



Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería

www.elsevier.es/rimni



Algoritmo compensador neuronal discreto de dinámica en robots móviles usando filtro de Kalman extendido

F.G. Rossomando*, C. Soria y R. Carelli

Instituto de Automática, Facultad de Ingeniería, Universidad Nacional de San Juan, Av. San Martín 1112 Oeste, 5400 San Juan, Argentina

INFORMACIÓN DEL ARTÍCULO

Historia del artículo:

Recibido el 10 de junio de 2010

Aceptado el 28 de noviembre de 2011

On-line el 5 de marzo de 2013

Palabras clave:

Robots móviles

Redes neuronales

Filtro de Kalman extendido

Control no lineal

R E S U M E N

Este artículo presenta el diseño de un algoritmo basado en redes neuronales en tiempo discreto para su aplicación en robótica móvil. También se muestran las condiciones de estabilidad y una evaluación de los resultados.

El robot móvil en el cual se aplicó el algoritmo neural posee 2 controladores en cascada, uno para la cinemática y otro para la dinámica; ambos controladores están basados en la linealización por realimentación. El controlador de la dinámica solo posee la información de la dinámica nominal (parámetros). La red neuronal de compensación se adapta para reducir las perturbaciones ocasionadas por las variaciones en la dinámica y las incertidumbres existentes en el modelo, y esas diferencias en la dinámica entre el modelo nominal y el real son aprendidas por una red neuronal RBF (funciones de base radial) usando el filtro de Kalman extendido para el ajuste de los pesos de salida de las funciones de base radial. El algoritmo de compensación neuronal es eficiente, ya que el costo computacional es menor que el necesario para aprender la totalidad de la dinámica y al mismo tiempo posee la robustez que podría aprender la totalidad de la dinámica en caso de fallo del controlador dinámico. En este trabajo se muestra un análisis de estabilidad del algoritmo neuronal adaptable, y además se comprueba que los errores de control están acotados en función del error de aproximación de la red neuronal RBF. Se muestran resultados de experimentación sobre un robot móvil que prueban la viabilidad práctica y el rendimiento para el control de los mismos.

© 2010 CIMNE (Universitat Politècnica de Catalunya). Publicado por Elsevier España, S.L. Todos los derechos reservados.

Discrete neural compensator algorithm of dynamic in mobile robots using extended Kalman filter

A B S T R A C T

This paper presents the design of an algorithm based on neural networks in discrete time for its application in mobile robots. In addition, the system stability is analyzed and an evaluation of the experimental results is shown.

The mobile robot has two controllers, one addressed for the kinematics and the other one designed for the dynamics. Both controllers are based on the feedback linearization. The controller of the dynamics only has information of the nominal dynamics (parameters). The neural algorithm of compensation adapts its behaviour to reduce the perturbations caused by the variations in the dynamics and the model uncertainties. Thus, the differences in the dynamics between the nominal model and the real one are learned by a neural network RBF (radial basis functions) where the output weights are set using the extended Kalman filter. The neural compensation algorithm is efficient, since the consumed processing time is lower than the one required to learning the totality of the dynamics. In addition, the proposed algorithm is robust with respect to failures of the dynamic controller. In this work, a stability analysis of the adaptable neural algorithm is shown and it is demonstrated that the control errors are bounded depending on the error of approximation of the neural network RBF. Finally, the results of experiments performed by using a mobile robot are shown to test the viability in practice and the performance for the control of robots.

© 2010 CIMNE (Universitat Politècnica de Catalunya). Published by Elsevier España, S.L. All rights reserved.

Keywords:

Mobile robots

Neural networks

Extended Kalman filter

Nonlinear control

* Autor para correspondencia.

Correo electrónico: frosoma@inaut.unsj.edu.ar (F.G. Rossomando).

1. Introducción

Las redes neuronales (RN) poseen un amplio campo de aplicación, que incluye la clasificación, la identificación y el control de sistemas. En este trabajo se presenta el diseño de un algoritmo basado en redes neuronales RBF (funciones de base radial) para su aplicación en robótica móvil, cuyo objetivo es mejorar el control de trayectorias en tiempo real, pero debido a deslizamientos, perturbaciones, ruidos y errores de sensado y medición (odometría) resulta muy difícil reducir los errores entre la posición deseada y la posición real del robot. Asimismo, lograr controlar eficazmente un robot móvil para realizar el seguimiento preciso de una trayectoria deseada es aún un problema abierto en robótica.

Se han publicado varios estudios en relación con el diseño de controladores para el seguimiento de trayectoria de robots móviles. La mayoría de los controladores diseñados hasta el momento se basan únicamente en la cinemática, como los controladores presentados en Wu et al. [1], Kühne et al. [2] y en Gracia y Tornero [3].

Sin embargo, para realizar tareas que requieren movimientos de alta velocidad o transporte de cargas pesadas es importante tener en cuenta, además de la cinemática, la dinámica del robot. Más aún: el controlador debería contemplar las incertidumbres o los eventuales cambios en esa dinámica sin degradar el desempeño. A modo de ejemplo, en el caso de transporte de carga, las características dinámicas como la masa, el centro de masa y la inercia varían con la carga. Para mantener un buen desempeño, el controlador debe ser capaz de adaptarse a este tipo de cambios. Esta capacidad de adaptación también es importante debido a la dificultad de modelar el sistema con exactitud, aun cuando no se verifiquen cambios dinámicos de tarea a tarea.

Otros controladores de seguimiento que contemplan la incertidumbre en la dinámica del robot son desarrollados por Liu et al. [4] y Dong y Guo [5], cuyo desempeño es mostrado a través de simulaciones. Otros trabajos similares se pueden ver en Liyong y Wei [6] y en Chaitanya y Sarkar [7]. Das y Kar [8] presentan un controlador de lógica difusa adaptable donde la incertidumbre del sistema, que incluye variación de parámetros del robot móvil y no linealidades desconocidas, se estima por un sistema de lógica difusa cuyos parámetros son sintonizados en línea.

En Fukao et al. [9] se propone el diseño de un controlador de seguimiento de trayectoria adaptativo para generar pares motores sobre la base de un modelo dinámico cuyos parámetros son desconocidos.

En Velagic et al. [10] se presenta el diseño de un controlador adaptable directo utilizando RN recurrentes. La eficiencia de la técnica de control es investigada sobre el modelo cinemático del robot móvil.

Otros tipos de controladores de seguimiento de trayectoria que consideran la incertidumbre en la dinámica del robot son los desarrollados en Dong y Guo [5] y en Dong y Huo [11], los cuales presentan resultados de simulación.

Martins et al. [12] y Rossomando et al. [13] utilizan un controlador de dinámica inversa adaptable y neural adaptable respectivamente que identifica toda la dinámica real del robot; estos trabajos presentan todo el diseño en el dominio continuo.

Bugeja et al. [14] presentan el uso de una red neuronal RBF en tiempo discreto para las estimaciones de las incertidumbres en el diseño del control, en la que se optó por emplear técnicas de control adaptable estocástico, más concretamente, el principio de doble control, que fueron introducidas por Fel'dbaum [15].

Para lograr ese objetivo la red RBF se entrena por un algoritmo basado en el filtro de Kalman extendido (EKF). Esta red aproxima la dinámica directa completa de los robots móviles. Con esta estimación de la dinámica se obtiene un funcional de coste para el

desarrollo del controlador. Los autores solo muestran resultados de simulación y no presentan un análisis de estabilidad.

Alanís et al. [16] presentan un controlador *backstepping* basado en RN; el controlador está diseñado en tiempo discreto sobre la dinámica completa, solo muestra un ejemplo de simulación y no es aplicado en robótica móvil. En este trabajo se realiza el ajuste de los pesos de la RN del controlador a través de un filtro de Kalman extendido.

Kim et al. [17] han propuesto un controlador adaptable robusto para un robot móvil dividido en 2 partes. El primero se basa en la cinemática de robots y se encarga de generar las referencias para la segunda, que compensa la dinámica del modelo. Sin embargo, los parámetros adaptados no son los parámetros reales del robot, y no se presentan resultados experimentales. Además, las acciones de control se dan en términos de pares motores, aun cuando usualmente los robots comerciales aceptan comandos de velocidad.

En el trabajo de De la Cruz y Carelli [18] se presenta una parametrización lineal de un robot móvil unicycle y el diseño de un controlador de seguimiento de trayectoria que se basa en el modelo completo conocido. Una de las ventajas de su controlador es que sus parámetros están directamente relacionados con los parámetros del robot. Sin embargo, si los parámetros no son correctamente identificados o cambian con el tiempo, debido por ejemplo a la variación de la carga, el desempeño del controlador resulta afectado.

