

Diseño de una estrategia de control para robots móviles utilizando técnicas de álgebra lineal (LABC) y estimación neuronal en el seguimiento de trayectorias

Carlos A. Vacca^{a,*}, Eduardo G. Scaglia^b, Fernando C. Ulloa-Vasquez^c, Francisco G. Rossomando^d

^aUTN Facultad Regional La Rioja, Av. San Nicolás de Bari 1100 (Este), La Rioja, Argentina.

^bInstituto de Investigaciones Químicas, Facultad de Ingeniería, UNSJ, C/Av Libertador 1109 (Oeste), CP 5400, San Juan, Argentina.

^cPrograma Investigación Radio Digital, PIRD. Facultad de Ingeniería, Universidad Tecnológica Metropolitana. Virgino Arias 1369, Ñuñoa, CP 7750000 Santiago-Chile.

^dInstituto de Automática, Facultad de Ingeniería, UNSJ-CONICET, C/Av Libertador 1109 (Oeste), CP 5400, San Juan, Argentina.

To cite this article: Vacca, C.A., Scaglia, G.E., Ulloa-Vasquez, F.C., Rossomando, F.G. (2024). Linear algebra-based controller (LABC) for trajectory tracking in mobile robots using neural estimation. *Revista Iberoamericana de Automática e Informática Industrial* 00, 1-5. <https://doi.org/10.4995/riai.2024.21484>

Resumen

Los inconvenientes planteados por el seguimiento de trayectorias usando robots móviles es un tema vigente en la teoría de control, en esta propuesta se presenta el diseño de un controlador de álgebra lineal en combinación con un estimador neuronal. Donde además el robot móvil cuenta con incertidumbres aditivas. Los valores de incertidumbre en cada momento de muestreo se obtienen mediante estimación basada en Redes Neuronales, donde se incluye el diseño de un estimador neuronal del error de modelado junto con la demostración de la convergencia a cero del error de seguimiento. La técnica de control propuesta se valida mediante simulación y resultados experimentales. El controlador de Álgebra Lineal y el estimador neuronal demuestran que se puede utilizar para reducir el efecto de las incertidumbres aditivas en el error de control de seguimiento.

Palabras clave: Álgebra Lineal, Control Neuronal, Control No lineal, Identificación, Robots Móviles

Linear algebra-based controller (LABC) for trajectory tracking in mobile robots using neural estimation

Abstract

The drawbacks posed by trajectory tracking using mobile robots is a current issue in control theory, in this proposal we present the design of a linear algebra controller in combination with a neural estimator. In addition, the mobile robot has additive uncertainties. The uncertainty values at each sampling time are obtained by estimation based on Neural Networks, where the design of a neural estimator of the modeling error is included together with the demonstration of the convergence to zero of the tracking error. The proposed control technique is validated by simulation and experimental results. The Linear Algebra controller and the neural estimator demonstrate that it can be used to reduce the effect of additive uncertainties on the tracking control error.

Keywords: Linear Algebra, Neural Control, Non Linear Control, Identification, Mobile Robots.

1. Introducción

En las últimas décadas, la robótica móvil ha incrementado su participación en las distintas actividades socio-económicas. Siendo el caso de la automatización industrial, la agricultura y los distintos tipos de servicios, incluyendo la defensa, donde se destaca la vigilancia, la exploración y de rescate en caso de

sinistros. Yang et al. (2019) Hassan y Saleem (2021) Deng et al. (2021) Zhang et al. (2022).

Los diferentes robots móviles con ruedas (WMR), debido a su adaptabilidad, performance y robustez, se utilizan muy a menudo en las tareas citadas anteriormente. Los WMR tradicionales que en sus cuatro ruedas, posee accionamiento independiente utilizan por lo general motores eléctricos Zhang et al.

*Autor para correspondencia: carlosvaccasis@gmail.com

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

(2024), siendo la principal ventaja que tiene un modelo sencillo y es de fácil mantenimiento. Si se compara con otros sistemas, como el sistema de tracción de correa dentada, este último es ligero, puede absorber las vibraciones y por lo general ofrece un torque elevado. En las técnicas de control de seguimiento de trayectoria aplicadas a robot móviles sobre ruedas existen diferentes tipos de controladores cinemáticos y dinámicos. En el caso de los controladores diseñados teniendo en cuenta el modelo cinemático del robot, se puede calcular directamente el comando de velocidad de salida deseada para cada rueda motriz, sin tener en cuenta otros fenómenos físicos del robot Wit et al. (1993), De La Cruz et al. (2011). Además, los robots móviles (WMR), al no poseer dirección del tipo CAR-LIKE, consiguen determinar la orientación aplicando diferentes velocidades en las ruedas de los lados izquierdo y derecho del robot, lo que también es propenso a producir un fenómeno de deslizamiento Tian y Sarkar (2014). Para mejorar el rendimiento del control del robot móvil, considerando el fenómeno de deslizamiento, se ha propuesto la estrategia de control del robot móvil de dirección deslizante basada en el modelo dinámico Rossomando et al. (2014), Tzafestas (2013), Andaluz Ortiz (2011). Comparado con el controlador cinemático, el controlador dinámico puede planificar directamente el comando de la fuerza motriz de cada rueda. Jung (2012). Diferentes autores han propuesto varias metodologías de control para el seguimiento de la trayectoria, algunas de las cuales se basan en los modelos cinemáticos o dinámicos de los vehículos (Wu et al. (2020); Lopez-Cortés et al. (2021)), los cuales se diferencian en función de la velocidad operativa y la precisión del modelo dinámico. En el caso de Chwa (2004) propone dos controladores denominados controlador de posición y controlador de rumbo, respectivamente. El primer controlador se encarga de asegurar el seguimiento de la posición y el segundo se activa cuando el error de seguimiento es suficientemente pequeño y la referencia de seguimiento no cambia de posición. Esto permite reducir el error sobre la orientación del vehículo al final de la trayectoria. En el control de vehículos autónomos es importante destacar los efectos retardos de propagación que aparecen en el control del vehículo y como compensar sus efectos indeseados (Acosta (2004)). En el trabajo de Yue et al. (2017), presentan una estructura de control de doble lazo cerrado con un observador de perturbaciones. Donde se podría realizar globalmente la estabilización de seguimiento de posturas y tiene capacidad de adaptación a perturbaciones. En el lazo exterior, se aplica una nueva función con características sinusoidales y de tangente inversa, que son continuamente diferenciables, para construir el controlador de velocidad basado en control backstepping.

