

Control neuro-robusto del sistema líder-seguidor de agentes móviles Neuro-robust control of leader-follower system of mobile agents

D. Rodríguez-Castellanos ^{a,*}, G. Solís-Perales ^a, M. Blas-Valdez ^a

^aDivisión de Tecnologías para la Integración Ciber-Humana, Departamento de Ciencias Computacionales, Universidad de Guadalajara, 44430, Guadalajara, Jalisco, México.

Resumen

En este trabajo haciendo uso de linealización por retroalimentación de salida se propone un controlador neuro-robusto basado en una red neuronal recurrente de alto orden, para el seguimiento del robot móvil diferencial. El seguimiento se realiza en las velocidades angulares de las ruedas. La estrategia consiste en diseñar un controlador linealizante y un estimador basado en redes neuronales para estimar las incertidumbres paramétricas y posibles discrepancias en el modelo. De esta manera solo se requieren las velocidades angulares como variables de salida. Se presenta una simulación numérica para ilustrar la propuesta.

Palabras Clave: Red Neuronal Recurrente de Alto Orden (RHONN), Robot Móvil Diferencial, Seguimiento Asintótico, Control Neuro-Robusto.

Abstract

In this work applying output feedback linearization, a neuro-robust controller based on a recurrent high-order neural network is proposed, applied to differential-drive mobile robot tracking. The tracking is done on the the wheels angular velocities. The strategy consists in desing a linearizing controller and an estimator based on neural networks for the parametric uncertainties and possible discrepancies in the model. In this way only angular velocities are required as output variables. A numerical simulation is presented to illustrate the proposal.

Keywords: Recurrent High-Order Neural Networks (RHONN), Differential-Drive Mobile Robot, Asymptotic Output Tracking, Neuro-Robust Control.

1. Introducción

En la actualidad el uso de robots móviles autónomos se incrementa gracias a su gran versatilidad de aplicación y su amplio espacio de trabajo. Los sistemas multi-robot han tenido un progreso significativo en lo que corresponde a su coordinación y cooperación adquiriendo mayor importancia en aplicaciones cotidianas. Algunos ejemplos son: Roomba un pequeño robot móvil aspiradora el cual calcula la ruta de limpieza óptima a medida que va limpiando (iRobot Corporation, 2022). En la planta de Mercedes-Benz Türk en Aksaray, se utiliza la plataforma móvil KMP 1500 de KUKA para transportar las diferentes cabinas de conducción de camiones a los diferentes procesos en la línea de producción (KUKA, 2022). El Q3-600C de Hik Robot es un robot móvil para almacenes han sido aplicados en el centro de distribución de DHL en Shanghai logrando un incremento del 33 % en la eficiencia de trabajo y un 40 %

de utilización del espacio de almacenamiento (Hikrobot, 2022). Múltiples agentes pueden resolver tareas más rápido y a un menor costo si trabajan cooperativamente, esto comparado con un solo robot intentando realizar las mismas tareas. La estrategia líder-seguidor es el método más común en el control de formación de sistemas multi-robot (Wang *et al.*, 2016),(Lian *et al.*, 2019) y (Soorki *et al.*, 2011).

Para el presente trabajo realizaremos el seguimiento de las velocidades angulares de ambas ruedas, siendo así la trayectoria deseada en el plano cartesiano parametrizada en velocidades angulares de las ruedas derecha e izquierda. Uno de los problemas a resolver en este tema es el seguimiento de trayectoria, se han llevado a cabo diversas investigaciones cuyos resultados han permitido un avance significativo para la solución de esta problemática. En (Rabbani y Memon, 2021) se propone un controlador de seguimiento de trayectoria y estabilización para un robot móvil diferencial mediante backstepping recursivo

*Autor para correspondencia: david.rcastellanos@academicos.udg.mx

Correo electrónico: david.rcastellanos@academicos.udg.mx (David Rodríguez-Castellanos), gualberto.solis@academicos.udg.mx (Gualberto Solís-Perales), marco.blas5697@alumnos.udg.mx (Marco Blas-Valdez).

integral para el cual es necesario el conocimiento de todos los parámetros y estados. En (Hassan y Saleem, 2022) se propone un controlador híbrido que combina un controlador cinemático basado en una red neuronal y un modelo de referencia de control adaptativo. En este trabajo los parámetros del controlador son adaptados en línea usando la red neuronal. La sintonización del controlador cinemático de forma adaptable logra una rápida convergencia a la trayectoria deseada. El modelo de control adaptable de referencia conserva el rendimiento de seguimiento deseado cuando se producen incertidumbres en los parámetros y el modelo. En (Meng *et al.*, 2018) utilizan un controlador PID para la velocidad angular de los motores del robot móvil y en (Majid *et al.*, 2016) la dinámica de los parámetros del robot es controlada usando dos bloques de control PID. Ambos casos tienen como limitación la precisión del modelo del robot. El robot móvil diferencial es el más común debido a los recursos mínimos empleados para su diseño y construcción además de su capacidad para trasladarse a cualquier posición mediante rectas, curvas o giros sobre su propio eje. Sin embargo, son sistemas que presentan parámetros que por lo general son inciertos, esto es presentan discrepancias en los modelos, perturbaciones externas, incertidumbre en los parámetros e inercia, etc. Estas incertidumbres, discrepancias en el modelo y perturbaciones externas son dificultades que se deben compensar mediante un sistema de control. Se propone un sistema compuesto por dos partes, el diseño de un controlador linealizante y un estimador con una red neuronal de alto orden.