La contribución de este trabajo es el diseño de un controlador de seguimiento de trayectoria adaptable basado en la dinámica nominal del robot y en un algoritmo compensador neuronal. Todo el sistema de control está diseñado en 2 partes: una incluye un controlador cinemático y otra un controlador dinámico con su compensador neuronal. Esto presenta como ventaja, respecto de trabajos previos, que la compensación se hace solo sobre la parte incierta de la dinámica. El método presentado aquí muestra algunas ventajas: la primera es la aplicación a un sistema no lineal *multiple input-multiple output* (MIMO) en tiempo discreto; la segunda es la garantía de que el error está acotado en presencia de perturbaciones, y la tercera es que el diseño puede ser aplicado a diferentes sistemas. Finalmente, este trabajo propone el uso de una red RBF entrenada por un EKF que ajusta los pesos de salida de las RBF, con el fin de aproximar la ley de control de compensación. También se incluye un análisis de los errores de control de trayectoria para el sistema robótico MIMO no lineal en función de los errores de aproximación de la red neural.

Este trabajo está organizado de la siguiente manera: la sección 2 presenta un panorama general del sistema y muestra la representación matemática de todo el modelo de unicycle del robot. La cinemática y la dinámica de los controladores y de la red de compensación RBF se examinan, respectivamente, en las secciones 3 y 4, así como los correspondientes análisis de errores. En la sección 5 se presentan algunos resultados de experimentación que muestran el rendimiento del compensador adaptable. Por último, en la sección 6 se presentan las conclusiones.

2. Modelo del robot

2.1. Visión general del sistema

En esta sección se describe el modelo del robot móvil tipo unicycle como el que se muestra en la figura 1, donde se indican los parámetros y las variables de interés. Las variables v y ω representan las velocidades lineal y angular, respectivamente, desarrolladas por el robot, G es el centro de masa del robot, c es la posición de la rueda castor, E es la ubicación de la herramienta, h

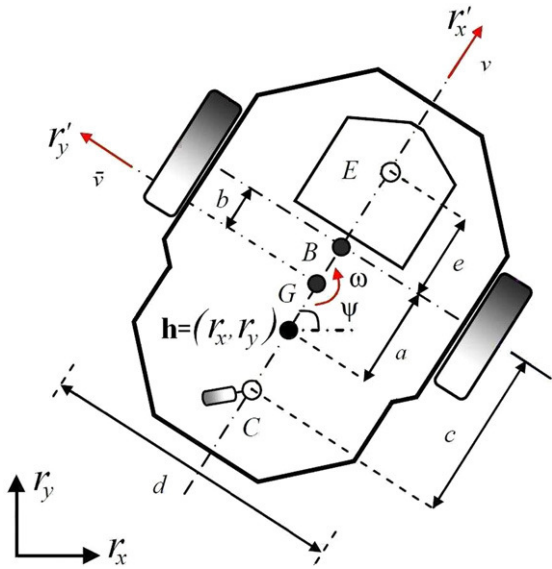


Figura 1. Parámetros del robot móvil uniciclo.

es el punto de interés con coordenadas r_x, r_y en el plano XY, ψ es la orientación del robot, a es la distancia entre el punto de interés y el punto central del eje virtual vinculado a las ruedas de tracción. La representación matemática del modelo completo [18] está dada por:

Modelo cinemático

$$\begin{bmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \psi & -a \sin \psi \\ \sin \psi & a \cos \psi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} + \begin{bmatrix} \delta_{rx} \\ \delta_{ry} \\ 0 \end{bmatrix} \quad (1)$$

Modelo dinámico

$$\begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} \frac{\vartheta_3}{\vartheta_1} \omega^2 - \frac{\vartheta_4}{\vartheta_1} v \\ -\frac{\vartheta_5}{\vartheta_2} v \omega - \frac{\vartheta_6}{\vartheta_2} \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{\vartheta_1} & 0 \\ 0 & \frac{1}{\vartheta_2} \end{bmatrix} \begin{bmatrix} v_{ref} \\ \omega_{ref} \end{bmatrix} + \begin{bmatrix} \delta_v \\ \delta_\omega \end{bmatrix} \quad (2)$$

Los parámetros identificados $\vartheta = [\vartheta_1 \ \vartheta_2 \ \vartheta_3 \ \vartheta_4 \ \vartheta_5 \ \vartheta_6]^T$ del modelo dinámico del robot móvil están descritos por las siguientes relaciones (validadas en [18]):

$$\vartheta_1 = \left(\frac{\left(\frac{R_a}{k_a} (mR_t r + 2I_e) + 2rk_{DT} \right)}{2rk_{PT}} \right)$$

$$\vartheta_2 = \left(\frac{\left(\frac{R_a}{k_a} (I_e d^2 + 2R_t r (I_z + mb^2)) + 2rdk_{DR} \right)}{2rdk_{PR}} \right)$$

$$\vartheta_3 = \left(\frac{\frac{R_a}{k_a} mbR_t}{2k_{PT}} \right); \vartheta_4 = \left(\frac{\frac{R_a}{k_a} \left(\frac{k_a k_b}{R_a} + B_e \right)}{rk_{PT}} + 1 \right)$$

$$\vartheta_5 = \left(\frac{\frac{R_a}{k_a} mbR_t}{dk_{PR}} \right); \vartheta_6 = \left(\frac{\frac{R_a}{k_a} \left(\frac{k_a k_b}{R_a} + B_e \right) d}{2rk_{PR}} + 1 \right)$$

En estas relaciones m es la masa del robot; r es el radio de las ruedas izquierda y derecha; k_b es igual a la constante electromotriz multiplicada por la constante de reducción; R_a es la constante de

resistencia eléctrica; k_a es la constante de torque multiplicada por la constante de reducción; k_{PR} , k_{PT} , y k_{DT} son constantes positivas; I_e y B_e son el momento de inercia y el coeficiente de fricción viscosa de la combinación rotor del motor, caja de reducción y rueda, y R_t es el radio nominal del neumático.

El valor de los parámetros identificados de los robots móviles está indicado en la tabla 1.

El vector de incertidumbres asociados al robot móvil es, $\delta = [\delta_{rx} \ \delta_{ry} \ 0 \ \delta_v \ \delta_\omega]^T$, donde las variables δ_{rx} y δ_{ry} son funciones de la velocidad de deslizamiento y de la orientación del robot, δ_v y δ_ω son funciones de los parámetros físicos como la masa, la inercia, diámetros de las ruedas, del motor y parámetros de los servos, fuerzas aplicadas sobre las ruedas, y otras variables. El vector de incertidumbres es considerado como una perturbación al sistema.

El modelo del robot presentado en (1) y (2) se divide en una parte cinemática y una dinámica, respectivamente, tal como se muestra en la figura 2. Por lo tanto, se aplican 2 controladores, basados en la linealización por realimentación, para la cinemática y la dinámica del modelo del robot, respectivamente.

El modelo del robot se discretiza, a los fines de su control digital, con un intervalo de muestreo $T_0 = 0,1$ s. Este valor es el mismo que usa el robot internamente para el muestro de las señales de velocidad y posición. El índice k es el índice de tiempo discreto. El resultado de la discretización directa del modelo del robot resulta:

Modelo cinemático discretizado

$$\begin{bmatrix} r_x(k+1) \\ r_y(k+1) \\ \psi(k+1) \end{bmatrix} = T_0 \begin{bmatrix} \cos \psi(k) & -a \sin \psi(k) \\ \sin \psi(k) & a \cos \psi(k) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(k) \\ \omega(k) \end{bmatrix} + \begin{bmatrix} r_x(k) \\ r_y(k) \\ \psi(k) \end{bmatrix} + \begin{bmatrix} \delta_{rx} \\ \delta_{ry} \\ 0 \end{bmatrix} \quad (4)$$

Modelo dinámico discretizado

$$\begin{bmatrix} v(k+1) \\ \omega(k+1) \end{bmatrix} = \begin{bmatrix} \frac{\Omega_3}{\Omega_1} \omega^2(k) + \frac{\Omega_4}{\Omega_1} v(k) \\ -\frac{\Omega_5}{\Omega_2} v(k) \omega(k) + \frac{\Omega_6}{\Omega_2} \omega(k) \end{bmatrix} + \begin{bmatrix} \frac{1}{\Omega_1} & 0 \\ 0 & \frac{1}{\Omega_2} \end{bmatrix} \begin{bmatrix} v_{ref}(k) \\ \omega_{ref}(k) \end{bmatrix} + \begin{bmatrix} \delta_v \\ \delta_\omega \end{bmatrix} \quad (5)$$

Donde:

$$\frac{\Omega_3}{\Omega_1} = \left(\frac{\vartheta_3}{\vartheta_1} \cdot T_0 \right); \frac{\Omega_4}{\Omega_1} = \left(1 - \frac{\vartheta_4}{\vartheta_1} \cdot T_0 \right); \frac{1}{\Omega_1} = \left(\frac{1}{\vartheta_1} \cdot T_0 \right) \quad (6)$$

$$\frac{\Omega_5}{\Omega_2} = \left(-\frac{\vartheta_5}{\vartheta_2} \cdot T_0 \right); \frac{\Omega_6}{\Omega_2} = \left(1 - \frac{\vartheta_6}{\vartheta_2} \cdot T_0 \right); \frac{1}{\Omega_2} = \left(\frac{1}{\vartheta_2} \cdot T_0 \right)$$

