{\mathbf{W}}^T \boldsymbol{\Phi}(\mathbf{x}) - \mathbf{K}_v \left[ \dot{\tilde{\mathbf{q}}} + \boldsymbol{\Lambda}\tilde{\mathbf{q}} \right] \right] - \boldsymbol{\Lambda}\dot{\tilde{\mathbf{q}}} \end{bmatrix} \quad (7)$$

$$\dot{\widetilde{\mathbf{W}}} = -\boldsymbol{\Gamma} \boldsymbol{\Phi}(\mathbf{x}) \left[ \dot{\tilde{\mathbf{q}}} + \boldsymbol{\Lambda}\tilde{\mathbf{q}} \right], \quad (8)$$

la cual es una ecuación diferencial no autónoma y el origen  $[\tilde{\mathbf{q}}, \dot{\tilde{\mathbf{q}}}]^T = \mathbf{0}$ ,  $\widetilde{\mathbf{W}} = \mathbf{0}$  es un punto de equilibrio.

Eligiendo la siguiente función candidata de Lyapunov

$$V = \frac{1}{2} \mathbf{r}^T \mathbf{M} \mathbf{r} + \tilde{\mathbf{q}}^T \mathbf{K}_p \tilde{\mathbf{q}} + \frac{1}{2} \text{tr} \left( \widetilde{\mathbf{W}}^T \boldsymbol{\Gamma}^{-1} \widetilde{\mathbf{W}} \right), \quad (9)$$

entonces la derivada temporal de  $V$  está dada por

$$\dot{V} = \mathbf{r}^T \mathbf{M} \dot{\mathbf{r}} + 2\tilde{\mathbf{q}}^T \mathbf{K}_p \dot{\tilde{\mathbf{q}}} + \text{tr} \left( \widetilde{\mathbf{W}}^T \boldsymbol{\Gamma}^{-1} \dot{\widetilde{\mathbf{W}}} \right).$$

Sustituyendo (6) en la ecuación anterior se produce

$$\begin{aligned} \dot{V} &= \mathbf{r}^T \left( -\mathbf{K}_v \mathbf{r} + \widetilde{\mathbf{W}}^T \boldsymbol{\Phi}(\mathbf{x}) \right) \\ &+ 2\tilde{\mathbf{q}}^T \mathbf{K}_p \dot{\tilde{\mathbf{q}}} + \text{tr} \left( \widetilde{\mathbf{W}}^T \boldsymbol{\Gamma}^{-1} \dot{\widetilde{\mathbf{W}}} \right), \\ \dot{V} &= -\mathbf{r}^T \mathbf{K}_v \mathbf{r} + \mathbf{r}^T \widetilde{\mathbf{W}}^T \boldsymbol{\Phi}(\mathbf{x}) \\ &+ 2\tilde{\mathbf{q}}^T \mathbf{K}_p \dot{\tilde{\mathbf{q}}} + \text{tr} \left( \widetilde{\mathbf{W}}^T \boldsymbol{\Gamma}^{-1} \dot{\widetilde{\mathbf{W}}} \right), \end{aligned} \quad (10)$$

posteriormente se debe saber que

$$\mathbf{a}^T \mathbf{b} = \text{tr}(\mathbf{b} \mathbf{a}^T), \quad \mathbf{r}^T \widetilde{\mathbf{W}}^T \boldsymbol{\Phi}(\mathbf{x}) = \text{tr} \left( \widetilde{\mathbf{W}}^T \boldsymbol{\Phi}(\mathbf{x}) \mathbf{r}^T \right),$$

así (10) resulta en

$$\dot{V} = -\mathbf{r}^T \mathbf{K}_v \mathbf{r} + 2\tilde{\mathbf{q}}^T \mathbf{K}_p \dot{\tilde{\mathbf{q}}} + \text{tr} \left( \widetilde{\mathbf{W}}^T \left[ \boldsymbol{\Gamma}^{-1} \dot{\widetilde{\mathbf{W}}} + \boldsymbol{\Phi}(\mathbf{x}) \mathbf{r}^T \right] \right).$$