En la linealización la idea principal es transformar de forma parcial o total, la dinámica no lineal del sistema a controlar en una dinámica linealizable mediante el control y así poder aplicar las técnicas de control lineal ya conocidas. Este método tiene un número importante de limitaciones:

1. No puede ser usado para todos los sistemas.
2. El estado completo tiene que ser medido.
3. No garantiza robustez en presencia de incertidumbres no estructuradas o dinámicas no modeladas.

Como se menciona la principal desventaja de esta técnica de control es que requiere del conocimiento de la dinámica no lineal del sistema y sus estados, sin embargo, existen resultados donde se puede reducir el número de estados a conocer, y más aún, las dinámicas no lineales pueden ser estimadas para que el controlador las pueda contrarrestar, por esto se considera un control robusto como se reporto en (Femat *et al.*, 1999) y (Femat y Solis-Perales, 2008).

Las redes neuronales recurrentes se basan principalmente en el modelo de Hopfield (Hopfield, 1984). Estas redes son consideradas como buenas candidatas para aplicaciones de sistemas no lineales que enfrentan incertidumbres y son atractivas por su fácil implementación, estructura relativamente simple, robustez y capacidad para ajustar sus parámetros en línea (Ricalde, 2005), es por esto que para lograr lo mencionado anteriormente, aquí se propone el uso de las redes neuronales recurrentes de alto orden que estimen las funciones no lineales inciertas a partir solo de la medición de las velocidades y que el controlador linealizante las tenga disponibles para compensarlas en el sistema, de esta manera el controlador compensará o atenuará el efecto de las variaciones paramétricas, errores en el modelo o perturbaciones.

El presente trabajo está organizado de la siguiente manera: En las secciones 2 y 3 se presenta la teoría preliminar para el desarrollo de la síntesis de control propuesta. La sección 4 presenta el diseño del controlador neuro-robusto para el robot móvil diferencial. Los resultados de la simulación realizada se muestran en la sección 5. La sección 6 presenta una breve comparación de resultados del controlador propuesto con un controlador PID. Finalmente en la sección 7 se presentan las conclusiones de este trabajo.

2. Linealización exacta mediante retroalimentación de estados

Los siguientes antecedentes fueron tomados de (Isidori, 1995).

El sistema propuesto como ya se mencionó consiste en transformar el sistema no lineal del modelo del agente móvil, entonces el sistema es de la siguiente forma:

$$\begin{aligned}\dot{x} &= f(x) + \sum_{i=1}^m g_i(x)u_i \\ y_1 &= h_1(x) \\ &\dots \\ y_m &= h_m(x)\end{aligned}\quad (1)$$

donde $f(x), g_1(x) \dots g_m(x)$ son campos vectoriales suaves, y $h_1(x) \dots h_m(x)$ funciones suaves que definen las salidas del sistema, estas están definidas sobre un conjunto abierto $U \in \mathbb{R}^n$.

El punto de partida del análisis es una versión apropiada multivariable de la noción de grado relativo.

El grado relativo (r_i) es exactamente el número de veces que la salida i -ésima tiene que derivarse (evaluada en $t = t^0$) para que aparezca al menos un componente del vector de entrada explícitamente.

Un sistema no lineal multivariable de la forma (1) tiene un vector de grado relativo $\rho = [r_1, \dots, r_m]$ en un punto x^0 si:

1. $L_{g_j} L_f^k h_i(x) = 0$, para todo $1 \leq j \leq m, k < r_i - 1$ y para toda x en una vecindad de x^0 .
2. La siguiente matriz $m \times m$:

$$A(x) = \begin{bmatrix} L_{g_1} L_f^{r_1-1} h_1(x) & \dots & L_{g_m} L_f^{r_1-1} h_1(x) \\ L_{g_1} L_f^{r_2-1} h_2(x) & \dots & L_{g_m} L_f^{r_2-1} h_2(x) \\ \vdots & \ddots & \vdots \\ L_{g_1} L_f^{r_m-1} h_m(x) & \dots & L_{g_m} L_f^{r_m-1} h_m(x) \end{bmatrix}\quad (2)$$

es no singular en x^0 .

Donde $L_f^i h(x) = L(L_f^{i-1} h(x))$, $L_f h(x) = (\partial h(x)/\partial x)f(x)$ es la derivada de Lie de $h(x)$ con respecto $f(x)$. (Khalil, 2002) La derivada de Lie es una derivada direccional con la cual podemos generar espacios de detectabilidad y alcanzabilidad y así transformar el sistema de coordenadas de un sistema no lineal a un sistema de coordenadas locales de interés.

Si estas condiciones se satisfacen podemos definir ρ nuevas coordenadas $z_{i+1} = L_f^i h(x)$, $i = 0, 1, \dots, \rho - 1$. El nuevo sistema

transformado tiene la siguiente estructura (forma canónica):

$$\begin{aligned} \dot{z}_1^i &= z_2^i \\ \dot{z}_2^i &= z_3^i \\ &\dots \\ \dot{z}_{r_i-1}^i &= z_{r_i}^i \\ \dot{z}_{r_i}^i &= b_i(z, \eta) + \sum_{j=1}^m a_{ij}(z, \eta)u_j \\ y_i &= z_1^i \end{aligned} \tag{3}$$

donde

$$a_{ij}(z, \eta) = L_{g_j} L_f^{r_i-1} h_i(\Phi^{-1}(z, \eta)) \quad \text{para } 1 \leq i, j \leq m \tag{4}$$

$$b_i(z, \eta) = L_f^{r_i} h_i(\Phi^{-1}(z, \eta)) \quad \text{para } 1 \leq i \leq m \tag{5}$$

Considere un sistema no lineal que tiene un vector de grado relativo $\{r_1, \dots, r_m\}$ en x^0 y suponga que la suma $r = r_1 + \dots + r_m$ es exactamente igual a n (la dimensión del espacio de estado), en este caso el sistema transformado está caracterizado totalmente por (3) y se considera de grado relativo completo el cual es nuestro caso.