También se presentan distintos métodos de control clásicos, donde Normey-Rico et al. (2001) implementan un control del tipo PID para el seguimiento de referencias de cambio de escalón con resultados aceptables, aunque el rendimiento disminuye cuando sigue referencias variables o cuando aparecen incertidumbres aditivas. Una alternativa para mejorar estos resultados es combinar una estructura de control simple (control PID) con esquemas robustos, como el control adaptable y las redes neuronales (Damodaran et al. (2017); Gu y Hu (2002); Park et al. (2016); Ye (2008a)). Esta mejora compensa estas incertidumbres y permite un seguimiento de referencia variable. Sin embargo, en la aplicación real son menos factibles de aplicar.

En esta área existen técnicas que se basan en los métodos "look-ahead" (Das y Kar (2006), Rossomando et al. (2012)), donde, en lugar del centro de masa en los robots móviles con ruedas, se elige el punto de intersección de una línea recta que pasa por el centro del vehículo y un eje de las dos ruedas para definir la configuración de la postura para hacer uso de la técnica de linealización por realimentación. Este enfoque presenta el inconveniente de que cuanto mayor es la distancia entre el centro de masa y el punto de intersección, mayor es la disminución del rendimiento del seguimiento. Por otro lado, cuando esta distancia es pequeña, la entrada de control tiende a un valor elevado, debido a la matriz que contiene este parámetro su inversa tiende a un valor singular. Por lo tanto, también es deseable desarrollar el modelo cinemático analítico que adopta el centro de masa sin problemas de singularidad. En Scaglia et al. (2009) se propone una técnica de control de seguimiento de trayectoria, donde su originalidad se enfoca en la aplicación del álgebra lineal para desarrollar un controlador para el seguimiento de trayectorias, en el cual las acciones de control se obtienen resolviendo un sistema de ecuaciones lineales. La metodología desarrollada para el seguimiento de la trayectoria deseada (x_{ref}, y_{ref}) se basa en determinar las trayectorias del resto de variables de estado con el objetivo de que el error de seguimiento tiende a cero. Estos estados se determinan analizando las condiciones para que el sistema de ecuaciones lineales tenga una solución exacta. Esta técnica de diseño se ha aplicado con éxito en varios sistemas (Candelino et al. (2014); Rosales et al. (2011); Scaglia et al. (2014); Serrano et al. (2013)). Además, esta metodología se ha extendido al problema de seguimiento de trayectorias con limitación de velocidades (Serrano et al. (2016), Serrano et al. (2017)). A pesar de esta ventaja varios de los trabajos mencionados no tuvieron en cuenta la incertidumbre aditiva en la etapa de diseño del controlador. Este trabajo de investigación se presenta una alternativa para el diseño de técnicas de control para el seguimiento de trayectorias en sistemas no lineales multivariables bajo la influencia de incertidumbres aditivas. Este tipo de problemas es de continuo interés para la investigación en el área de control. En la literatura se presentan distintas soluciones (ver por ejemplo Farooq et al. (2014); Lee et al. (2009); Wang (2012) y muchos otros). Esta propuesta emplea una estrategia de control que utiliza la técnica de álgebra lineal en combinación con una red neuronal para el seguimiento de trayectorias en robots móviles sobre ruedas considerando incertidumbres aditivas. Para el análisis de este problema, se añade un nuevo término que representa la incertidumbre en el modelo del robot móvil. A continuación, este término se estima utilizando una red neuronal del tipo RBF. En esta propuesta se considera que la incertidumbre aditiva es continua y además que posee derivadas continuas. Y considerando los valores pasados de estas incertidumbres es posible estimar sus valores futuros. En cada período, la red neuronal utiliza una regla de ajuste de los parámetros para reducir el error de estimación que fue obtenida utilizando la teoría de Lyapunov. Por último, se deduce una nueva ley de control que tiene en cuenta la condición para que un sistema de ecuaciones lineales tenga solución exacta. El principal aporte de este trabajo es que la técnica propuesta en la sencillez de su implementación, y no necesita aumentar el costo computacional para obtener la acción de control. La técnica aquí propuesta presenta una novedosa combinación de LABC

junto con estimación neuronal para el seguimiento de trayectoria con bajo costo computacional, obteniendo un pequeño error de seguimiento y bajo esfuerzo de control. Además, en comparación con Scaglia et al. (2020) y Serrano et al. (2017), este tiene en cuenta la incertidumbre aditiva en la formulación del problema, y el empleo de un estimador neuronal para reducir el error de seguimiento. También puede considerarse como un aporte importante la prueba de la convergencia a cero del error de seguimiento bajo el efecto de las incertidumbres. También se incluye la implementación del estimador junto con su correspondiente análisis de estabilidad en tiempo discreto que se muestra en el apéndice. Este enfoque de control esta totalmente diseñado en tiempo discreto, por lo tanto, el algoritmo desarrollado es más sencillo de implementar en un sistema real, ya que el uso de ecuaciones en tiempo discreto permite la aplicación directa a cualquier sistema microcontrolado o informático que ejecute instrucciones secuenciales a velocidad de reloj programable. En este trabajo de investigación, la propuesta de control es validada sobre un modelo de robot móvil PIONEER 3AT (P3-AT). Donde en la Sección 2 se describe el modelo cinemático, en la Sección siguiente (Sección 3) se enuncia el problema de esta propuesta de control, y se formula el estimador neuronal que será utilizado en la aproximación de las incertidumbres. A continuación, (Sección 4) se plantea el controlador propuesto. Y en las siguientes Secciones (Sección 5) se muestran los resultados, los cuales fueron realizados sobre modelos de simulación, como así también su discusión. Y por último (Sección 6) se presentan las conclusiones en base a los resultados obtenidos.

2. Cinemática del Robot Móvil

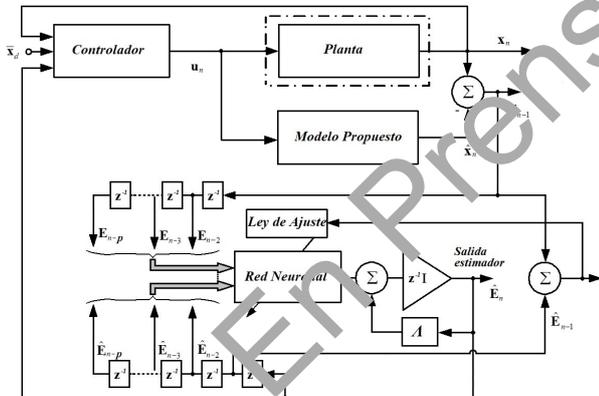


Figura 1: Estructura de Control

En esta sección se presenta el modelo cinemático de un robot móvil noholonómico (Scaglia et al. (2020); Yang et al. (2016); Jung (2012)) indicado en (1). Este modelo se utilizara para demostrar la viabilidad de la técnica de control propuesta sobre la cinemática del robot móvil.

$$\frac{d}{dt} \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} = \begin{pmatrix} v_n \cos(\theta) \\ v_n \sin(\theta) \\ \omega_n \end{pmatrix} \quad (1)$$

Donde:

v : indica la velocidad lineal del robot móvil, ω : es la velocidad angular del robot móvil. (x, y) : representa la posición cartesiana. θ : es la orientación del robot.