Ahora seleccionando  $\dot{\widetilde{\mathbf{W}}} = -\dot{\widehat{\mathbf{W}}} = -\boldsymbol{\Gamma} \boldsymbol{\Phi}(\mathbf{x}) \mathbf{r}^T$ , se produce

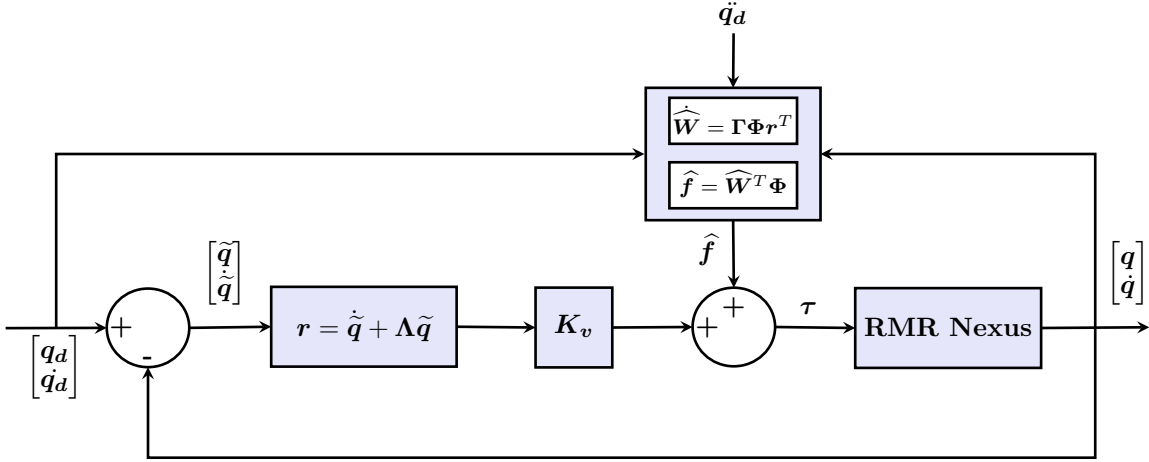


Figura 2. Diagrama del controlador PD con compensación neuro-adaptable.

$$\begin{aligned}
\dot{V} &= -\mathbf{r}^T \mathbf{K}_v \mathbf{r} + 2\tilde{\mathbf{q}}^T \mathbf{K}_p \dot{\tilde{\mathbf{q}}}, \\
\dot{V} &= -\left[\dot{\tilde{\mathbf{q}}}^T + \tilde{\mathbf{q}}^T \mathbf{\Lambda}^T\right] \mathbf{K}_v \left[\dot{\tilde{\mathbf{q}}} + \mathbf{\Lambda} \tilde{\mathbf{q}}\right] + 2\tilde{\mathbf{q}}^T \mathbf{K}_p \dot{\tilde{\mathbf{q}}}, \\
\dot{V} &= -\dot{\tilde{\mathbf{q}}}^T \mathbf{K}_v \dot{\tilde{\mathbf{q}}} - \tilde{\mathbf{q}}^T \mathbf{\Lambda}^T \mathbf{K}_v \mathbf{\Lambda} \tilde{\mathbf{q}} - \dot{\tilde{\mathbf{q}}}^T \mathbf{K}_v \mathbf{\Lambda} \tilde{\mathbf{q}} \\
&\quad - \tilde{\mathbf{q}}^T \mathbf{\Lambda}^T \mathbf{K}_v \dot{\tilde{\mathbf{q}}} + 2\tilde{\mathbf{q}}^T \mathbf{K}_p \dot{\tilde{\mathbf{q}}}, \\
\dot{V} &= -\dot{\tilde{\mathbf{q}}}^T \mathbf{K}_v \dot{\tilde{\mathbf{q}}} - \tilde{\mathbf{q}}^T \mathbf{\Lambda}^T \mathbf{K}_p \tilde{\mathbf{q}}, \\
\dot{V} &= -\begin{bmatrix} \tilde{\mathbf{q}} \\ \dot{\tilde{\mathbf{q}}} \end{bmatrix}^T \begin{bmatrix} \mathbf{\Lambda}^T \mathbf{K}_p & 0 & 0 \\ 0 & \mathbf{K}_v & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{q}} \\ \dot{\tilde{\mathbf{q}}} \end{bmatrix}. \quad (11)
\end{aligned}$$