Tabla 1

Parámetros identificados de los robots móviles

	Pioneer 3DX	Pioneer 2DX	Pioneer 2DX (c/cámara)
ϑ_1	0,24089	0,3037	0,1992
ϑ_2	0,2424	0,2768	0,13736
ϑ_3	-0,00093603	-0,0004018	-0,001954
ϑ_4	0,99629	0,9835	0,9907
ϑ_5	-0,0037256	-0,003818	-0,01554
ϑ_6	1,0915	1,0725	0,9866

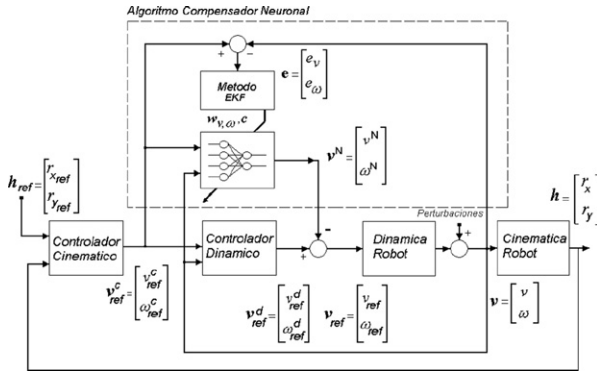


Figura 2. Estructura de control adaptable.

3. Controlador cinemático discreto

El diseño del controlador cinemático se basa en el modelo de la cinemática del robot móvil (4). Se propone el siguiente controlador cinemático:

$$\begin{bmatrix} v_{ref}^c(k) \\ \omega_{ref}^c(k) \end{bmatrix} = \begin{bmatrix} \frac{\cos \psi(k)}{T_0} & \frac{\sin \psi(k)}{T_0} \\ -\frac{1}{a} \frac{\sin \psi(k)}{T_0} & \frac{1}{a} \frac{\cos \psi(k)}{T_0} \end{bmatrix} \begin{bmatrix} r_{xs}(k+1) + l_x \tanh\left(\frac{k_x}{l_x} \tilde{r}_x(k)\right) \\ r_{ys}(k+1) + l_y \tanh\left(\frac{k_y}{l_y} \tilde{r}_y(k)\right) \end{bmatrix} - \begin{bmatrix} r_x(k) \\ r_y(k) \end{bmatrix} \quad (7)$$

donde $[v_{ref}^c(k), \omega_{ref}^c(k)]^T$ es el vector de valores deseados de velocidad lineal y angular, y $[r_{xs}(k), r_{ys}(k)]^T$ es el vector de posición deseada del robot. Considerando solo las variables de posición del robot en la trayectoria $[r_x(k), r_y(k)]^T$ y bajo la suposición de seguimiento perfecto de velocidad, $[v_{ref}^c(k), \omega_{ref}^c(k)]^T = [v(k), \omega(k)]^T$, puede sustituirse (7) en (4), resultando la ecuación de lazo cerrado:

$$\begin{bmatrix} \tilde{r}_x(k+1) \\ \tilde{r}_y(k+1) \end{bmatrix} + \begin{bmatrix} l_x & 0 \\ 0 & l_y \end{bmatrix} \begin{bmatrix} \tanh\left(\frac{k_x}{l_x} \tilde{r}_x(k)\right) \\ \tanh\left(\frac{k_y}{l_y} \tilde{r}_y(k)\right) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (8)$$

Definiendo el vector de error de salida $\tilde{\mathbf{h}}(k) = [\tilde{r}_x(k) \ \tilde{r}_y(k)]^T = [r_{xs}(k) \ r_{ys}(k)]^T - [r_x(k) \ r_y(k)]^T$, la ecuación (8) puede ser escrita como:

$$\tilde{\mathbf{h}}(k+1) = \begin{bmatrix} l_x \tanh\left(\frac{k_x}{l_x} \tilde{r}_x(k)\right) & l_y \tanh\left(\frac{k_y}{l_y} \tilde{r}_y(k)\right) \end{bmatrix}^T \quad (9)$$

lo cual implica que $\tilde{\mathbf{h}}(k) \rightarrow 0$ cuando $k \rightarrow \infty$. La suposición de seguimiento perfecto de velocidad será abandonada luego, en el análisis de estabilidad del controlador dinámico.

4. Controlador dinámico discreto

4.1. Formulación del problema

El controlador dinámico recibe las referencias de velocidad lineal y angular, las cuales son generadas por el controlador cinemático, y este genera otro par de comandos de velocidades lineal y angular que son enviados a los servos del robot móvil, como se muestra en la figura 2.

Primeramente se diseña el controlador dinámico principal basado sobre la dinámica nominal del robot. Esta dinámica nominal representa una estimación de la dinámica principal del robot móvil. La dinámica inversa del robot es obtenida en base a (5), sin considerar las incertidumbres dinámicas, y es parametrizada de la siguiente forma:

$$\begin{bmatrix} v_{ref}(k) \\ \omega_{ref}(k) \end{bmatrix} = \begin{bmatrix} v(k+1) & 0 & -\omega^2(k) & v(k) & 0 & 0 \\ 0 & \omega(k+1) & 0 & 0 & v(k)\omega(k) & \omega(k) \end{bmatrix} \boldsymbol{\Omega} \quad (10)$$

donde $\boldsymbol{\Omega}$ es el vector de parámetros discreto y es igual a:

$$\boldsymbol{\Omega} = [\Omega_1 \ \Omega_2 \ \Omega_3 \ \Omega_4 \ \Omega_5 \ \Omega_6]^T \quad (11)$$

La ecuación (10) puede ser reescrita como:

$$\begin{bmatrix} v_{ref}(k) \\ \omega_{ref}(k) \end{bmatrix} = \begin{bmatrix} \Omega_1 & 0 \\ 0 & \Omega_2 \end{bmatrix} \begin{bmatrix} v(k+1) \\ \omega(k+1) \end{bmatrix} + \begin{bmatrix} 0 & 0 & -\omega^2(k) & v(k) & 0 & 0 \\ 0 & 0 & 0 & 0 & v(k)\omega(k) & \omega(k) \end{bmatrix} \boldsymbol{\Omega} \quad (12)$$

En notación compacta se puede expresar de la siguiente forma:

$$\mathbf{v}_{ref}(k) = \mathbf{D}\mathbf{v}(k+1) + \boldsymbol{\eta}(k) \quad (13)$$

donde

$$\mathbf{v}_{ref}(k) = [v_{ref}(k) \ \omega_{ref}(k)]^T, \quad \mathbf{v}(k) = [v(k) \ \omega(k)]^T, \quad \boldsymbol{\eta}(k) = \begin{bmatrix} 0 & 0 & -\omega^2(k) & v(k) & 0 & 0 \\ 0 & 0 & 0 & 0 & v(k)\omega(k) & \omega(k) \end{bmatrix} \boldsymbol{\Omega}$$

donde $\mathbf{D} = \text{diag}(\Omega_1, \Omega_2)$, y siendo $\Omega_v = \Omega_1$, $\Omega_\omega = \Omega_2$.

La ley de control propuesta de la dinámica inversa es:

$$\mathbf{v}_{ref}^d(k) = \mathbf{G}(v(k), \omega(k), v_{ref}^c(k), \omega_{ref}^c(k), \dots, v_{ref}^c(k+1), \omega_{ref}^c(k+1)) \boldsymbol{\Omega} \quad (14)$$

donde

$$\mathbf{G} = \begin{bmatrix} \sigma_1(k) & 0 & -\omega^2(k) & v(k) & 0 & 0 \\ 0 & \sigma_2(k) & 0 & 0 & v(k)\omega(k) & \omega(k) \end{bmatrix} \quad (15)$$

$$\sigma_1(k) = v_{ref}^c(k+1) + \Lambda_v (v_{ref}^c(k) - v(k))$$

$$\sigma_2(k) = \omega_{ref}^c(k+1) + \Lambda_\omega (\omega_{ref}^c(k) - \omega(k))$$

En forma matricial se expresa:

$$\boldsymbol{\sigma}(k) = \mathbf{v}_{ref}^c(k+1) + \boldsymbol{\Lambda} (\mathbf{v}_{ref}^c(k) - \mathbf{v}(k)) \quad (16)$$

donde $\boldsymbol{\sigma}(k) = [\sigma_1(k) \ \sigma_2(k)]^T$ y $\boldsymbol{\Lambda} = \text{diag}(\Lambda_v, \Lambda_\omega)$

La ecuación (13) puede ser expresada como:

$$\mathbf{v}_{ref}^d(k) = \mathbf{D}\boldsymbol{\sigma}(k) + \boldsymbol{\eta}(k) \quad (17)$$

En (14), $\boldsymbol{\Omega}$ es el vector de parámetros, el cual es constante. Debido a las incertidumbres en el modelo nominal (incertidumbres paramétricas $\hat{\boldsymbol{\Omega}}$ y dinámica no considerada en el modelo $\hat{\mathbf{d}}$), se propone el uso de una compensación neuronal RBF. La ley de control completa puede ser expresada por:

$$[v_{ref}(k), \omega_{ref}(k)] = [\mathbf{D}\boldsymbol{\sigma}(k) + \boldsymbol{\eta}(k)] - \dots$$

$$NN(v(k), \omega(k), v_{ref}^c(k), \omega_{ref}^c(k), v_{ref}^c(k+1), \omega_{ref}^c(k+1)) \quad (18)$$

$$\mathbf{v}_{ref}(k) = \mathbf{v}_{ref}^d(k) - \mathbf{v}^N(k)$$

donde $NN(\cdot)$ es una función que indica la compensación neuronal que aprende la diferencia entre en la dinámica nominal y actual del robot móvil, y $\mathbf{v}_{ref}(k)$ es la acción de control del robot móvil.