Es importante aclarar que estas variables estan indicadas en el dominio continuo.

3. Descripción del Problema

Para la obtención de la ley de control que que el robot móvil siga fielmente la trayectoria predeterminada (x_d, y_d) , pero para obtener el controlador es necesario discretizar el modelo cinemático (1) por el método de Euler, donde se obtiene un conjunto de ecuaciones en tiempo discreto:

$$\mathcal{X}_{n+1} = \begin{pmatrix} x_{n+1} \\ y_{n+1} \\ \theta_{n+1} \end{pmatrix} = \begin{pmatrix} x_n + \tau_s v_n \cos \theta_n \\ y_n + \tau_s v_n \sin \theta_n \\ \theta_n + \tau_s \omega_n \end{pmatrix} = \hat{f}(\mathcal{X}_n, \mathbf{u}_n) \quad (2)$$

Donde las variables mencionadas en la ecuación (2)son las siguientes:

(x_n, y_n) : representa la posición instantanea del robot en el plano XY en el instante $n\tau_s$.

v_n : representa la velocidad lineal del robot.

ω_n : indica la velocidad angular del robot.

θ_n : es la orientación del robot móvil con respecto al eje X.

n : Es el índice de tiempo discreto. $n \in 1, 2, 3, \dots$

τ_s : representa el período de muestreo.

Si ahora consideramos una planta real de la forma:

$$\mathcal{X}_{n+1} = f(\mathcal{X}_n, \mathbf{u}_n) \quad (3)$$

Entonces la incertidumbre aditiva se puede expresar de la siguiente forma:

$$\mathbf{E}_n = f(\mathcal{X}_n, \mathbf{u}_n) - \hat{f}(\mathcal{X}_n, \mathbf{u}_n) \quad (4)$$

Donde:

$\hat{f}(\mathcal{X}_n, \mathbf{u}_n)$: es el modelo no lineal del robot expresado en tiempo discreto.

Si se considera que, \mathcal{X} y \mathbf{u} están acotados y f es una función Lipschitz, es posible modelar \mathbf{E}_n como una incertidumbre acotada (Rosales et al. (2019); Rossomando et al. (2014)) Si se incorpora la incertidumbre aditiva en el modelo del sistema (3), se puede expresar como:

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \\ \theta_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \\ \theta_n \end{pmatrix} + \tau_s \begin{pmatrix} \cos(\theta_n) & 0 \\ \sin(\theta_n) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_n \\ \omega_n \end{pmatrix} + \underbrace{\begin{pmatrix} E_{x,n} \\ E_{y,n} \\ E_{\theta,n} \end{pmatrix}}_{\mathbf{E}_n} \quad (5)$$

Donde \mathbf{E}_n : representa la incertidumbre aditiva. El vector \mathbf{E}_n se obtiene de la incertidumbre muestreada en el instante n . Suponiendo que la incertidumbre es una función continua; entonces es posible usar \mathbf{E}_n para representar sistemas perturbados, (SiraRamírez et al. (2013); Scaglia et al. (2020); Sun et al. (2017) y Fareh et al. (2021)). Teniendo en cuenta que \mathbf{E}_n es una función continua, entonces es posible aproximarla por la aplicación de redes neuronales Rossomando et al. (2012). Es importante obtener un valor lo mas próximo posible de esta incertidumbre para obtener una ley de control con el mínimo error. El desafío de este trabajo es obtener una buena estimación de esta incertidumbre para que el robot siga la trayectoria con mayor fidelidad a la trayectoria deseada. En la siguiente subsección se desarrollará el análisis para implementar un estimador neuronal adaptable para esta clase de incertidumbres.

3.1. Estimación Neuronal

De la ec. (5) se puede ver que \mathbf{E}_n es la incertidumbre en el instante n , pero en realidad nosotros conoceremos esa incertidumbre en el instante $n + 1$. Entonces en el instante n conocemos \mathbf{E}_{n-1} , \mathbf{E}_{n-2} , \mathbf{E}_{n-3} ... y utilizando toda esa información podemos estimar \mathbf{E}_n . Así, según la teoría de Redes Neuronales, es posible estimar el valor de la incertidumbre en el momento actual considerando los valores anteriores. A continuación, se propone un enfoque para eliminar la influencia de la incertidumbre aditiva en el error de seguimiento. Uno de los objetivos de este trabajo es estimar la incertidumbre utilizando el vector $\hat{\mathbf{E}}_n$ que representa una buena aproximación de \mathbf{E}_n , el cual es definido por una red neuronal (9); para lograr este objetivo se emplean los valores pasados de \mathbf{E}_n (ver Pantano et al. (2017))

$$\hat{\mathbf{E}}_n = \begin{pmatrix} \hat{E}_{x,n} \\ \hat{E}_{y,n} \\ \hat{E}_{\theta,n} \end{pmatrix} \quad (6)$$

Definiendo el error de modelado observado como:

$$\tilde{\mathbf{E}}_n = \mathbf{E}_n - \hat{\mathbf{E}}_n \quad (7)$$

Donde $\hat{\mathbf{E}}_n \in \mathbb{R}^{3 \times 1}$ es el vector de error de modelado estimado por el observador neuronal y $\tilde{\mathbf{E}}_n \in \mathbb{R}^{3 \times 1}$ es la diferencia de error entre el error real y el error de modelado estimado. El error de modelado se considera como una función no lineal desconocida de los errores de modelado pasados, siendo el principal objetivo de esta sección estimar esta función mediante una red neuronal. Ahora representando (6) como función de los pesos neuronales óptimos en una función neuronal

$$\mathbf{E}_n = f(\mathbf{E}_{n-1}, \mathbf{E}_{n-2}, \dots, \mathbf{E}_{n-p}) = -\Lambda \mathbf{E}_{n-1} + \mathbf{w}^{*T} \boldsymbol{\varphi}(\bar{\mathbf{E}}) + \varepsilon \quad (8)$$