2.1. Seguimiento asintótico de salida

En (Isidori, 1995) se obtiene una salida que, independientemente del estado inicial del sistema, converge asintóticamente a una función de referencia $y_R(t)$.

Considerando un sistema en la forma normal (3) y seleccionando la siguiente acción de control:

$$u = A^{-1}(z) \left(-b(z) + y_R^{(r)} - \sum_{i=1}^r c_{i-1} e^{(i-1)} \right) \tag{6}$$

donde c_i, \dots, c_{i-1} son números reales, $e(t) = y(t) - y_R(t)$, se puede observar que los elementos de la matriz A son los mencionados en la ecuación (4) y a su vez es la matriz $A(x)$ de grado relativo (2) y el vector $b(z)$ está conformado por las derivadas de Lie como se indican en (5), $y_R^{(r)}$ es la r -ésima derivada de la función de referencia.

Por lo tanto, bajo el efecto de una entrada de la forma (6), la salida del sistema sigue a la señal deseada $y_R(t)$ con un error, el cual puede hacerse converger a cero cuando $t \rightarrow \infty$ en forma exponencial arbitrariamente rápido (Isidori, 1995).

3. Redes neuronales recurrentes de alto orden

Considérese una red con n neuronas y m entradas. El estado de cada neurona está gobernado por una ecuación diferencial de la siguiente forma:

$$\dot{x}_i = -\lambda_i x_i + \mu_i \left[\sum_{k=1}^L w_{ik} \prod_{j \in I_k} y_j^{d_j(k)} \right] \tag{7}$$

donde x_i es el estado de la i -ésima neurona; L es el número de conexiones de alto orden; $\{I_1, I_2, \dots, I_L\}$ es una colección de L subconjuntos no ordenados $\{1, 2, \dots, m + n\}$; λ_i, μ_i son coeficientes reales, w_{ik} son los coeficientes reales ajustables de la

red neuronal; $d_j(k)$ son enteros no negativos, e y es un vector construido por las entradas a cada neurona, definido como:

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \\ y_{n+1} \\ \vdots \\ y_{n+m} \end{bmatrix} = \begin{bmatrix} S(x_1) \\ \vdots \\ S(x_n) \\ S(u_1) \\ \vdots \\ S(u_m) \end{bmatrix} \tag{8}$$

donde $u = [u_1 u_2 \dots u_m]^T$ es el vector de entradas externas a la red neuronal. La función $S(\bullet)$ es una sigmoide suave, monótonamente creciente (Rovithakis y Christodoulou, 2000).

Se define el vector de dimensión L , tal que:

$$\zeta = \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \vdots \\ \zeta_L \end{bmatrix} = \begin{bmatrix} \prod_{j \in I_1} y_j^{d_j(1)} \\ \prod_{j \in I_2} y_j^{d_j(2)} \\ \vdots \\ \prod_{j \in I_L} y_j^{d_j(L)} \end{bmatrix} \tag{9}$$

y de esta manera, (7) se expresa como sigue:

$$\dot{x}_i = -\lambda_i x_i + \mu_i \left[\sum_{k=1}^L w_{ik} \zeta_k \right] \tag{10}$$

Más aún, si se define el vector de pesos como $w_i = \mu_i [w_{i1} w_{i2} \dots w_{iL}]^T$, entonces:

$$\dot{x}_i = -\lambda_i x_i + w_i^T \zeta \tag{11}$$

donde $\lambda_i > 0, \forall i = 1, 2, \dots, n$; por lo tanto, el modelo de la RHONN queda de la siguiente forma:

$$\dot{x} = \Lambda x + W^T \zeta \tag{12}$$

donde: $x = [x_1 \dots x_n]^T \in \mathbb{R}^n$; $W = [w_1, w_2, \dots, w_n]^T \in \mathbb{R}^{n \times L}$ y $\Lambda = \text{diag}\{-\lambda_1, -\lambda_2, \dots, -\lambda_n\} \in \mathbb{R}^{n \times n}$ y se supone que $\lambda_i > 0$.

Considérese el problema de aproximación de un sistema dinámico no lineal de la forma:

$$\dot{\chi} = F(\chi, u) \tag{13}$$

donde $\chi \in \mathbb{R}^n$ es el estado del sistema, $u \in \mathbb{R}^m$ es la entrada y $F : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^n$ es un campo vectorial suave definido en un conjunto compacto $\gamma \in \mathbb{R}^{n+m}$. En (Kosmatopoulos et al., 1995) se ha demostrado que el modelo (12) puede aproximar cualquier sistema de la forma (13) con el grado de precisión deseado, siempre y cuando el número de conexiones L y neuronas sea seleccionado adecuadamente (Kosmatopoulos et al., 1997). Suponiendo que el sistema (13) y el modelo (12) son iniciados en el mismo estado $x(0) = \chi(0)$; entonces para cualquier $\varepsilon > 0$ y cualquier $T > 0$ finito, existe un entero L y una matriz $W^* \in \mathbb{R}^{L \times n}$ tal que el estado $x(t)$ de la RHONN modelada en (12) con L conexiones de alto orden y pesos $W = W^*$ satisfacen:

$$\sup_{0 \leq t \leq T} |x(t) - \chi(t)| \leq \varepsilon. \tag{14}$$

3.1. Ley de adaptación de pesos

Los pesos w_i , para $i = 1, 2, \dots, n$, son ajustados de acuerdo a la ley de aprendizaje:

$$\dot{w}_i = -\Gamma_i \zeta e_i \quad (15)$$

donde $e_i := x_i - \chi_i$ es el estado de la red menos el estado a identificar, ζ esta definida como en (9), la ganancia adaptativa Γ_i es una matriz definida positiva $L \times L$. En el caso especial que $\Gamma_i = \gamma_i I$, donde $\gamma_i > 0$ es un escalar, entonces Γ_i en (15) puede ser remplazada por γ_i (Rovithakis y Christodoulou, 2000).

4. Control neuro-robusto del robot móvil diferencial

4.1. Modelo dinámico del robot móvil diferencial.

El modelo dinámico es derivado de la aproximación Newton-Euler y fue tomado de (Dhaouadi y Hatan, 2013)

$$\begin{bmatrix} \dot{\omega}_R \\ \dot{\omega}_L \end{bmatrix} = \begin{bmatrix} \frac{m_1 V \omega_L (\omega_L - \omega_R)}{m_1^2 - m_2^2} + \frac{m_2 V \omega_R (\omega_L - \omega_R)}{m_1^2 - m_2^2} \\ -\frac{m_2 V \omega_L (\omega_L - \omega_R)}{m_1^2 - m_2^2} - \frac{m_1 V \omega_R (\omega_L - \omega_R)}{m_1^2 - m_2^2} \end{bmatrix} + \begin{bmatrix} \frac{m_1}{R(m_1^2 - m_2^2)} \\ -\frac{m_2}{R(m_1^2 - m_2^2)} \end{bmatrix} \tau_R + \begin{bmatrix} -\frac{m_2}{R(m_1^2 - m_2^2)} \\ \frac{m_1}{R(m_1^2 - m_2^2)} \end{bmatrix} \tau_L \quad (16)$$

$$y_1 = \omega_R \quad y_2 = \omega_L$$

donde

$$m_1 = \frac{R(Md^2 + J)}{4L^2} + \frac{MR}{4} \quad m_2 = -\frac{R(Md^2 + J)}{4L^2} + \frac{MR}{4} \quad (17)$$

$$V = \frac{MdR^2}{4L^2}$$

La ecuación (16) muestra el modelo dinámico a utilizar en este trabajo. En la Figura 1 podemos observar el diagrama del robot móvil diferencial.

donde

- M Masa total del robot móvil diferencial.
- R Radio de la rueda.
- d distancia del centro del eje de ruedas al centro de masa.
- J Momento de inercia del robot móvil diferencial sobre el eje vertical z a través del centro de masa.
- L distancia del centro del robot al eje de la rueda.
- ω_R y ω_L velocidades angulares de las ruedas derecha e izquierda respectivamente.
- ω_{Rd} y ω_{Ld} velocidades angulares deseadas de las ruedas derecha e izquierda respectivamente.

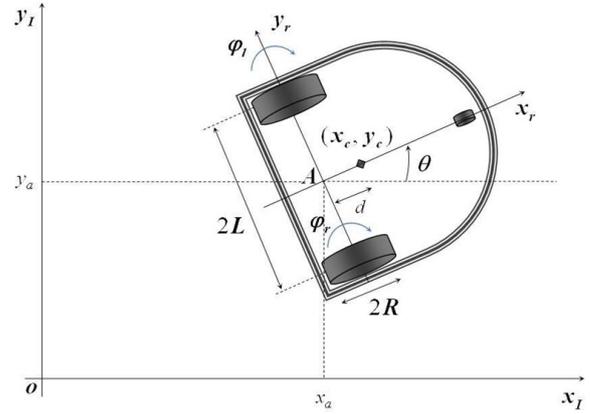


Figura 1: Robot móvil diferencial (Dhaouadi y Hatan, 2013).

4.2. Diseño del controlador no lineal

Aplicando la teoría presentada en la sección anterior para obtener el sistema control linealizante obtenemos la matriz de grado relativo que está dada por

$$A(x) = \begin{bmatrix} \frac{m_1}{R(m_1^2 - m_2^2)} & -\frac{m_2}{R(m_1^2 - m_2^2)} \\ -\frac{m_2}{R(m_1^2 - m_2^2)} & \frac{m_1}{R(m_1^2 - m_2^2)} \end{bmatrix} \quad (18)$$

rango $A(x) = 2$ y $\det \neq 0$. Los elementos del vector $b(x)$ corresponden a la derivada de Lie como se indica en la ecuación (5).

$$b(x) = \begin{bmatrix} \frac{m_1 V \omega_L (\omega_L - \omega_R)}{m_1^2 - m_2^2} + \frac{m_2 V \omega_R (\omega_L - \omega_R)}{m_1^2 - m_2^2} \\ -\frac{m_2 V \omega_L (\omega_L - \omega_R)}{m_1^2 - m_2^2} - \frac{m_1 V \omega_R (\omega_L - \omega_R)}{m_1^2 - m_2^2} \end{bmatrix} \quad (19)$$