Recordando que  $\mathbf{\Lambda}$  es una matriz no singular,  $\mathbf{K}_v$  es una matriz simétrica definida positiva, luego  $\mathbf{\Lambda}^T \mathbf{K}_p$  será también una matriz simétrica definida positiva. Lo anterior conlleva a que  $\dot{V}$  expresada en (11) sea una función semi-definida negativa. Además la función candidata de Lyapunov  $V$  en (9) es definida positiva, radialmente desacotada y decreciente, esto garantiza que el origen de la ecuación de lazo cerrado (7-8) es uniformemente estable y todas las soluciones están acotadas para cualquier condición inicial.

#### 4. APLICACIÓN AL RMR NEXUS

En esta sección se presentan los resultados del esquema implementado en simulación; se presenta también una comparación con un control PID clásico.

##### 4.1 Especificaciones del controlador neuro-adaptable

Las redes neuronales suelen usar diferentes tipos de funciones de activación tanto lineales como no lineales, como lo son sigmoideas, escalones, signo, de base radial, tangente hiperbólica entre otras, en este caso se seleccionó de manera holística la función tangente hiperbólica

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

por ser la función que mejor desempeñaba la tarea de entre las funciones relu, de base radial y sigmoideas. El vector de funciones de activación está dado por

$$\Phi(\mathbf{x}) = [\tanh(x_1) \cdots \tanh(x_L)]^T,$$

donde  $\mathbf{x} \in \mathbb{R}^L$  es el vector de entradas de la red neuronal que se selecciona como se muestra en (3).

Las ganancias que se aplicaron al controlador (4-5) están dadas por

$$\begin{aligned}
\mathbf{K}_v &= \text{diag}\{30.80, 30.80, 23.10\}, \\
\mathbf{K}_p &= \text{diag}\{33.88, 33.88, 25.41\}, \\
\mathbf{\Lambda} &= \text{diag}\{0.9091, 0.9091, 0.9091\}, \\
\mathbf{\Gamma} &= \text{diag}\{108.11, 108.11, 108.11\}.
\end{aligned}$$

##### 4.2 Parámetros del RMR Nexus

Los resultados en simulación se llevaron a cabo considerando el modelo (2) y los parámetros del robot móvil Nexus, los cuales se muestran en la tabla 1 (Ovalle et al., 2019).

Tabla 1. Parámetros físicos del robot Nexus

Notación	Valor	Unidad
$m_1$	2.8	Kg
$m_2$	0.38	Kg
$J_1$	0.0608	Kg m <sup>2</sup>
$J_2$	$3.24 \times 10^{-4}$	Kg m <sup>2</sup>
$J_3$	$4.69 \times 10^{-4}$	Kg m <sup>2</sup>
$l_1$	0.1524	m
$l_2$	0.1505	m
$r$	0.042	m

##### 4.3 Perturbaciones

Se consideró un vector de perturbaciones para evaluar el desempeño del controlador, la perturbación está