Donde, $\bar{\mathbf{E}}$ es un vector de los valores anteriores de salida de \mathbf{E}_n , formado por $\bar{\mathbf{E}} = [\mathbf{E}_{n-1}, \mathbf{E}_{n-2}, \dots, \mathbf{E}_{n-p}]^T$, y $\mathbf{w} \in \mathbb{R}^{3 \times m}$ es una matriz de pesos óptimos y el vector $\boldsymbol{\varphi} \in \mathbb{R}^m$ son los vectores de función de base radial (RBF) vinculados al peso \mathbf{w}^* . Y ε es el error de aproximación neuronal. Dado que el vector de estado de salida se ve afectado por las perturbaciones del ruido, es necesario estimar sus valores. Un observador neuronal de estado basado en ec.(8) se determina de la siguiente manera:

$$\hat{\mathbf{E}}_n = -\Lambda \hat{\mathbf{E}}_{n-1} + \hat{\mathbf{w}}^T \boldsymbol{\varphi}(\hat{\mathbf{E}}) \quad (9)$$

Donde Λ es una ganancia matricial y es una matriz diagonal (será $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$), el parámetro $\hat{\mathbf{w}}$ es la matriz de ponderación neuronal obtenida de la estimación del error adaptable de aproximación $\hat{\mathbf{f}}(\hat{\mathbf{E}}, \hat{\mathbf{w}})$, y se puede expresar por:

$$\hat{\mathbf{f}}(\hat{\mathbf{E}}, \hat{\mathbf{w}}) = \hat{\mathbf{w}}^T \boldsymbol{\varphi}(\hat{\mathbf{E}}) \quad (10)$$

Donde $\hat{\mathbf{w}}_i$ son vectores i -fila La función gaussiana se utiliza como función de activación de cada neurona en el vector RBF $\boldsymbol{\varphi}$ (capa oculta de (9)).

$$\varphi_j(\hat{\mathbf{E}}) = \exp\left(-\frac{\|\hat{\mathbf{E}} - \mathbf{c}_j\|^2}{\mu_j^2}\right) \quad (11)$$

donde j es la j -ésima neurona de la capa oculta, \mathbf{c}_j es la posición central de la j -ésima neurona, y μ_j es el factor de dispersión de

la función gaussiana, y el regresor utilizado en la función gaussiana es: $\hat{\mathbf{E}} = [\hat{\mathbf{E}}_{n-1}, \dots, \hat{\mathbf{E}}_{n-p}, \mathbf{E}_{n-1}, \dots, \mathbf{E}_{n-p}]^T$ Para ajustar los parámetros neuronales se utiliza una regla de aprendizaje y se define como:

$$\Delta \hat{\mathbf{w}}_i = \gamma \tilde{E}_{n,i} \boldsymbol{\varphi}(\hat{\mathbf{E}}) \quad (12)$$

De acuerdo a lo visto anteriormente y teniendo en cuenta (7) en $(n + 1)$ y haciendo la diferencia entre (8) y (9) y sumando y restando el término $\mathbf{w}^{*T} \boldsymbol{\varphi}(\hat{\mathbf{E}})$, el error de modelado puede ser descrito por:

$$\begin{aligned} \tilde{\mathbf{E}}_n &= -\Lambda \mathbf{E}_{n-1} + \mathbf{w}^{*T} \boldsymbol{\varphi}(\bar{\mathbf{E}}_n) + \varepsilon - (-\Lambda \hat{\mathbf{E}}_{n-1} + \hat{\mathbf{w}}^T \boldsymbol{\varphi}(\hat{\mathbf{E}})) + \dots \\ &+ (\mathbf{w}^{*T} \boldsymbol{\varphi}(\hat{\mathbf{E}}_n) - \hat{\mathbf{w}}^T \boldsymbol{\varphi}(\hat{\mathbf{E}}_n)) = \dots \\ &= -\Lambda \tilde{\mathbf{E}}_{n-1} + \tilde{\mathbf{w}}^T \boldsymbol{\varphi}(\hat{\mathbf{E}}) + \bar{\varepsilon} \end{aligned} \quad (13)$$

donde $\bar{\varepsilon} = \mathbf{w}^{*T} (\boldsymbol{\varphi}(\bar{\mathbf{E}}) - \boldsymbol{\varphi}(\hat{\mathbf{E}}))$, ε es una función de una diferencia de función gaussiana $\boldsymbol{\varphi}$ y se puede considerar como un error de aproximación (incluido en $\bar{\varepsilon}$).

Consideración 1: el error de aproximación $\bar{\varepsilon}$ está limitado por una constante $\|\bar{\varepsilon}\| \leq \varepsilon_{max}$. Ahora expresando (13) en forma escalar.

$$\tilde{E}_{n,i} = -\lambda_i \tilde{E}_{n-1,i} + \tilde{\mathbf{w}}_i^T \boldsymbol{\varphi}(\hat{\mathbf{E}}) + \bar{\varepsilon}_i \quad (14)$$

El índice i denota el i -ésima fila de \mathbf{E}_n , para la realización de los experimentos se adoptara el valor de 0,1 para las constantes λ_i . Esta ecuación se utilizará para demostrar la estabilidad del estimador y se muestra en el apéndice.

3.2. Descripción de la estructura de la red neuronal RBF

Para la implementación del estimador se utilizó una red neuronal del tipo RBF debido a su capacidad como aproximador universal de funciones (Poggio y Girosi (1990)). Este tipo de redes neuronales utiliza funciones de base radial (gaussianas) como funciones de activación (11). Estas redes son especialmente efectivas para la aproximación de funciones no lineales y pueden manejar problemas de alta dimensionalidad sin los problemas de mínimos locales que a menudo afectan a otras arquitecturas de redes neuronales, como el perceptrón multicapa (MLP) (Gomez et al. (2023)). La arquitectura de las RBF es generalmente más simple, compuesta por tres capas: entrada, oculta y salida, como se observa en la Figura 2, lo que facilita en gran manera su implementación y entrenamiento (Yu et al.(2011)).

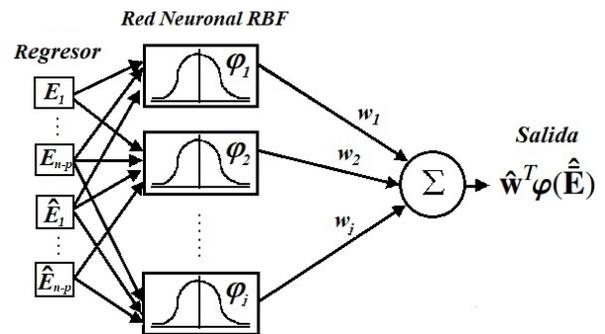


Figura 2: Estructura de la red neuronal del tipo RBF

Los pesos de la capa de salida están actualizados de forma online por la regla de ajuste indicada en la ecuación (12), que fue

obtenida respetando el criterio de estabilidad de Lyapunov (ver Apéndice). Para la construcción de este estimador se utilizaron cinco neuronas del tipo RBF cuyos resultados se muestran en la sección (5).