La transformación de coordenadas queda de la siguiente manera

$$\begin{aligned} z_1^1 &= h_1(x) \\ z_2^1 &= b_1 + a_{11}u_1 + a_{12}u_2 \\ z_1^2 &= h_2(x) \\ z_2^2 &= b_2 + a_{21}u_1 + a_{22}u_2 \end{aligned} \quad (20)$$

donde $z_1^1 = \omega_R$ y $z_2^1 = \omega_L$, b_1 y b_2 son los elementos de vector b , ecuación (19) y a_{11} , a_{12} , a_{21} y a_{22} son los elementos de la matriz A ecuación (18) así el sistema transformado es:

$$\begin{aligned} \dot{z}_1^1 &= \underbrace{\frac{m_1 V z_1^2 (z_2^1 - z_1^1)}{m_1^2 - m_2^2} + \frac{m_2 V z_1^1 (z_2^1 - z_1^1)}{m_1^2 - m_2^2}}_{b_1} \\ &+ \underbrace{\frac{m_1}{R(m_1^2 - m_2^2)}}_{a_{11}} u_1 - \underbrace{\frac{m_2}{R(m_1^2 - m_2^2)}}_{a_{12}} u_2 \end{aligned} \quad (21)$$

$$\begin{aligned} \dot{z}_2^1 &= -\underbrace{\frac{m_2 V z_1^2 (z_2^1 - z_1^1)}{m_1^2 - m_2^2} - \frac{m_1 V z_1^1 (z_2^1 - z_1^1)}{m_1^2 - m_2^2}}_{b_2} \\ &- \underbrace{\frac{m_2}{R(m_1^2 - m_2^2)}}_{a_{21}} u_1 + \underbrace{\frac{m_1}{R(m_1^2 - m_2^2)}}_{a_{22}} u_2 \end{aligned} \quad (22)$$

La ley de control para el seguimiento asintótico de trayectorias entonces es de la forma (6) donde $y_R^{(r)}$ es la r -ésima derivada de la velocidad angular de referencia de cada rueda.

Es obvio que las funciones no lineales señaladas como b_1 y b_2 en (21) y (22) son inciertas y por tanto deben estimarse para que el controlador pueda compensarlas. Para lo anterior se propone una RHONN para dichas funciones inciertas y entonces poderlas compensar así, por lo tanto el sistema tiene la siguiente forma:

$$\begin{aligned} \dot{z}_1^1 &= \overbrace{b_1(z)}^{\text{Estimar}} + a_{11}(z)u_1 + a_{12}(z)u_2 \\ \dot{z}_1^2 &= \overbrace{b_2(z)}^{\text{Estimar}} + a_{21}(z)u_1 + a_{22}(z)u_2 \end{aligned} \quad (23)$$

cómo se puede observar en las ecuaciones (21) y (22) los términos a_{11} , a_{12} , a_{21} y a_{22} no dependen de los estados y son constantes, además, noten que estos términos dependen de los parámetros del sistema, sin embargo es suficiente con conocer el signo de estos términos y no necesariamente el valor exacto tampoco es necesario estimarlos, ya que por la estructura del controlador es suficiente con conocer su signo. Entonces la matriz A expresada en la ecuación (4) nos queda de la siguiente manera tomando en cuenta lo mencionado anteriormente

$$A = \begin{bmatrix} \delta_1 & -\delta_2 \\ -\delta_3 & \delta_4 \end{bmatrix} \quad (24)$$

donde δ_1 , δ_2 , δ_3 y δ_4 son los parámetros de la matriz A desconocidos. La ley de control con parámetros estimados mediante la RHONN es de la siguiente forma:

$$u_1 = -\frac{\delta_2(\hat{b}_2 - \omega_{Ld}^{(r)} + e\omega_L k_2)}{\delta_1\delta_4 - \delta_2\delta_3} - \frac{\delta_4(\hat{b}_1 - \omega_{Rd}^{(r)} + e\omega_R k_1)}{\delta_1\delta_4 - \delta_2\delta_3} \quad (25)$$

$$u_2 = -\frac{\delta_1(\hat{b}_2 - \omega_{Ld}^{(r)} + e\omega_L k_2)}{\delta_1\delta_4 - \delta_2\delta_3} - \frac{\delta_3(\hat{b}_1 - \omega_{Rd}^{(r)} + e\omega_R k_1)}{\delta_1\delta_4 - \delta_2\delta_3} \quad (26)$$

donde $e\omega_L = \omega_L - \omega_{Ld}$ y $e\omega_R = \omega_R - \omega_{Rd}$, $\omega_{Rd}^{(r)}$ y $\omega_{Ld}^{(r)}$ es la r -ésima derivada de la velocidad angular derecha e izquierda respectivamente que en nuestro caso dado que el grado relativo de cada salida es igual a uno es la primer derivada, \hat{b}_1 y \hat{b}_2 son los parámetros estimados mediante la RHONN los cuales se obtienen en la sección siguiente, k_1 y k_2 son las ganancias las cuales son seleccionadas como en el control proporcional de lazo cerrado. El proceso anterior fue solamente para un agente móvil, para formar el sistema líder-seguidor es necesario duplicar el procedimiento para el agente seguidor.

Al estimar el parámetro b eliminamos algunas restricciones del método de linealización exacta y garantizamos cierta robustez debido a que esta función se adaptará mediante la RHONN si algún parámetro cambia de valor. Note que la función b contiene los parámetros que pueden variar y las funciones del modelo, y para generar la acción de control es suficiente con estimar el comportamiento dinámico de la función b y no cada parámetro del modelo.