La estructura completa de control propuesta se muestra en la figura 2, en la cual la implementación es expresada en función del tiempo discreto k .

4.2. Algoritmo compensador neural

Un compensador neuronal puede ser obtenido usando la información pasada y presente del vector de salida de la red $\mathbf{v}^N(k)$, y del vector de salida del sistema \mathbf{v} . Muchas RN son implementadas por computadores digitales y la parte dinámica de un sistema no lineal es obtenida utilizando operadores de atraso de tiempo (*time-delays*). Una red neuronal RBF implementada en un computador digital puede ser considerada como un sistema de control de tiempo discreto.

Para identificar la dinámica del robot se propone la siguiente RN cuya estructura está basada en una red neural de base radial (RBF):

$$\begin{cases} \mathbf{w}^*(k+1) = \mathbf{w}^*(k) \\ \mathbf{v}^N(k) = \mathbf{w}^{*T}(k) \boldsymbol{\xi}(\boldsymbol{\zeta}(k)) + \boldsymbol{\varepsilon} \end{cases} \quad (19)$$

donde \mathbf{w}^* representa los pesos óptimos del compensador neuronal $\mathbf{w}^* = [\mathbf{w}_v^* \ \mathbf{w}_\omega^*]$ y $\boldsymbol{\varepsilon} \in \mathbb{R}^{2 \times 1}$ es un vector de perturbaciones acotado.

El vector de funciones RBF es indicado por $\boldsymbol{\xi}^*$, siendo el número de estas funciones igual a M .

En la figura 3 se muestra la estructura interna del compensador neuronal, la cual está basada en una red RBF, donde t_a, t_b son enteros que están relacionados con el orden del sistema, donde z^{-1} es el operador de atraso de tiempo y $\mathbf{v}_{ref}^c(k+1)$ es el vector de salida deseado del sistema dinámico del robot móvil.

El vector de salida $\mathbf{v}^N(k)$ de la red de compensación neuronal es adicionada al vector $\mathbf{v}_{ref}(k)$ para generar la acción de control total del robot $\mathbf{v}_{ref}^d(k)$. Definimos el vector del patrón de entrada a la RN:

$$\boldsymbol{\zeta}(k) = [\mathbf{v}^N(k), \dots, \mathbf{v}^N(k-a+1), \mathbf{v}(k), \mathbf{v}(k-1), \dots, \mathbf{v}(k-b+1), \mathbf{v}_{ref}^c(k+1)] \quad (20)$$

donde $\mathbf{v}_{ref}^c(k) = [\mathbf{v}_{ref}^c(k), \omega_{ref}^c(k)]^T$, $\mathbf{v}^N = [\mathbf{v}^N(k), \omega^N(k)]^T$ y $\mathbf{v} = [\mathbf{v}(k), \omega(k)]^T$

Aquí, ξ es la función de base radial no lineal de la red, y se representa por:

$$\xi_i(\boldsymbol{\zeta}(k)) = \exp\left(-\frac{(\boldsymbol{\zeta}(k) - \mathbf{c}_i)^T (\boldsymbol{\zeta}(k) - \mathbf{c}_i)}{2\lambda^2}\right) \quad (21)$$

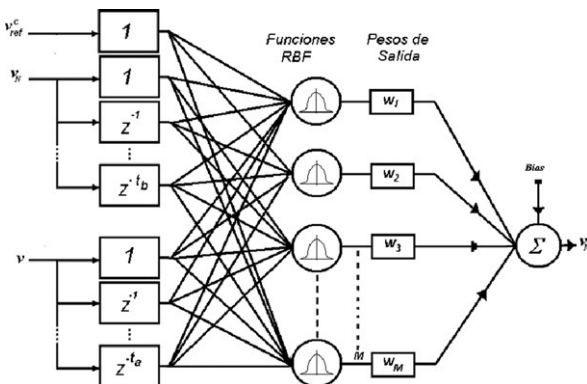


Figura 3. Estructura interna de la red neuronal RBF.

donde \mathbf{c} es el vector de centros de cada función $\xi(\cdot)$; λ es una constante de la función $\xi(\cdot)$, en nuestro caso es igual a 1, y M es el número de neuronas RBF.

El vector de pesos ideales \mathbf{w}^* es una cantidad artificial requerida solo para propósitos analíticos. En general se supone que existe un vector de pesos constantes \mathbf{w}^* pero desconocido cuya estima es $\mathbf{w}(k)$. Definiendo el error de pesos:

$$\tilde{\mathbf{w}}(k) = \mathbf{w}(k) - \mathbf{w}^* \quad (22)$$

donde la salida de la RN $\mathbf{v}^N(k)$ queda expresada por:

$$\mathbf{v}^N(k) = \mathbf{w}^T(k) \boldsymbol{\xi}(\boldsymbol{\zeta}(k)) - \tilde{\mathbf{w}}^T(k) \boldsymbol{\xi}(\boldsymbol{\zeta}(k)) + \boldsymbol{\varepsilon} \quad (23)$$

4.3. Algoritmo de entrenamiento usando filtro de Kalman extendido

El filtro de Kalman estima de modo óptimo los estados de un sistema lineal que está afectado con ruido blanco sobre los estados y las salidas. Para el entrenamiento de la RN basado en el filtro de Kalman, los pesos neuronales son los estados a ser estimados. En este caso, el error entre la salida de la referencia de velocidad ($\mathbf{v}_{ref}^c(k)$) y la salida de la dinámica del robot ($\mathbf{v}(k)$) puede ser considerado como un ruido blanco aditivo. Debido a este hecho y a que el mapeo de la RN es no lineal, se necesita un EKF [19].

El objetivo del entrenamiento es encontrar los valores óptimos de los pesos que minimizan la predicción del error [20]. Se aplica un algoritmo de entrenamiento basado en el filtro de Kalman desacoplado descrito por:

$$\begin{aligned} \mathbf{K}_{v,\omega}(k) &= \mathbf{P}_{v,\omega}(k) \mathbf{H}_{v,\omega}^T(k) \mathbf{M}_{v,\omega}(k) \\ \mathbf{w}_{v,\omega}(k+1) &= \mathbf{w}_{v,\omega}(k) + \gamma \mathbf{K}_{v,\omega}(k) \mathbf{e}_{v,\omega}(k) \end{aligned} \quad (24)$$

$$\mathbf{P}_{v,\omega}(k+1) = \mathbf{P}_{v,\omega}(k) - \mathbf{K}_{v,\omega}(k) \mathbf{H}_{v,\omega}^T(k) \mathbf{P}_{v,\omega}(k) + \boldsymbol{\Gamma}_{v,\omega}(k)$$

siendo

$$\begin{aligned} \mathbf{M}_{v,\omega}(k) &= [\mathbf{R}_{v,\omega}(k) + \mathbf{H}_{v,\omega}^T(k) \mathbf{P}_{v,\omega}(k) \mathbf{H}_{v,\omega}(k)] \\ \mathbf{e}(k) &= [\mathbf{v}_{ref}^c(k) - \mathbf{v}(k)] \end{aligned} \quad (25)$$

donde $\mathbf{e}_{v,\omega}(k)$ son los elementos del vector de error de salida, donde el subíndice v y ω indican su correspondencia con la velocidad lineal y la velocidad angular, respectivamente, $\mathbf{P}_{v,\omega}(k) \in \mathbb{R}^{M \times M}$ es la matriz de covarianza del error de predicción, $\mathbf{w}_{v,\omega}(k) \in \mathbb{R}^{M \times 1}$ es la estimación del vector de pesos de salida de la red RBF para v y ω , respectivamente, γ es el factor de aprendizaje ($0 < \gamma < 1$), $\mathbf{K}_{v,\omega}(k) \in \mathbb{R}^{M \times 1}$ es el vector de ganancia de Kalman, $\boldsymbol{\Gamma}_{v,\omega}(k) \in \mathbb{R}^{M \times M}$ es la matriz de covarianza del ruido que afecta los estados (pesos), $\mathbf{R}_{v,\omega}(k) \in \mathbb{R}^{1 \times 1}$ es la matriz de covarianza del ruido de medición. Usualmente $\mathbf{P}_{v,\omega}(k)$ y $\boldsymbol{\Gamma}_{v,\omega}(k)$ son inicializadas como matrices diagonales. Además, $\mathbf{H}_{v,\omega} \in \mathbb{R}^{M \times 1}$ es la matriz de medición, la cual puede obtenerse aplicando el método de la regla de la cadena sobre el vector de salida con respecto al vector de pesos. Esta matriz puede expresarse como:

$$\mathbf{H}_{v,\omega}(k) = \left[\frac{\partial \mathbf{v}(k)}{\partial \mathbf{w}(k)} \right] = \frac{\partial \mathbf{v}}{\partial \mathbf{v}^N} \frac{\partial \mathbf{v}^N}{\partial \mathbf{w}} \quad (26)$$