4. Controlador Propuesto

En esta sección se propone la formalización de la ley de control, donde básicamente es hallar el valor de las variables v_n y ω_n para que el robot móvil realice un seguimiento de trayectoria predeterminada. Las variables estimadas $\hat{x}_n, \hat{y}_n, \hat{\theta}_n$ obtenidas en (13) representan los valores estimados de las variables x_n, y_n, θ_n y se calculan de acuerdo con (15).

$$\begin{pmatrix} \hat{x}_n \\ \hat{y}_n \\ \hat{\theta}_n \end{pmatrix} = \begin{pmatrix} \hat{x}_{n-1} \\ \hat{y}_{n-1} \\ \hat{\theta}_{n-1} \end{pmatrix} + \tau_s \begin{pmatrix} v_{n-1} \cos(\theta_{n-1}) \\ v_{n-1} \sin(\theta_{n-1}) \\ \omega_{n-1} \end{pmatrix} \quad (15)$$

donde $\hat{x}_{n-1}, \hat{y}_{n-1}$ y $\hat{\theta}_{n-1}$ corresponden a la posición y orientación del robot en el momento $n-1$, and v_{n-1} y ω_{n-1} corresponden a los valores de las acciones de control calculados en el momento $n-1$. Es fácil demostrar que si $\nabla_n \mathbf{E}_n = 0$ y $\hat{\mathbf{E}}_n$ se calcula de acuerdo a (13) $\Rightarrow \mathbf{E}_n = \hat{\mathbf{E}}_n$. Incorporando la estimación de incertidumbres $\hat{\mathbf{E}}_n$ (calculado por (9)), se obtiene:

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \\ \theta_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \\ \theta_n \end{pmatrix} + \tau_s \begin{pmatrix} \cos(\theta_n) & 0 \\ \sin(\theta_n) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_n \\ \omega_n \end{pmatrix} + \underbrace{\begin{pmatrix} \hat{E}_{x,n} \\ \hat{E}_{y,n} \\ \hat{E}_{\theta,n} \end{pmatrix}}_{\hat{\mathbf{E}}_n} \quad (16)$$

Una vez calculada la estimación de las incertidumbres y a su vez, la trayectoria deseada $x_{d,n+1}$ y $y_{d,n+1}$ es conocida. Entonces las variables x_{n+1}, y_{n+1} y θ_{n+1} en (2) que en su lugar son reemplazadas por $x_{d,n+1}, y_{d,n+1}$ y $\theta_{ex,n+1}$, respectivamente, donde la variable θ_{ex} es obtenida a partir de (25) que representa el ángulo de orientación del robot móvil para que el error de seguimiento tienda a cero (ver Prueba del Teorema 1). De esta forma, para obtener un seguimiento de posición sin discontinuidades, el siguiente estado deseado $(x_{d,n+1}, y_{d,n+1}, \theta_{ex,n+1})$, no es necesariamente el nuevo valor de estado de referencia $(x_{ref,n+1}, y_{ref,n+1}, \theta_{ex,n+1})$. Estos nuevos estados son reemplazados asumiendo una aproximación proporcional al error.

$$\begin{pmatrix} x_{d,n+1} \\ y_{d,n+1} \\ \theta_{d,n+1} \end{pmatrix} = \begin{pmatrix} x_{ref,n+1} - k_v(x_{ref,n} - x_n) \\ y_{ref,n+1} - k_v(y_{ref,n} - y_n) \\ \theta_{ex,n+1} - k_\omega(\theta_{ex,n} - \theta_n) \end{pmatrix}; \quad (17)$$

$$\mathbf{e}_n = \begin{pmatrix} e_{x,n} \\ e_{y,n} \\ e_{\theta,n} \end{pmatrix} = \begin{pmatrix} x_{ref,n} - x_n \\ y_{ref,n} - y_n \\ \theta_{ref,n} - \theta_n \end{pmatrix}$$

Se define k_v y k_ω como parámetros de diseño de valores positivos menores a uno, ($0 < k_v < 1$) y ($0 < k_\omega < 1$). Y Teniendo en cuenta que:

- si $0 < k_v < 1$, el error debe decrecer, cuya tendencia es llegar a cero paulatinamente.
- si $k_v = 0$, ($x_{d,n+1} = x_{ref,n+1}$), debe alcanzar la trayectoria de referencia en un solo paso;

Por lo tanto, el enfoque propuesto en (17) se aplica para obtener una trayectoria sin discontinuidades. De la misma forma se aplica a $y_{d,n+1}$ y $\theta_{d,n+1}$, entonces definimos a:

$$\begin{pmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{pmatrix} = \begin{pmatrix} x_{ref,n+1} - k_v(x_{ref,n} - x_n) - x_n \\ y_{ref,n+1} - k_v(y_{ref,n} - y_n) - y_n \\ \theta_{ex,n+1} - k_\omega(\theta_{ex,n} - \theta_n) - \theta_n \end{pmatrix}; \quad (18)$$

$$\mathbf{\Gamma} = \begin{pmatrix} \cos(\theta_n) & 0 \\ \sin(\theta_n) & 0 \\ 0 & 1 \end{pmatrix}$$

Definición: La diferencia entre la referencia y la trayectoria real se denominará error de seguimiento, y esta definida por:

$$\begin{pmatrix} e_{x,n} \\ e_{y,n} \end{pmatrix} = \begin{pmatrix} x_{ref,n} - x_n \\ y_{ref,n} - y_n \end{pmatrix} \quad (19)$$

El error de seguimiento se define de la forma:

$$\|\mathbf{e}\| = \sqrt{e_{x,n}^2 + e_{y,n}^2} \quad (20)$$

El vector indicado por $(x_{n+1} - x_n, y_{n+1} - y_n, \theta_{n+1} - \theta_n)^T$ en (16) se sustituye por los elementos correspondientes según (18), y se obtiene el sistema indicado en (21)

$$\frac{1}{\tau_s} \left[\begin{pmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{pmatrix} - \hat{\mathbf{E}}_n \right] = \mathbf{\Gamma} \begin{pmatrix} v_n \\ \omega_n \end{pmatrix} \quad (21)$$

El sistema mostrado en la ecuación (21) posee tres ecuaciones y dos incógnitas. Y a su vez, las dos primeras filas en (21) se encuentran en función de una única variable v_n . Y para que (21) posea una solución única, el vector $\boldsymbol{\rho}$ debe estar contenido en el espacio columnas de $\mathbf{\Gamma} C(\mathbf{\Gamma})$; de esta forma, $\boldsymbol{\rho}$ debe ser una combinación lineal de las columnas de $\mathbf{\Gamma}$. Entonces, (22) las constantes reales d_1 y d_2 deben existir.

$$d_1 \begin{pmatrix} \cos(\theta_n) \\ \sin(\theta_n) \\ 0 \end{pmatrix} + d_2 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{pmatrix} - \hat{\mathbf{E}}_n; \quad d_1, d_2 \in \mathbf{R} \quad (22)$$