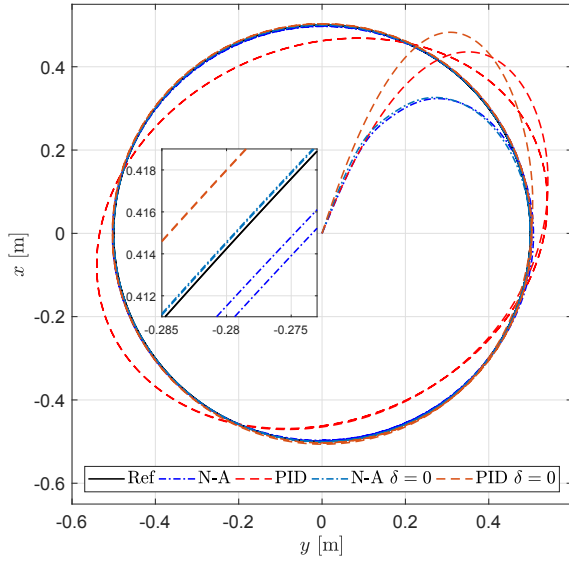


Figura 3. Respuesta del sistema en seguimiento de posición.

dada por (Ovalle et al., 2019)  $\delta = [2.5 \sin(\frac{\pi}{10}t) + 0.1, -2.5 \cos(\frac{\pi}{10}t) + 0.1, 2.5 \cos(\frac{\pi}{10}t) + 0.1]^T$ .

#### 4.4 Control PID

Con propósitos de comparación se presentan los resultados en simulación de un controlador PID el cual está dado por

$$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}} + K_i \int_0^t \tilde{q}(\sigma) d\sigma,$$

donde

$$\begin{aligned} K_p &= \text{diag}\{30.80, 30.80, 23.10\}, \\ K_v &= \text{diag}\{30.80, 30.80, 23.10\}, \\ K_i &= \text{diag}\{13.86, 13.86, 23.10\}. \end{aligned}$$

Se llegó a estas ganancias buscando la respuesta más rápida sin sobrepasar las capacidades de los actuadores.

#### 4.5 Resultados en simulación

Enseguida se muestran los resultados en simulación del esquema aplicado. Las simulaciones fueron realizadas con las siguientes señales de referencia  $q_d = [5 \sin(\frac{\pi}{10}t), 5 \cos(\frac{\pi}{10}t), -\frac{\pi}{10}t + \frac{\pi}{2}]$ . En la figura 3 se muestra la respuesta en seguimiento de posición tanto del esquema neuro-adaptable como el PID clásico contra la referencia. En la figura 4 se muestran los valores RMS para el error de las variables de configuración, este criterio está dado por

$$\tilde{q}_{RMS} = \sqrt{\frac{1}{\Delta T} \int_{t-\Delta T}^t \tilde{q}(\sigma)^T \tilde{q}(\sigma) d\sigma},$$

con  $\Delta T = 5$  [seg]. En la figura 5 se muestran los errores de seguimiento para evaluar su desempeño. Finalmente, en la figura 6 se muestran los pesos estimados de la red neuronal.

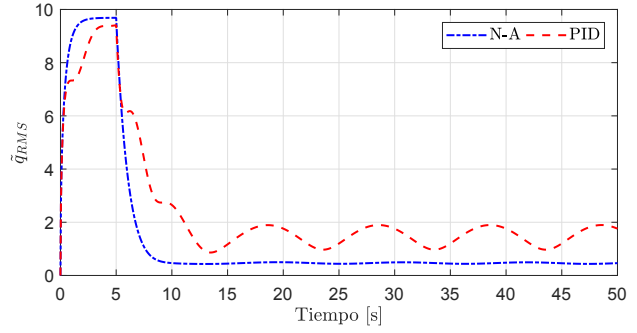


Figura 4. Valor RMS para el error de las variables de configuración.

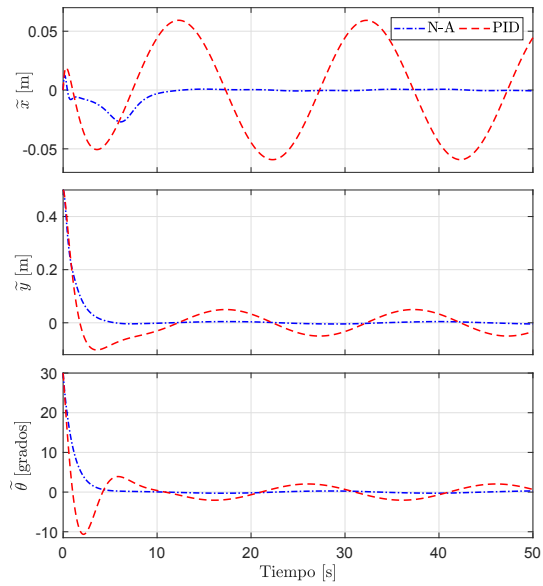


Figura 5. Errores de seguimiento.