Las matrices $(\mathbf{H}_{v,\omega})$ pueden ser expresadas por:

$$\mathbf{H}_{v,\omega}(k) = \frac{\partial \mathbf{v}}{\partial \mathbf{w}} = \frac{\partial \mathbf{v}(k)}{\partial \mathbf{v}^N(k)} \cdot \boldsymbol{\xi}(\boldsymbol{\zeta}(k)) \quad (27)$$

Para utilizar eficazmente esta matriz $(\mathbf{H}_{v,\omega})$ es necesario conocer $\partial \mathbf{v} / \partial \mathbf{v}^N$, que es difícil de calcular cuando el modelo del sistema es desconocido o los parámetros del mismo varían. Sin embargo, se puede utilizar una aproximación, siendo esta una función acotada.

$$\frac{\partial \mathbf{v}}{\partial \mathbf{v}^N} = \text{sign} \left(\frac{\Delta \mathbf{v}}{\Delta \mathbf{v}^N} \right) \rightarrow \left\| \frac{\partial \mathbf{v}}{\partial \mathbf{v}^N} \right\| = 1 \quad (28)$$

La red RBF de compensación neural puede ser entrenada por el método presentado con las ecuaciones precedentes y proporciona un mapeo no lineal expresado por (19).

Generalmente las matrices $\Gamma_{v,\omega}(k)$, $\mathbf{R}_{v,\omega}(k)$ y $\mathbf{P}_{v,\omega}(k)$ son inicializadas como matrices diagonales. Es importante indicar que $\mathbf{H}_{v,\omega}(k)$, $\mathbf{K}_{v,\omega}(k)$ y $\mathbf{P}_{v,\omega}(k)$ son acotadas [21].

4.4. Entrenamiento y ubicación de los centros en las funciones de base radial

El procedimiento de entrenamiento para las redes RBF se divide en 2 etapas: el ajuste de los centros de las funciones de base radial en la capa oculta, seguido del entrenamiento de los pesos entre la salida y la capa oculta indicada por (24b). Generalmente en aplicaciones de control solo se entrena los pesos entre la salida y la capa oculta, donde los centros son ajustados fuera de línea.

Los algoritmos de ajuste de los centros más usados son el *fast k-means* (método de entrenamiento autorganizado) y el gradiente descendente (método supervisado) [19]. Para esta aplicación se usa el método del gradiente descendente.

La posición de los centros (capa oculta) puede ser expresada:

$$\begin{aligned} \mathbf{c}(k+1) &= \mathbf{c}(k) + \gamma \frac{\partial J(k)}{\partial \mathbf{c}(k)} \\ J(k) &= \frac{1}{2} (\mathbf{e}^T(k) \mathbf{e}(k)) \\ &= \frac{1}{2} \left[(\mathbf{v}_{ref}^c(k) - \mathbf{v}(k))^T (\mathbf{v}_{ref}^c(k) - \mathbf{v}(k)) \right] \end{aligned} \quad (29)$$

donde $J(k)$ es una función de costo. Para poder usar el método del gradiente descendente debemos aplicar la regla de la cadena a $\partial J(k)/\partial \mathbf{c}_i(k)$, y se obtiene:

$$\frac{\partial J(k)}{\partial \mathbf{c}_i(k)} = \frac{\partial J(k)}{\partial \mathbf{v}(k)} \frac{\partial \mathbf{v}(k)}{\partial \mathbf{v}^N(k)} \frac{\partial \mathbf{v}^N(k)}{\partial \mathbf{c}(k)} \quad (30)$$

donde

$$\frac{\partial \mathbf{v}^N(k)}{\partial \mathbf{c}(k)} = - \left[\mathbf{w}^T \xi(\zeta(k)) + \epsilon \right] \|\zeta(k) - \mathbf{c}(k)\| \quad (31)$$

Para usar el método del gradiente debemos conocer $\partial v/\partial v^N$, que fue aproximado por (28).

Sustituyendo (31) y (28) en (30):

$$\begin{aligned} \mathbf{c}(k+1) &= \mathbf{c}(k) + \gamma \mathbf{e}(k) \left(\text{sign} \left(\frac{\Delta \mathbf{v}}{\Delta \mathbf{v}^N} \right) \right) \dots \\ &\quad \left[\mathbf{w}^T \xi(\zeta(k)) \right] \|\zeta(k) - \mathbf{c}(k)\| \end{aligned} \quad (32)$$

Este resultado permite ajustar los centros de la red neuronal RBF-NN, que para fines prácticos ϵ no es considerado.

4.5. Descripción del algoritmo compensador neuronal

Hay que tener en cuenta que para el vector de salida deseado $\mathbf{v}_{ref}^c(k+1)$, el compensador producirá la señal $\mathbf{v}^N(k)$:

$$\begin{aligned} \mathbf{v}^N(k) &= NN \left[\mathbf{v}^N(k-1), \dots, \mathbf{v}^N(k-a+1), \mathbf{v}(k), \mathbf{v}(k-1), \right. \\ &\quad \left. \dots, \mathbf{v}(k-b+1), \mathbf{v}_{ref}^c(k+1) \right] \end{aligned}$$

El algoritmo de compensación de este sistema neural es descrito por los siguientes pasos:

- Inicializar aleatoriamente los pesos y centros de la red de compensación RBF neural $\mathbf{w}_i(0)$, \mathbf{c}_i ; e inicializar las matrices $\Gamma_{v,\omega}(k)$, $\mathbf{R}_{v,\omega}(k)$ y $\mathbf{P}_{v,\omega}(k)$ como matrices diagonales.
- Proporcionar el patrón de entrada de compensación neural en el momento de k :

$$\begin{aligned} \zeta(k) &= [\mathbf{v}^N(k-1), \dots, \mathbf{v}^N(k-a+1), \mathbf{v}(k), \mathbf{v}(k-1), \\ &\quad \dots, \mathbf{v}(k-b+1), \mathbf{v}_{ref}^c(k+1)] \end{aligned}$$

- Calcular la salida del compensación neural (19) como acción de control en el momento k .
- Tomar la salida $\mathbf{v}^N(k)$ y adicionar $\mathbf{v}_{ref}(k)$ al instante k , calcular el error de salida del sistema como:

$$\mathbf{e}(k+1) = \mathbf{v}_{ref}^c(k+1) - \mathbf{v}(k+1);$$

- Actualizar los pesos por medio de la ecuación (24b) y los centros por (32).
- Hacemos $k = k+1$ y retornamos al paso b) para calcular el próximo ciclo.

4.6. Análisis de estabilidad del controlador

Para la ley de control propuesta, la red neuronal RBF (19) entrenada con el algoritmo basado en el EKF (24) asegura que el error de estimación de la salida (25b) es uniformemente acotado; por lo tanto, los pesos de la red neuronal RBF se mantienen acotados, siendo:

$$\|\tilde{\mathbf{w}}_i\| \leq w_{Max} \quad (33)$$

donde w_{Max} es un valor constante.