Separando las dos primeras filas es posible encontrar la condición a cumplir por orientación θ tal que el sistema (22) tiene solución exacta

$$\begin{cases} d_1 \cos(\theta_n) = \Delta x - \hat{E}_{x,n} \\ d_1 \sin(\theta_n) = \Delta y - \hat{E}_{y,n} \end{cases} \quad (23)$$

De la ecuación anterior se obtiene la siguiente igualdad,

$$\frac{\sin(\theta_n)}{\cos(\theta_n)} = \frac{\Delta y - \hat{E}_{y,n}}{\Delta x - \hat{E}_{x,n}} \quad (24)$$

Utilizando un calculo auxiliar para obtener las variables ϑ_y y ϑ_x que luego serán empleadas para determinar el ángulo de orientación, siendo:

$$\begin{aligned} \vartheta_y &= [y_{ref,n+1} - k_v(y_{ref,n} - y_n) - y_n - \hat{E}_{y,n}] \\ \vartheta_x &= [x_{ref,n+1} - k_v(x_{ref,n} - x_n) - x_n - \hat{E}_{x,n}] \end{aligned} \quad (25)$$

Para obtener una única solución, la dirección $\theta_{e_{x,n+1}}$ debe calcularse de la siguiente forma:

$$\theta_{e_{x,n+1}} = \tan^{-1} \left(\frac{\partial_y}{\partial_x} \right) \quad (26)$$

donde θ_{e_x} representa la orientación necesaria para que el robot móvil alcance y siga la trayectoria de referencia. Como se puede ver en comparación con los controladores de álgebra lineal anteriores (Scaglia et al.,(2020) ;Scaglia et al.,(2014)), el espacio columnas de la matriz γ se le agrega el vector de incertidumbres estimadas \hat{E}_n . Con esta nueva consideración, se pretende reducir de manera considerable los efectos indeseados del vector de incertidumbres sobre el error de trayectoria. Con este nuevo cálculo del ángulo de orientación deseado del robot, se alcanzará la trayectoria pre definida de manera correcta. Para definir la acción de control debe resolverse el sistema definido en (21), cuya solución óptima se obtiene aplicando el método de mínimos cuadrados, de la forma:

$$\Gamma^T \Gamma u = \Gamma^T \rho \quad (27)$$

Ahora si consideramos las ecuaciones (21), (25) y (26), la ley de control para el robot móvil se obtiene en (28)

$$\begin{cases} v_n = \frac{1}{\tau_s} [(\Delta y - \hat{E}_{y,n}) \sin(\theta_{e_{x,n}}) + (\Delta x - \hat{E}_{x,n}) \cos(\theta_{e_{x,n}})] \\ \omega_n = \frac{1}{\tau_s} (\Delta \theta - \hat{E}_{\theta,n}) \end{cases} \quad (28)$$

Teorema 1 El robot móvil descrito por (2) controlado por el enfoque desarrollado anteriormente conduce a un sistema asintóticamente estable, O sea, los valores de v_n and ω_n representan las acciones de control necesarias para que el error de seguimiento tienda a cero con n tendiendo a infinito. La prueba del **Teorema 1** y la convergencia a cero de los errores de estimación, de la red neuronal se muestran en el Apéndice.

5. Resultados Experimentales

Para demostrar la viabilidad de implementación del controlador propuesto, se propone que el robot siga diferentes trayectorias circulares de distinto diámetro, las cuales están unidas tangencialmente de tal forma que forman una trayectoria continua. el experimento es similar al realizado por Scaglia et al.(2020). La posición inicial del robot es el origen del sistema. En la prueba, se supone que existe una perturbación aditiva sobre las velocidades, donde la técnica propuesta debería mantener fielmente la trayectoria de referencia con el mínimo error de control. En la prueba se utilizará un control LABC sin estimador donde el robot debe seguir las trayectorias con perturbación actuando sobre las velocidades. Posteriormente se realizará la misma prueba con la técnica propuesta en este trabajo (LABC+NN) realizando las comparaciones entre ambos resultados. En la Figura 3, muestra la trayectoria de referencia y los resultados obtenidos con ambas propuestas. La Figura 6 presenta las acciones de control sobre el robot y el modelo, las acciones de control sobre el robot se encuentran afectada por una perturbación indeseada (línea continua). En cambio la que ingresa al modelo no se encuentra afectada perturbación (línea de trazo). Se puede observar que la perturbación actuante en el robot es creciente con

el tiempo y afecta de manera significativa la trayectoria seguida durante el experimento. En la Figura 4 muestra las gráficas del valor de los errores de seguimiento en las coordenadas x para ambos controles. De igual manera se muestra el error de seguimiento en las coordenadas y (Figura 5). Es posible apreciar que ambos errores permanecen acotados y cercanos a cero cuando el robot alcanza la trayectoria de referencia, donde la propuesta LABC+NN obtiene el menor error. De igual manera este resultado se ve reflejado en la Figura 7 donde se observa que el error acumulado se reduce drásticamente al emplear la técnica LABC con estimador neuronal en comparación con el método LABC clásico. En este caso se observa claramente el efecto de las incertidumbres en el error. Pero, el menor error se obtuvo por la combinación LABC y estimador neuronal propuesto en este trabajo, siendo el menos afectado por la perturbación aditiva y las posibles incertidumbres. También es más eficaz en el rechazo de perturbaciones y no produce error constante causado por ruido en los sensores o perturbaciones de todo tipo. Además, el controlador clásico es vulnerable a las variaciones de velocidad y a las posibles incertidumbres.

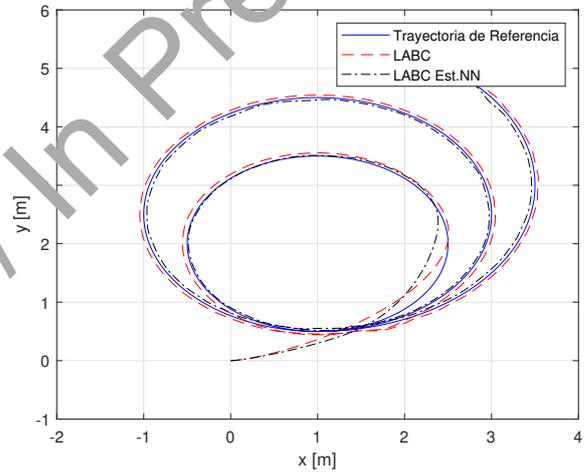


Figura 3: Trayectorias seguidas por el robot, línea de trazo indica el uso del controlador LABC, línea de trazo y punto uso de control LABC con estimador neuronal