4.3. RHONN

Modelo de la Red neuronal Recurrente de Alto Orden a implementar.

$$\begin{aligned} \dot{z}_1^1 &= -\lambda_1 \hat{z}_1^1 + w_1^T \zeta_I(\hat{z}, u) \\ \dot{z}_1^2 &= -\lambda_2 \hat{z}_1^2 + w_2^T \zeta_I(\hat{z}, u) \\ \dot{z}_2^1 &= -\lambda_3 \hat{z}_2^1 + w_3^T \zeta_I(\hat{z}, u) \\ \dot{z}_2^2 &= -\lambda_4 \hat{z}_2^2 + w_4^T \zeta_I(\hat{z}, u) \end{aligned} \quad (27)$$

usaremos una RHONN de 4 estados para identificar los estados del sistema transformado (21) y (22) del robot líder (z_1^1 y z_1^2) y del seguidor (z_2^1 y z_2^2). Los parámetros \hat{b}_{11} , \hat{b}_{12} son derivados de la diferencia de \hat{z}_1^1 y \hat{z}_1^2 y la matriz A en (24) y de la misma forma para el seguidor. La ley de adaptación de pesos para cada estado utilizada esta dada por (15).

5. Simulación numérica

Tabla 1: Valores de los parámetros para la simulación 1

Parámetro	Valor	
	Normales	Variación
M	2,2 kg	4,4kg
d	0,04 m	0,04 m
R	0,03 m	0,03 m
L	0,0975 m	0,0975 m
J	$4,1e^{-6}$ kg-m	$4,1e^{-6}$ kg-m
m_1	0.0172	0.0344
m_2	0,0158	0,0316
V	$5,2071e^{-4}$	0,0010

Se utilizo MATLAB-Simulink para simular el sistema multi-robot y la ley de control generada, los resultados se presentan gráficamente. Para la simulación consideramos como trayectoria deseada una curva de Lissajous en el plano, la cual fue necesario parametrizar en velocidades angulares ω_R y ω_L correspondientes de ambas ruedas con ayuda del modelo cinemático mencionado en (Dhaouadi y Hatan, 2013). La trayectoria se simuló durante 60 segundos, en el intervalo de 10 – 20 segundos el robot recibe peso extra, una variación del 100 % como se puede observar en la Tabla 1. En el intervalo de 30 – 40 segundos el sistema recibe una perturbación externa (un escalón de amplitud 7.5 rad/seg) que equivale a acelerar la rueda derecha solamente.

En la Figura 2 podemos observar las velocidades deseadas w_R y w_L en color azul y rojo, al igual que w_R y w_L del líder (amarillo y morado) y w_R y w_L del seguidor (verde y azul cielo). Se observa que ambos agentes siguen la velocidad deseada, el error del agente seguidor es mayor al del líder ya que toma como referencia las velocidades angulares del líder. En el intervalo de variación paramétrica el error de seguimiento es pequeño al igual que en el periodo de perturbación externa.

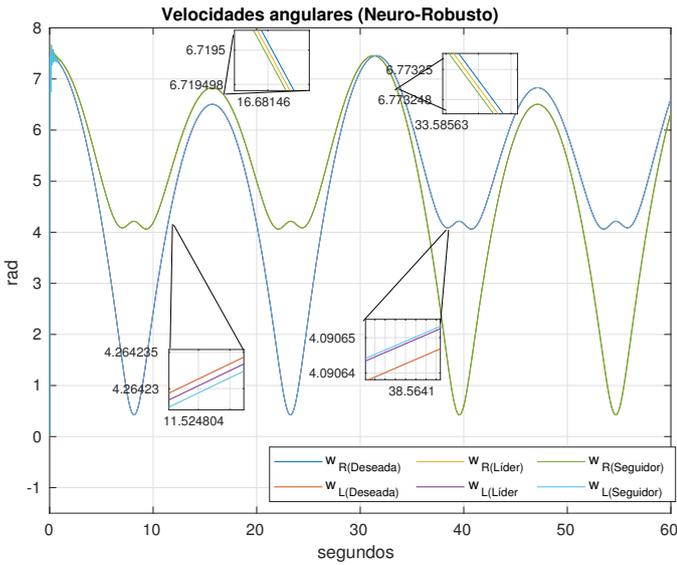


Figura 2: Velocidades angulares obtenidas.

En la Figura 3 observamos los parámetros b estimados mediante la RHONN.

En la Figura 4 tenemos los dos pares de torques que genera el controlador, ante la variación paramétrica la acción de control aumenta ya que los motores necesitan mas fuerza debido al aumento de masa. En el periodo de perturbación externa el torque es negativo y de mayor amplitud esto compensa el aumento en la velocidad.

En las Figura 5 se muestran los estimados de los estados de las trasformaciones los cuales nos ayudan para deducir los parámetros b_{ij} .

La Figura 6 muestra la trayectoria en el plano cartesiano y se puede observar el error del seguidor con respecto de la referencia deseada, el cual es mayor que el del líder esto es por que el seguidor tiene como referencia la trayectoria recorrida del líder y carga con el error de este. Los asteriscos nos marcan los periodos de tiempo correspondientes a la variación paramétrica y la perturbación externa.