Para realizar el análisis se debe determinar el error de control en el instante $k+1$ y teniendo en cuenta que la ley de control de la dinámica inversa representada por (14) se expresa como:

$$\mathbf{v}_{ref}^d(k) = \mathbf{G}(k) \tilde{\boldsymbol{\Omega}} + \mathbf{G}(k) \tilde{\boldsymbol{\Omega}} \quad (34)$$

La ley de control completa que incluye la compensación neuronal (18) puede ser expresada por:

$$\mathbf{v}_{ref}(k) = \mathbf{D}\sigma(k) + \mathbf{G}(k) \tilde{\boldsymbol{\Omega}} + \boldsymbol{\eta} - \mathbf{v}^N(k) \quad (35)$$

De (13), el modelo dinámico que incluye las incertidumbres está representado por:

$$\mathbf{v}_{ref}(k) = \mathbf{D}\mathbf{v}(k+1) + \boldsymbol{\eta} + \tilde{\boldsymbol{\delta}} \quad (36)$$

Igualando (36) y (35), el sistema de lazo cerrado resulta:

$$\mathbf{D}(\mathbf{v}(k+1) - \sigma(k)) - (\mathbf{G}(k) \tilde{\boldsymbol{\Omega}} - \tilde{\boldsymbol{\delta}}) = -\mathbf{v}^N(k) \quad (37)$$

Teniendo en cuenta la (25), que define el error control, (37) puede ser reescrita por:

$$\mathbf{D}(\mathbf{e}(k+1) + \boldsymbol{\Lambda}\mathbf{e}(k)) - (\mathbf{G}(k) \tilde{\boldsymbol{\Omega}} - \tilde{\boldsymbol{\delta}}) = -\mathbf{v}^N(k) \quad (38)$$

La compensación neuronal tiende a hacer el error de control igual a cero, y reemplazando el valor de $\mathbf{v}^N(k)$ expresado por (23):

$$\begin{aligned} \mathbf{D}(\mathbf{e}(k+1) + \boldsymbol{\Lambda}\mathbf{e}(k)) - (\mathbf{G}(k) \tilde{\boldsymbol{\Omega}} - \tilde{\boldsymbol{\delta}}) &= -\mathbf{w}^T(k) \xi(\zeta(k)) \\ &\quad + \tilde{\mathbf{w}}^T(k) \xi(\zeta(k)) - \epsilon \end{aligned} \quad (39)$$

e igualando ambos miembros en (39), queda:

$$\begin{aligned} \mathbf{D}(\mathbf{e}(k+1) + \boldsymbol{\Lambda}\mathbf{e}(k)) &= \tilde{\mathbf{w}}^T(k) \xi(\zeta(k)) - \epsilon \\ \mathbf{G}(k) \tilde{\boldsymbol{\Omega}} - \tilde{\boldsymbol{\delta}} &= \mathbf{w}^T(k) \xi(\zeta(k)) \end{aligned} \quad (40)$$

Reescribiendo (39):

$$\mathbf{e}(k+1) = -\boldsymbol{\Lambda}\mathbf{e}(k) + \mathbf{D}^{-1} (\tilde{\mathbf{w}}^T(k) \xi(\zeta(k)) - \epsilon) \quad (41)$$

Expresando (41) en función de sus componentes:

$$\begin{cases} e_v(k+1) = -\Lambda_v e_v(k) + \Omega_1^{-1} (\tilde{\mathbf{w}}_v^T(k) \xi(\zeta(k)) - \varepsilon_v) \\ e_\omega(k+1) = -\Lambda_\omega e_\omega(k) + \Omega_2^{-1} (\tilde{\mathbf{w}}_\omega^T(k) \xi(\zeta(k)) - \varepsilon_\omega) \end{cases} \quad (42)$$

Considerando la función candidata de Lyapunov para el análisis de estabilidad:

$$V(k) = \mathbf{e}^T(k) \mathbf{e}(k) = \sum_{i=v,\omega} e_i^2(k) = e_v^2(k) + e_\omega^2(k) \quad (43)$$

donde el subíndice i indica si corresponde a los elementos de v o de ω .

La ecuación en diferencias de Lyapunov expresada en función de los componentes de error de velocidad queda:

$$\Delta V(k) = \mathbf{e}^T(k+1) \mathbf{e}(k+1) - \mathbf{e}^T(k) \mathbf{e}(k) = \sum_{i=v,\omega} [e_i^2(k+1) - e_i^2(k)] \quad (44)$$

Reemplazando (42) en (43):

$$\begin{aligned} \Delta V(k) = \sum_{i=v,\omega} [& (\Lambda_i^2 - 1) e_i^2(k) - \dots \\ & - 2e_i(k) \Lambda_i^T \Omega_i^{-1} (\tilde{\mathbf{w}}_i^T \xi(\zeta(k)) - \varepsilon_i) + \Omega_i^{-2} (\tilde{\mathbf{w}}_i^T \xi(\zeta(k)) - \varepsilon_i)^2] \end{aligned} \quad (45)$$

Calculando las raíces de la ecuación (45) para determinar la cota de error de control, se obtiene:

$$\begin{aligned} -\frac{b}{2a} \pm \frac{\sqrt{b^2 - 4ac}}{2a} = -\frac{\Lambda_i^T \Omega_i^{-1} (\tilde{\mathbf{w}}_i^T \xi(\zeta(k)) - \varepsilon_i)}{(\Lambda_i^2 - 1)} \pm \dots \\ \Omega_i^{-1} \sqrt{\Lambda_i^2 (\tilde{\mathbf{w}}_i^T \xi(\zeta(k)) - \varepsilon_i)^2 - (\Lambda_i^2 - 1) (\tilde{\mathbf{w}}_i^T \xi(\zeta(k)) - \varepsilon_i)^2} \end{aligned} \quad (46)$$

El error queda acotado por:

$$|e_i(k)| \geq \frac{2 (\tilde{\mathbf{w}}_i^T \xi(\zeta(k)) - \varepsilon_i)}{\Omega_i (\Lambda_i + 1)} \quad (47)$$

De (47) para garantizar que $\Delta V(k)$ sea negativo y de esta forma el sistema de control propuesto sea asintóticamente estable se debe cumplir:

$$\min(|e_v(k)|, |e_\omega(k)|) \geq \frac{2 \|\tilde{\mathbf{w}}_i^T \xi(\zeta(k)) - \varepsilon_i\|}{\lambda_{\min}(\mathbf{D}) \lambda_{\min}(\mathbf{A} + \mathbf{I})} \quad (48)$$

Lo cual implica $\Delta V(k) \leq 0$. Esto asegura que $\Delta e(k) \rightarrow B_{\delta_e}$ está acotado.

Con este resultado se regresa al análisis del comportamiento del error de control de trayectoria $\tilde{\mathbf{h}}(k)$. Abandonando la suposición inicial de un seguimiento perfecto de velocidad de los controladores cinemáticos de la sección 3, (8) se reescribe ahora como:

$$\begin{bmatrix} \tilde{r}_x(k+1) \\ \tilde{r}_y(k+1) \end{bmatrix} + \begin{bmatrix} l_x & 0 \\ 0 & l_y \end{bmatrix} \begin{bmatrix} \tanh\left(\frac{k_x}{l_x} \tilde{r}_x(k)\right) \\ \tanh\left(\frac{k_y}{l_y} \tilde{r}_y(k)\right) \end{bmatrix} = \begin{bmatrix} \varepsilon_1(k) \\ \varepsilon_2(k) \end{bmatrix} \quad (49)$$

donde el vector de error $\mathbf{e}(k) = [\varepsilon_1(k) \ \varepsilon_2(k)]^T$ es el error de la velocidad de seguimiento previamente definido como $\Theta(k)\mathbf{e}(k)$. Siendo la matriz $\Theta(k)$:

$$\Theta(k) = T_0 \begin{bmatrix} \cos \psi(k) & -a \sin \psi(k) \\ \sin \psi(k) & a \cos \psi(k) \end{bmatrix} \quad (50)$$

De (49) hacemos:

$$L(\tilde{\mathbf{h}}(k)) = \begin{bmatrix} l_x & 0 \\ 0 & l_y \end{bmatrix} \begin{bmatrix} \tanh\left(\frac{k_x}{l_x} \tilde{r}_x(k)\right) \\ \tanh\left(\frac{k_y}{l_y} \tilde{r}_y(k)\right) \end{bmatrix} \quad (51)$$

Reescribiendo (49) en forma compacta, para pequeños errores de control, se puede aproximar $L(\tilde{\mathbf{h}}(k)) \approx \mathbf{K}_{xy} \tilde{\mathbf{h}}(k)$ con $\mathbf{K}_{xy} = \text{diag}(k_x, k_y)$

$$\tilde{\mathbf{h}}(k+1) + \mathbf{K}_{xy} \tilde{\mathbf{h}}(k) = \Theta \mathbf{e}(k) \quad (52)$$

Considerando la función candidata de Lyapunov:

$$V(k) = \frac{1}{2} \tilde{\mathbf{h}}^T(k) \tilde{\mathbf{h}}(k) > 0, \quad (53)$$

La diferencia discreta es:

$$\begin{aligned} \Delta V(k) &= V(k+1) - V(k) \\ &= \tilde{\mathbf{h}}^T(k+1) \tilde{\mathbf{h}}(k+1) - \tilde{\mathbf{h}}^T(k) \tilde{\mathbf{h}}(k) \\ &= \mathbf{e}^T(k) \Theta^T \Theta \mathbf{e}(k) - 2\mathbf{e}^T(k) \Theta^T \mathbf{K}_{xy} \tilde{\mathbf{h}}(k) + \\ &\quad \tilde{\mathbf{h}}^T(k) \mathbf{K}_{xy}^T \mathbf{K}_{xy} \tilde{\mathbf{h}}(k) - \tilde{\mathbf{h}}^T(k) \tilde{\mathbf{h}}(k) \end{aligned} \quad (54)$$