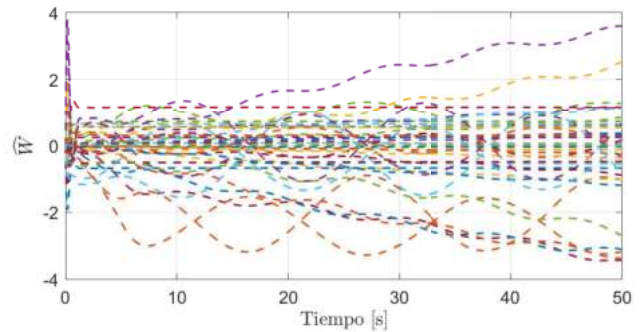


Figura 6. Pesos estimados  $\widehat{W}$  de la red neuronal.

## 5. CONCLUSIONES

En la comparación que se realizó, el control neuro-adaptable mostró ser superior al PID, además de ser capaz de compensar ciertas perturbaciones. El controlador neuro-adaptable propuesto funciona en un gran número de sistemas sin la necesidad de un modelado extenso ni un

análisis previo para encontrar una matriz de regresión. A diferencia de los controladores adaptables convencionales, este esquema no requiere linealidad en los parámetros o un análisis preliminar largo y tedioso para determinar una matriz de regresión; además, no condiciona el cumplimiento de requisitos como excitación persistente o el principio de equivalencia cierta.