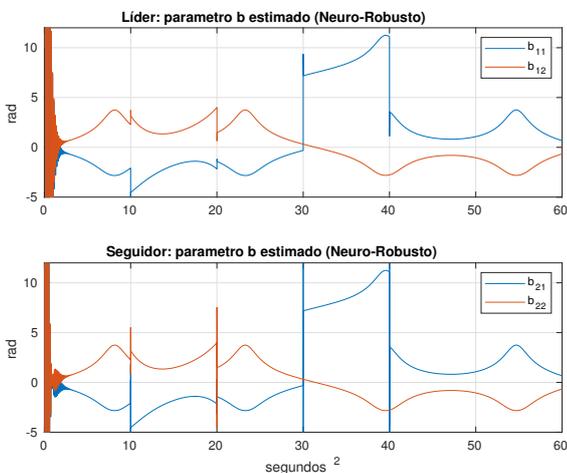


Figura 3: Parámetros b estimados.

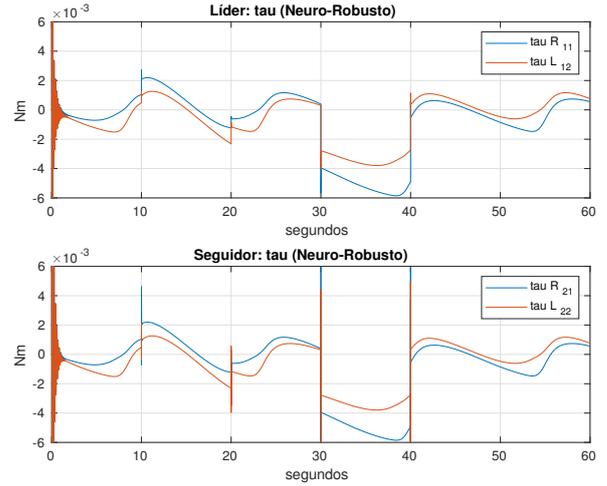


Figura 4: Leyes de control (torques).

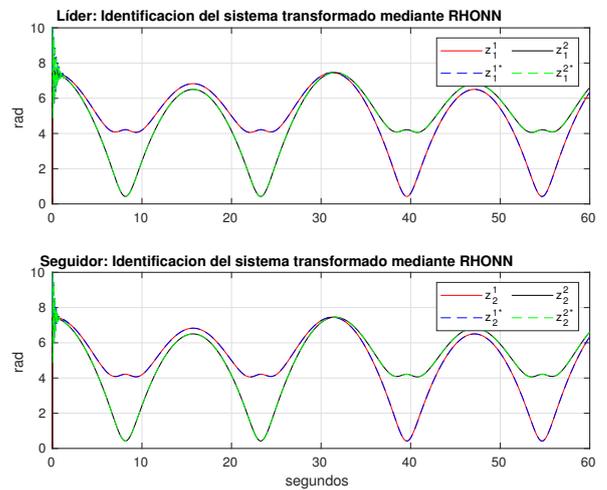


Figura 5: Estimado de las dinámicas de la transformación.

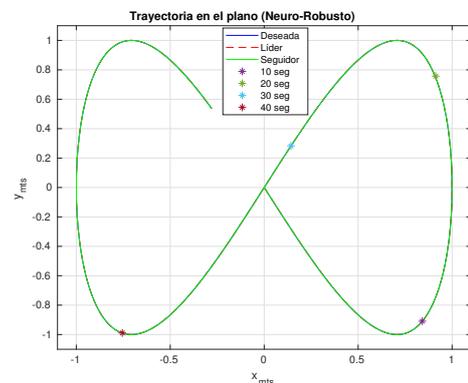


Figura 6: Trayectorias en el plano cartesiano.

6. Análisis comparativo

El controlador neuro-robusto obtenido es comparado con un controlador PID. Para esta comparación se utilizó el mismo

modelo dinámico junto con los parámetros mencionados anteriormente y se sometió a las mismas pruebas de variación paramétrica y perturbación externa obteniendo un comportamiento menos favorable. En las Figuras 7 y 8 observamos el error de las velocidades obtenidas del controlador neuro-robusto y del controlador PID. El error es calculado con la diferencia de la velocidad obtenida y la velocidad deseada, en las gráficas se puede observar un mejor comportamiento del controlador neuro-robusto, cabe resaltar que como se mencionó en la introducción y como se menciona en (Majid *et al.*, 2016) una limitación del controlador PID es que las ganancias dependen de la precisión del modelo, el modelo al recibir algún cambio paramétrico tendría que ser sintonizado nuevamente para obtener un comportamiento aceptable que a diferencia del controlador propuesto no es necesario ya que este lo compensa, otra ventaja del controlador neuro-robusto es que requiere poca información del modelo.

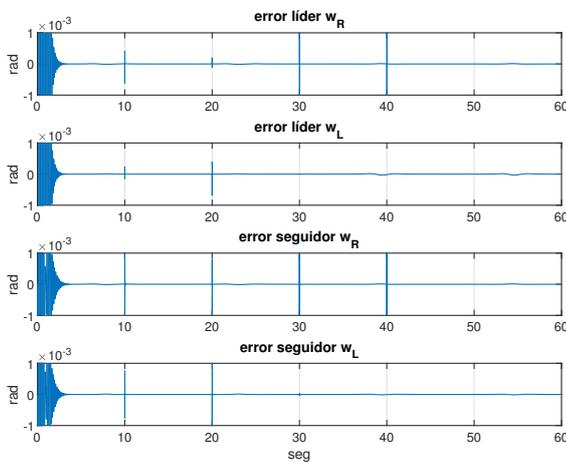


Figura 7: Error de velocidades (Neuro-Robusto).