La condición suficiente para que ΔV sea negativa:

$$\|\mathbf{I} - \mathbf{K}_{xy}^2\| \|\tilde{\mathbf{h}}(k)\|^2 \geq \|\Theta\|^2 \|\mathbf{e}(k)\|^2 + 2 \|\Theta\| \|\mathbf{K}_{xy}\| \|\mathbf{e}(k)\| \|\tilde{\mathbf{h}}(k)\| \quad (55)$$

Calculando las raíces de la ecuación (65):

$$\begin{aligned} \|\tilde{\mathbf{h}}\| &> \frac{-2 \|\Theta\| \|\mathbf{K}_{xy}\| \|\mathbf{e}(k)\|}{2 \|\mathbf{I} - \mathbf{K}_{xy}^2\|} \\ &\pm \frac{\sqrt{4 \|\Theta\|^2 \|\mathbf{K}_{xy}\|^2 \|\mathbf{e}(k)\|^2 + 4 \|\mathbf{I} - \mathbf{K}_{xy}^2\| \|\mathbf{e}(k)\|^2 \|\Theta\|^2}}{2 \|\mathbf{I} - \mathbf{K}_{xy}^2\|} \\ &= \frac{-2 \|\Theta\| \|\mathbf{K}_{xy}\| \|\mathbf{e}(k)\| \pm 2 \|\Theta\| \|\mathbf{e}(k)\|}{2 \|\mathbf{I} - \mathbf{K}_{xy}^2\|} \end{aligned} \quad (56)$$

La condición suficiente para estabilidad asintótica para que $\Delta V \leq 0$ pueda ser expresada como:

$$\|\tilde{\mathbf{h}}\| \geq \frac{(1 - \|\mathbf{K}_{xy}\|) \|\Theta\| \|\mathbf{e}(k)\|}{\|\mathbf{I} - \mathbf{K}_{xy}^2\|} \quad (57)$$

Esto implica que $\Delta V < 0$. Asegurando que $\tilde{\mathbf{h}} \rightarrow B_{\delta_{\tilde{\mathbf{h}}}}$.

Esto constituye un resultado práctico que permite afirmar que el error de control de posición queda finalmente acotado en términos del error de aproximación del algoritmo neuronal de compensación.

5. Resultados experimentales

El robot móvil Pioneer 2-DX posee a bordo una PC Pentium III con 800 MHz y 512 Mb de memoria RAM (fig. 4), y los algoritmos de control propuesto se aplicaron en el robot que admite velocidades lineales y angulares, como las señales de entrada de referencia. El controlador fue inicializado con los parámetros dinámicos del robot Pioneer 3-DX y se usaron 5 neuronas RBF ($M=5$). Los parámetros dinámicos fueron obtenidos a través de la identificación, según lo propuesto por Song y Grizzle [21].

El objetivo del experimento es demostrar que para distintas velocidades y distintas trayectorias el robot móvil sigue la

trayectoria de referencia lo más fielmente posible con el menor error. El robot debe seguir una trayectoria indicada por:

$$\begin{cases} r_{xs} = 0,3 \sin(0,05\pi kT_0) \\ r_{ys} = 1,5 \sin(0,025\pi kT_0) \end{cases} \quad (58)$$

En el experimento, el robot comienza en la posición $(r_x, r_y) = (0,0)$ m y sigue una trayectoria de referencia en forma de 8 horizontal (58). El experimento se realizó usando el algoritmo de compensación neuronal y se comparó con otro experimento desactivando el algoritmo de compensación neuronal. Los resultados se describen a continuación.

Las figuras 5 y 6 describen las velocidades y acciones de control del compensador neuronal RBF, así como las referencias y las variables de posición del robot móvil, respectivamente. La figura 7 muestra la trayectoria seguida por el robot móvil, sin compensación y con el compensador neuronal adaptable para el seguimiento de la trayectoria deseada. Se observa claramente en dicha figura que el robot con compensador neuronal sigue más fielmente la trayectoria deseada.

La figura 8 muestra la norma del error de trayectoria de los controladores con y sin compensación. El error de trayectoria se define como la distancia instantánea entre la referencia y la posición del robot. También muestra que el error con compensador neuronal discreto es menor cuando los parámetros dinámicos no son los correctos, debido a la mala sintonía del controlador o porque estos han variado a consecuencia de perturbaciones externas.



Figura 4. Robot Móvil Pioneer 2DX.

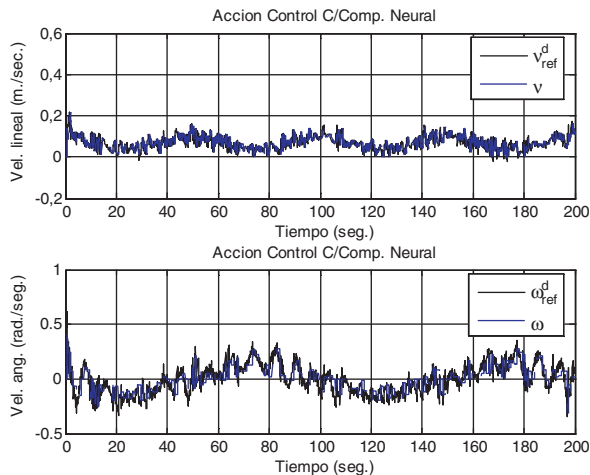


Figura 5. Velocidades lineal y angular del robot y acciones de control del controlador con compensador neuronal.

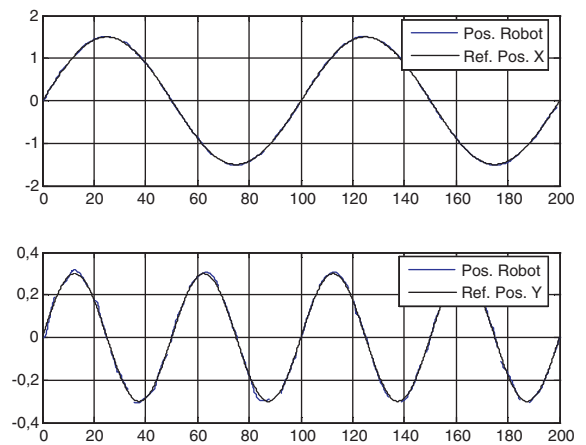


Figura 6. Referencia de posición y posición instantánea del robot móvil.

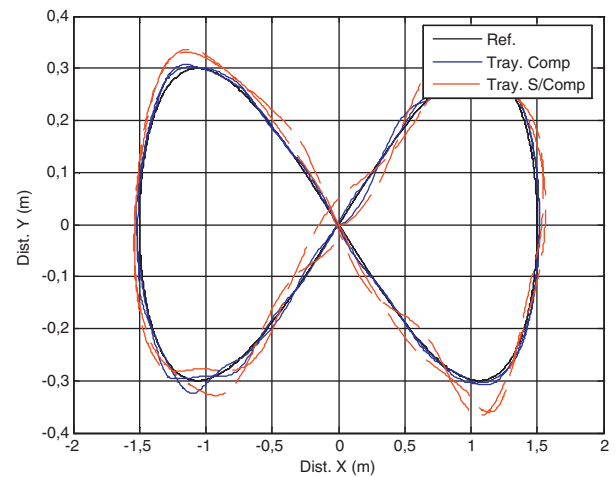


Figura 7. Trayectoria de referencia y posición instantánea del robot en el plano de coordenadas X-Y.

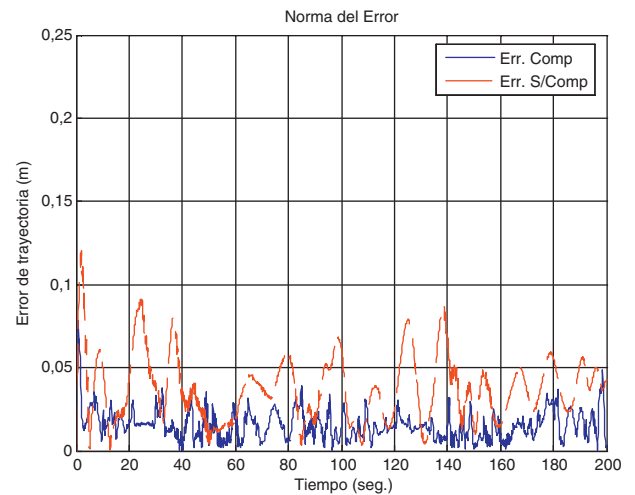


Figura 8. Norma del error de distancia con compensador neuronal. El error graficado con línea de trazo indica sin compensación; en línea continua, con compensación neuronal.