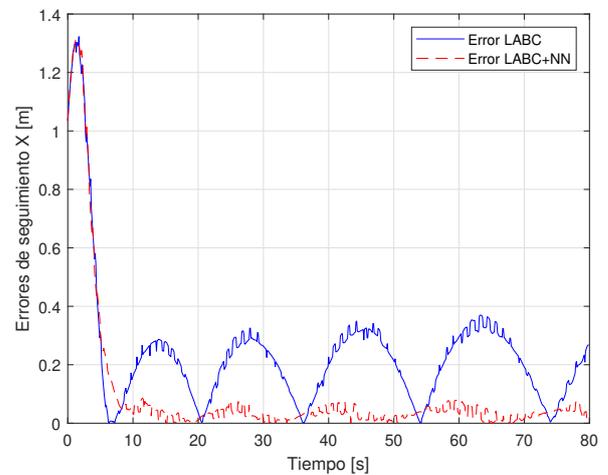


Figura 4: Error de trayectoria en la dirección del eje x , donde la línea continua indica el resultado del controlador LABC sin estimador, línea de trazo indica control LABC con estimador neuronal

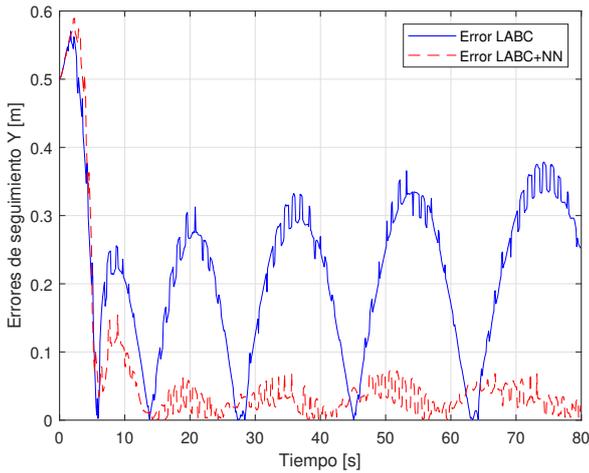


Figura 5: Error en la dirección del eje y , donde la línea continua indica el resultado del controlador LABC sin estimador, línea de trazo indica control LABC con estimador neuronal

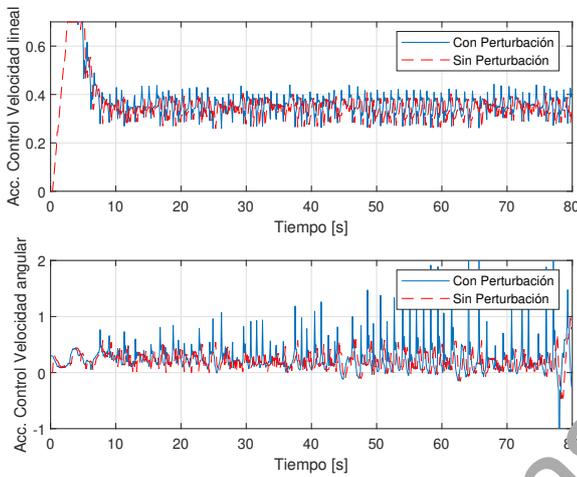


Figura 6: Acciones de Control, con perturbación (línea continua) y sin perturbación (línea de trazo)

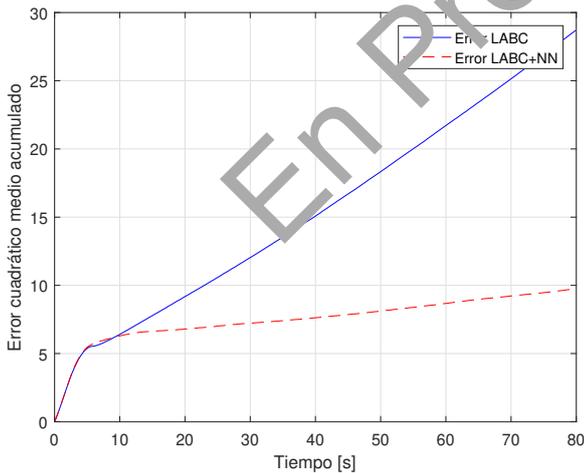


Figura 7: Error Acumulado durante el experimento, donde la línea de continua indica el uso del controlador LABC sin estimador, línea de trazo respuesta de control LABC con estimador neuronal

La técnica de control desarrollada en este trabajo, para un sistema robótico no lineal, no se necesita conocer las perturbaciones actuantes o el nivel de ruido en la odometría. Además, se sabe que una representación matemática de un modelo cinemático no describe con precisión el comportamiento real. Los efectos de estas no linealidades e incertidumbres, junto con las variaciones

en el sensado del robot demuestran la robustez de la propuesta empleada. Esta técnica de control inteligente puede ser considerada como una solución general para el control de sistemas no lineales y, en particular, para el caso de sistemas robóticos o cuando se necesita realizar seguimiento de trayectoria en entornos desconocidos. Este campo de investigación en control de robots sigue abierta a las cuestiones de modelado, la estabilidad matemática, la convergencia y el análisis de robustez de sistemas que continúan avanzando para la obtención de técnicas de control cada vez más precisas.

6. Conclusiones

En este trabajo se desarrolló una metodología basada en métodos numéricos y álgebra lineal para la creación de algoritmos de control destinados a robots móviles. Esta metodología se aplica especialmente en situaciones donde el sistema está representado por un modelo no lineal multivariable y presenta incertidumbre aditiva. Además, se ha introducido una red neuronal de estimación, que se ajusta en base a la hipótesis de variación de E_n . La estimación \hat{E}_n se ajusta considerando los valores pasados de E_n lo cual reduce la influencia del error de modelado en el error de seguimiento. El enfoque presentado reduce el efecto de las perturbaciones y los errores de modelado en el error de seguimiento independientemente de la complejidad del controlador y de las perturbaciones actuantes en el robot. Estos resultados muestran que el error de seguimiento disminuye cuando se realiza una buena estimación de \hat{E}_n . En comparación con otras leyes de control publicadas anteriormente (Blazic (2011); Damodaran et al. (2017); Farooq et al. (2014); Gu y Hu (2002); Lee et al. (2009); Park et al. (2016); Wang (2012); Ye (2008)). El controlador propuesto ofrece la ventaja de ser sencillo de diseñar y es altamente factible de implementar en aplicaciones prácticas. Así, el algoritmo se puede implementar directamente en el microcontrolador del robot sin necesidad de implementarlo en una computadora externa, ya que los cálculos son simples de realizar. Además, el método propuesto aquí no necesita una transformación del modelo en comparación con Blazic (2011). Además, nuestro enfoque no necesita desarrollar el modelo cinemático incierto analítico en comparación con la metodología propuesta en Das y Kar (2006) y Martins et al. (2008). Las características mencionadas hacen que la metodología propuesta se pueda aplicar a muchos sistemas no lineales, convirtiéndola en una técnica prometedora para su aplicación en varios sistemas de la industria.