## REFERENCIAS

- Alakshendra, V. y Chiddarwar, S.S. (2017a). Adaptive robust control of mecanum-wheeled mobile robot with uncertainties. *Nonlinear Dynamics*, 87(4), 2147–2169.
- Alakshendra, V. y Chiddarwar, S.S. (2017b). Simultaneous balancing and trajectory tracking control for an omnidirectional mobile robot with a cylinder using switching between two robust controllers. *International Journal of Advanced Robotic Systems*, 14(6), 1729881417738728. doi:10.1177/1729881417738728.
- Borisov, A.V., Kilin, A.A., y Mamaev, I.S. (2015). Dynamics and control of an omniwheel vehicle. *Regular and Chaotic Dynamics*, 20(2), 153–172.
- Campion, G., Bastin, G., y Dandrea-Novet, B. (1996). Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 12(1), 47–62. doi:10.1109/70.481750.
- Chen, T., Xie, L., Han, Y., y Luo, J. (2018). Lane keeping control on mecanum wheeled omnidirectional vehicles using laser scanner. En *2018 Chinese Control And Decision Conference (CCDC)*, 3404–3409. doi: 10.1109/CCDC.2018.8407712.
- Dawson, D.M., Abdallah, C.T., y Lewis, F.L. (2003). *Robot manipulator control: theory and practice*. CRC Press.
- Ha, I.C. y Han, M.C. (2004). A robust control with a neural network structure for uncertain robot manipulator. *KSME International Journal*, 18(11), 1916. doi: 10.1007/BF02990433.
- Henzel, Z. et al. (2017). Modelling of dynamics of a wheeled mobile robot with mecanum wheels with the use of Lagrange equations of the second kind. *International Journal of Applied Mechanics and Engineering*, 22(1), 81–99.
- Hernandez, D., Yu, W., y Moreno-Armendariz, M.A. (2011). Neural PD control with second-order sliding mode compensation for robot manipulators. En *The 2011 International Joint Conference on Neural Networks*, 2395–2402. doi:10.1109/IJCNN.2011.6033529.
- Kelly, R., Davila, V., y Perez, J. (2006). *Control of Robot Manipulators in Joint Space*. Advanced Textbooks in Control and Signal Processing. Springer London.
- Kwan, C., Dawson, D., y Lewis, F. (1995). Robust adaptive control of robots using neural network: global tracking stability. En *Proceedings of 1995 34th IEEE Conference on Decision and Control*, volumen 2, 1846–1851. IEEE.
- Lewis, F.L. y Ge, S.S. (2006). *Neural Networks in Feedback Control Systems*, 791–825. John Wiley & Sons, Ltd. doi:10.1002/0471777455.ch19.
- Lewis, F.L., Yesildirak, A., y Jagannathan, S. (1998). *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Taylor & Francis, Inc., Bristol, PA, USA.
- Lewis, F. (1999). Nonlinear network structures for feedback control. *Asian Journal of Control*, 1(4), 205–228.
- Lin, L.C. y Shih, H.Y. (2013). Modeling and adaptive control of an omni-mecanum-wheeled robot. *Intelligent Control and Automation*, 4(02), 166.
- Lu, X., Zhang, X., Zhang, G., y Jia, S. (2018). Design of adaptive sliding mode controller for four-mecanum wheel mobile robot. En *2018 37th Chinese Control Conference (CCC)*, 3983–3987. doi: 10.23919/ChiCC.2018.8483388.
- Montoya-Cháirez, J., Santibáñez, V., y Moreno-Valenzuela, J. (2019). Adaptive control schemes applied to a control moment gyroscope of 2 degrees of freedom. *Mechatronics*, 57, 73 – 85. doi: https://doi.org/10.1016/j.mechatronics.2018.11.011.
- Moreno-Valenzuela, J., Aguilar-Avelar, C., Puga-Guzmán, S.A., y Santibáñez, V. (2016). Adaptive neural network control for the trajectory tracking of the furuta pendulum. *IEEE Transactions on Cybernetics*, 46(12), 3439–3452. doi:10.1109/TCYB.2015.2509863.
- Ovalle, L., Ríos, H., Llama, M., Santibáñez, V., y Dzul, A. (2019). Omnidirectional mobile robot robust tracking: Sliding-mode output-based control approaches. *Control Engineering Practice*, 85, 50 – 58. doi: https://doi.org/10.1016/j.conengprac.2019.01.002.
- Selmic, R.R. y Lewis, F.L. (2000). Backlash compensation in nonlinear systems using dynamic inversion by neural networks. *Asian Journal of Control*, 2(2), 76–87.
- Siegwart, R., Nourbakhsh, I.R., y Scaramuzza, D. (2011). *Introduction to autonomous mobile robots*. MIT press.
- Slotine, J.J.E. y Li, W. (1987). On the adaptive control of robot manipulators. *The International Journal of Robotics Research*, 6(3), 49–59. doi: 10.1177/027836498700600303.
- Slotine, J. y Li, W. (1991). *Applied Nonlinear Control*. Prentice Hall.
- Sun, T., Pei, H., Pan, Y., Zhou, H., y Zhang, C. (2011). Neural network-based sliding mode adaptive control for robot manipulators. *Neurocomputing*, 74(14-15), 2377–2384.
- Tsai, C., Tai, F., y Lee, Y. (2011). Motion controller design and embedded realization for mecanum wheeled omnidirectional robots. En *2011 9th World Congress on Intelligent Control and Automation*, 546–551. doi: 10.1109/WCICA.2011.5970573.
- Viboonchaiheep, P., Shimada, A., y Kosaka, Y. (2003). Position rectification control for mecanum wheeled omni-directional vehicles. En *IECON'03. 29th Annual Conference of the IEEE Industrial Electronics Society (IEEE Cat. No.03CH37468)*, volumen 1, 854–859 vol.1. doi:10.1109/IECON.2003.1280094.
- Wang, L., Chai, T., y Fang, Z. (2009). Neural-network-based two-loop control of robotic manipulators including actuator dynamics in task space. *Journal of Control Theory and Applications*, 7(2), 112.