En el experimento no se utilizó un sistema de referencia absoluto. Se comparó el controlador sin compensación y con compensación, y se muestra que con este último se mejora el seguimiento. El mismo resultado se esperaría con un sistema de referencia absoluto.

6. Conclusiones

En este trabajo se propuso un algoritmo de compensación neuronal adaptable para el seguimiento de la trayectoria de un robot móvil del tipo unicycle. Este compensador, junto al controlador no lineal, es capaz de generar los comandos de velocidad con mínimo error para el mismo. Se demostró que los errores de control de seguimiento están acotados, y que los límites son calculados en función del error de aproximación de la red neuronal RBF.

El algoritmo neuronal adaptable RBF aprende la diferencia entre una dinámica nominal conocida y la dinámica real del robot para su posterior compensación. Por lo tanto, el esfuerzo computacional es significativamente menor que un modelo completo de RN inversa que tuviera que aprender toda la dinámica del robot. Los resultados experimentales muestran el buen desempeño del controlador de seguimiento propuesto y su capacidad para adaptarse a la dinámica de robots reales.

Apéndice 1. Listado de variables

v	Velocidad lineal del robot móvil
ω	Velocidad angular desarrollada por el robot móvil
\mathbf{v}	Vector de velocidades del robot móvil
r_x, r_y	Coordenadas del robot (pto. h) en el plano XY
\mathbf{h}	Vector de interés con coordenadas r_x, r_y en el plano XY
G	Centro de masa del robot
C	Posición de la rueda castor
E	Ubicación de la herramienta
ψ	Orientación del robot
a	Distancia entre el punto de interés y el punto central del eje virtual vinculado a las ruedas de tracción
δ	Vector de incertidumbres del robot
\mathfrak{P}	Vector de parámetros del robot (tiempo continuo)
\mathfrak{P}_i	Elementos del vector de parámetros (tiempo continuo) donde $i = 1, \dots, 6$
Ω	Vector de parámetros del robot (tiempo discreto)
Ω_i	Elementos del vector de parámetros (tiempo discreto) donde $i = 1, \dots, 6$
\mathbf{D}	Matriz diagonal $\text{diag}(\Omega_1, \Omega_2)$
T_0	Periodo de muestreo
k	Índice de tiempo discreto
M	Número de neuronas RBF
$\mathbf{w}_{v,\omega}(k)$	Pesos de salida de la red neuronal RBF
$\tilde{\mathbf{w}}_{v,\omega}(k)$	Error de pesos de salida
$\mathbf{w}_{v,\omega}^*$	Pesos óptimos de salida
$\xi_i(\cdot)$	Funciones RBF
$\xi^T(\cdot)$	Vector de funciones RBF
λ	Constante de la función $\xi_i(\cdot)$
\mathbf{c}_i	Vector de centros asociados a la función $\xi_i(\cdot)$
$\mathbf{v}^N(k)$	Vector de salida de la red RBF
$\mathbf{v}_{ref}^c(k)$	Vector de salida del controlador cinemático
$\mathbf{v}_{ref}^d(k)$	Vector de salida del controlador dinámico
$\mathbf{v}_{ref}(k)$	Vector de acción de control total para la dinámica del robot ($v_{ref}(k) = v_{ref}^d(k) - v^N(k)$)
$\mathbf{P}_{v,\omega}(k) \in \Re^{M \times M}$	Matriz de covarianza del error de predicción
$\mathbf{w}_{v,\omega}(k) \in \Re^{M \times 1}$	Estimación del vector de pesos de salida

γ	Factor de aprendizaje ($0 < \gamma < 1$)
$\mathbf{K}_{v,\omega}(k) \in \Re^{M \times 1}$	Vector de ganancia de Kalman. $\Gamma_{v,\omega}(k) \in \Re^{M \times M}$ es la matriz de covarianza del ruido que afecta los estados (pesos)
$\mathbf{R}_{v,\omega}(k) \in \Re^{1 \times 1}$	Matriz de covarianza del ruido de medición
$\mathbf{H}_{v,\omega} \in \Re^{M \times 1}$	Matriz de medición
$J(k) \in \Re^{1 \times 1}$	Funcional de costo que es usado para la localización de centros de la RBF
$\mathbf{e}(k) \in \Re^{2 \times 1}$	Vector de error de velocidades de salida, siendo $\mathbf{e}(k) = [e_v(k), e_\omega(k)]^T$
$e_{v,\omega}(k) \in \Re^{1 \times 1}$	Error de salida, para v y ω respectivamente
$\boldsymbol{\varepsilon} \in \Re^{2 \times 1}$	Vector de perturbaciones (acotado)
$\Theta(k)$	Matriz $T_0 \begin{bmatrix} \cos \psi(k) & -a \sin \psi(k) \\ \sin \psi(k) & a \cos \psi(k) \end{bmatrix}$

Bibliografía

- [1] W. Wu, H. Chen, Y. Wang, P. Woo, Adaptive Exponential Stabilization of Mobile Robots with Uncertainties, en: Proceedings of the IEEE 38th Conference on Decision and Control, Phoenix, Arizona, EE.UU., 1999, pp. 3484–3489.
- [2] F. Kühne, J. Gomes, W. Fetter, Mobile Robot Trajectory Tracking Using Model Predictive Control, en: II IEEE Latin-American Robotics Symposium, São Luis, Brasil, 2005.
- [3] L. Gracia, J. Tornero, Kinematic control of wheeled mobile robots, Lat. Am. Appl. Res. 38 (2008) 7–16.
- [4] S. Liu, H. Zhang, S.X. Yang, J. Yu, Dynamic Control of a Mobile Robot Using an Adaptive Neurodynamics and Sliding Mode Strategy, en: Proceedings of the 5th World Congress on Intelligent Control and Automation, Hangzhou, China, 2004, pp. 5007–5011.
- [5] W. Dong, Y. Guo, Dynamic tracking control of uncertain mobile robots, IEEE/RSJ International Conference on Intelligent Robots and Systems (2005) 2774–2779.
- [6] Y. Liyong, X. Wei, An Adaptive Tracking Method for Non-holonomic Wheeled Mobile Robots, en: Proceedings of the 26th Chinese Control Conference, Zhangjiajie, Hunan, China, July 26–31, 2007.
- [7] V.S.K. Chaitanya, P.K. Sarkar, A Neural Network Algorithm for the Error Optimization in the Path Tracking Control of a Mobile Robot, en: International Joint Conference on Neural Networks Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, July 16–21, 2006.
- [8] T. Das, I.N. Kar, Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots, IEEE Trans. Control Syst. 14 (2006) 501–510.
- [9] T. Fukao, H. Nakagawa, N. Adachi, Adaptive Tracking Control of a Mobile Robot, IEEE Robot & Autom 16 (2000) 609–615.
- [10] J. Velagic, N. Osmic, B. Lacevic, Neural network controller for mobile robot motion control, WASET 47 (2008) 93–198.
- [11] W. Dong, W. Huo, Tracking Control of Wheeled Mobile Robots with Unknown Dynamics, en: Proceedings of the IEEE International Conference on Robotics & Automation, Detroit, Michigan, 1999, pp. 2645–2650.
- [12] F.N. Martins, C.C. Wanderley, R. Carelli, M. Sarcinelli Filho, T.F. Bastos Filho, An adaptive dynamic controller for autonomous mobile robot trajectory tracking, Control Eng. Pract. 16 (2008) 1354–1363.
- [13] F.G. Rossomando, C. Soria, R. Carelli, Control de robots móviles con incertidumbres dinámicas usando redes de base radial, Rev. Iberoam. Autom. Inf. Ind. 7 (2010) 28–35.
- [14] M.K. Bugeja, S.G. Fabri, L. Camilleri, Dual adaptive dynamic control of mobile robots using neural networks, IEEE Trans. Syst. Man Cybern. 39 (2009) 129–141.
- [15] A. Fel'dbaum, Optimal Control Systems, Academic Press, New York, NY, 1965.
- [16] A.Y. Alanis, E.N. Sanchez, A.G. Loukianov, Discrete-time adaptive backstepping nonlinear control via high-order neural networks, IEEE Trans. Neural Networks 18 (2007) 1185–1195.
- [17] M.S. Kim, J.H. Shin, J.J. Lee, Design of a Robust Adaptive Controller for a Mobile Robot, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (2000) 1816–1821.
- [18] C. De la Cruz, R. Carelli, Dynamic Modeling and Centralized Formation Control of Mobile Robots, en: 32nd Annual Conference of the IEEE Industrial Electronics Society IECON, París, 2006.
- [19] S. Haykin, Kalman Filtering and Neural Networks, John Wiley & Sons, Inc, 2001.
- [20] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd edition, Prentice Hall, 2001.
- [21] Y. Song, J.W. Grizzle, The extended Kalman filter as local asymptotic observer for discrete-time nonlinear system, JMSEC 5 (1995) 59–78.