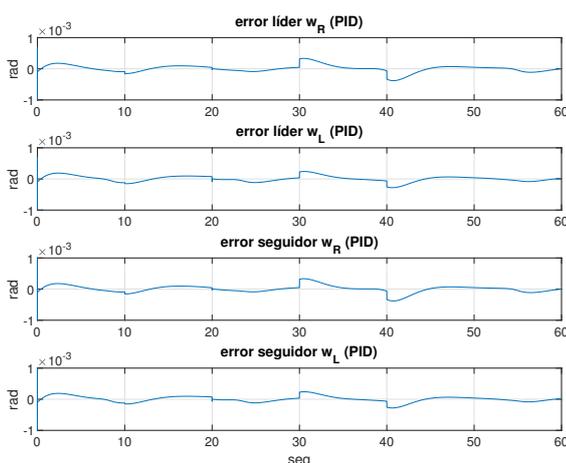


Figura 8: Error de velocidades (PID).

7. Conclusión

El diseño del controlador consta de dos partes, un controlador linealizante y una red neuronal, con la transformación del

sistema se aglutinan todos los parámetros del sistema en una sola función incierta no lineal. Esta función entonces es estimada mediante una red neuronal con lo que no es necesario estimar cada parámetro del sistema y tampoco las incertidumbres por alguna dinámica no modeladas. La técnica aglutina todas estas incertidumbres cuyo estimado es enviado al controlador que las compensa. Por lo tanto, el esquema completo es robusto y solo requiere las mediciones de la salida y un estimado o valor nominal de algunos parámetros. El controlador neuro-robusto diseñado ha mostrado un comportamiento congruente con lo esperado referente a las variaciones paramétricas y perturbaciones añadidas logrando la robustez del sistema y eliminando la necesidad de conocer los parámetros del sistema. Lo cual se ilustra en una simulación. Por otro lado, este mismo esquema puede ser aplicado en otro tipo de sistemas o modelos y puede extenderse al caso de seguimiento de trayectorias con medidas de posición y no solo de velocidad.

Referencias

- Dhaouadi, R. y Hatan, A. A. (2013). Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies: A unified framework. *Advances in Robotics and Automation*, 2:107.
- Femat, R., Alvarez-Ramirez, J., Castillo-Toledo, B., y Gonzalez, J. (1999). On robust suppression in a class of nondriven oscillators: application to the chua's circuit. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 46(9):1150–1152.
- Femat, R. y Solis-Perales, G. (2008). *Robust Synchronization Via Geometrical Control: A General Framework*, pp. 99–137. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Hassan, N. y Saleem, A. (2022). Neural network-based adaptive controller for trajectory tracking of wheeled mobile robots. *IEEE Access*, 10:13582–13597.
- Hikrobot, C. (2022). Latent mobile robot.
- Hopfield, J. J. (1984). Neurons with graded responses have collective computational properties like those of two state neurons. *Proc. Nat. Acad. Sci.*, 81:3088–3092.
- iRobot Corporation (2022). Roomba.
- Isidori, A. (1995). *Nonlinear Control System*. Springer-Verlag, 3 edición.
- Khalil, H. (2002). *Nonlinear Systems*. Pearson Education. Prentice-Hall, Upper Saddle River, NJ, 3 edición.
- Kosmatopoulos, E., Polycarpou, M., Christodoulou, M., e Ioannou, P. (1995). High-order neural network structures for identification of dynamical systems. *IEEE Transactions on Neural Networks*, 6(2):422–431.
- Kosmatopoulos, E., Chistodoulou, M., e Ioannou, P. (1997). Dynamical neural networks that ensure identification error convergence. *Neural Networks*, 10(2):299–314.
- KUKA (2022). Intelligent automated guided vehicle systems revolutionize the future of production.
- Lian, K.-Y., Hsu, W.-H., y Tsai, T.-S. (2019). Leader-follower mobile robots control based on light source detection. *IEEE Sensors Journal*, 19(23):11142–11150.
- Majid, N., Mohamed, Z., y Basri, A. (2016). Velocity control of a unicycle type of mobile robot using optimal pid controller. *Jurnal Teknologi*, 78.
- Meng, J., Liu, A., Yang, Y., Wu, Z., y Xu, Q. (2018). Two-wheeled robot platform based on pid control. En *2018 5th International Conference on Information Science and Control Engineering (ICISCE)*, pp. 1011–1014.
- Rabbani, M. J. y Memon, A. Y. (2021). Trajectory tracking and stabilization of nonholonomic wheeled mobile robot using recursive integral backstepping control. *Electronics*, 10(16).
- Ricalde, L. (2005). *Inverse optimal adaptive recurrent neural control with constrained inputs*. Tesis doctoral. Cinvestav.
- Rovithakis, G. A. y Christodoulou, M. A. (2000). *Adaptive control with recurrent high-order neural networks*. Springer Verlag.
- Soorki, M., Talebi, H., y Nikravesh, S. (2011). Robust leader-following formation control of multiple mobile robots using lyapunov redesign. En *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*, pp. 277–282.
- Wang, Y., Cheng, L., Hou, Z., Yu, J., y Tan, M. (2016). Optimal formation of multi-robot systems based on recurrent neural network. *IEEE Transaction on Neuronal Networks and Learning Systems*, 27:322–333.