Agradecimientos

Este trabajo ha sido realizado gracias al apoyo del Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET).

Referencias

- Acosta, F. U. V.-S., 2004. Diseño de un sistema telemando y telemétrico experimental para una aerostación de baja altura. Researchgate.
- Andaluz Ortiz, G. M., 2011. Modelación, identificación y control de robots móviles. B.S. thesis, QUITO/EPN/2011.
- Blazic, S., 2011. A novel trajectory tracking control law for wheeled mobile robots. Robotics and Autonomous Systems 59 (11), 1001-1007.
- Chwa, D., 2004. Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates. IEEE Transactions on Control Systems Technology 12 (4), 637-644. DOI: 10.1109/TCST.2004.824953

- Damodaran, S., Kumar, T. S., Sudheer, A. P., 2017. Design and implementation of ga tuned pid controller for desired interaction and trajectory tracking of wheeled mobile robot. En: Proceedings of the Advances in Robotics. pp.1-6.
- Das, T., Kar, I. N., 2006. Design and implementation of an adaptive fuzzy logic based controller for wheeled mobile robots. IEEE Transactions on Control systems technology 14 (3), 501-510.
- De La Cruz, C., Bastos, T. F., Carelli, R., 2011. Adaptive motion control law of a robotic wheelchair. Control Engineering Practice 19 (2), 113-125.
- Deng, R., Zhang, Q., Gao, R., Li, M., Liang, P., Gao, X., 2021. A trajectory tracking control algorithm of nonholonomic wheeled mobile robot. En: 2021 6th IEEE International Conference on Advanced Robotics and Mechatronics(ICARM). pp. 823-828. DOI:10.1109/ICARM52023.2021.9536154
- Fareh, R., Khadraoui, S., Abdallah, M. Y., Baziyad, M., Bettayeb, M., 2021. Active disturbance rejection control for robotic systems: A review. Mechatronics 80, 102671.
- Farooq, U., Hasan, K., Hanif, A., Amar, M., Asad, M. U., 2014. Fuzzy logic based path tracking controller for wheeled mobile robots. International Journal of Computer and Electrical Engineering 6 (2), 77.
- Gandolfo, D., Rosales, C., Patiño, D., Scaglia, G., Jordan, M., 2014. Trajectory tracking control of a pvtol aircraft based on linear algebra theory. Asian Journal of Control 16 (6), 1849-1858.
- Gomez, J., Rossomando, F., Capraro, F., Soria, C., 2023. Real-time neuroadaptive pi control of soil moisture by using a hybrid model. Revista Iberoamericana de Automática e Informática industrial 20 (1), 93-103.
- Gu, D., Hu, H., 2002. Neural predictive control for a car-like mobile robot. Robotics and Autonomous Systems 39 (2), 73-86.
- Hassan, N., Saleem, A., 2021. Analysis of trajectory tracking control algorithms for wheeled mobile robots. En: 2021 IEEE Industrial Electronics and Applications Conference (IEACon). pp.236-241. DOI: 10.1109/IEA-Con51066.2021.9654675
- Jung, S., 2012. Experiences in developing an experimental robotics course program for undergraduate education. IEEE Transactions on Education 56 (1), 129-136.
- Lee, J. H., Lin, C., Lim, H., Lee, J. M., 2009. Sliding mode control for trajectory tracking of mobile robot in the rfid sensor space. International Journal of Control, Automation and Systems 7, 429-435.
- López-Cortés, L. F., Lozano-Hernandez, Y., Torres, L., Guerrero-Castellanos, J. F., Aguirre-Anaya, J. A., 2021. Comparison of dynamic model-based control algorithms for trajectory tracking in an omnidirectional robot. En: 2021 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE). pp. 60-65. DOI:10.1109/ICMEAE55138.2021.96001
- Martins, F. N., Celeste, W. C., Carelli, R., Sarcinelli-Filho, M., Bastos, F., Neto, T. F., 2008. An adaptive dynamic controller for autonomous mobile robot trajectory tracking. Control Engineering Practice 16 (11), 1334-1333.
- Normey-Rico, J. E., Alcalá, I., Gómez-Ortega, J., Camacho, E. J., 2001. Mobile robot path tracking using a robust pid controller. Control Engineering Practice 9 (11), 1209-1214.
- Pantano, M. N., Serrano, M. E., Fernández, M. C., Rossomando, F. G., Ortiz, O. A., Scaglia, G. J., 2017. Multivariable control for tracking optimal profiles in a nonlinear fed-batch bioprocess integrated with state estimation. Industrial and Engineering Chemistry Research 56 (24), 6043-6056.
- Park, J.-W., Kwak, H.-J., Kang, I.-C., Kim, D.W., et al., 2016. Advanced fuzzy potential field method for mobile robot obstacle avoidance. Computational intelligence and neuroscience 2016.
- Poggio, T., Girosi, F., 1990. Networks for approximation and learning. Proceedings of the IEEE 78 (9), 1481-1497.
- Rosales, A., Scaglia, G., Mut, V., di Sciascio, F., 2011. Formation control and trajectory tracking of mobile robotic systems-a linear algebra approach. Robotica 29 (3), 335-349.
- Rosales, C., Soria, C. M., Rossomando, F. G., 2019. Identification and adaptive pid control of a hexacopter uav based on neural networks. International journal of Adaptive control and signal processing 33 (1), 74-91.
- Rossomando, F. G., Soria, C., Carelli, R., 2012. Neural network-based compensation control of mobile robots with partially known structure. IET Control Theory and Applications 6 (12), 1851-1860.
- Rossomando, F. G., Soria, C., Carelli, R., 2014. Sliding mode neuro adaptive control in trajectory tracking for mobile robots. Journal of Intelligent and Robotic Systems 74, 931-944.
- Scaglia, G., Aballay, P. M., Serrano, M. E., Ortiz, O. A., Jordan, M., Vallejo, M. D., 2014. Linear algebra based controller design applied to a bench-scale oenological alcoholic fermentation. Control Engineering Practice 25, 66-74.
- Scaglia, G., Montoya, L. Q., Mut, V., di Sciascio, F., 2009. Numerical methods based controller design for mobile robots. Robotica 27 (2), 269-279.
- Scaglia, G. J. E., Serrano, M. E., Godoy, S. A., Rossomando, F., 2020. Linear algebra-based controller for trajectory tracking in mobile robots with additive uncertainties estimation. IMA Journal of Mathematical Control and Information 37 (2), 607-624.
- Serrano, M. E., Godoy, S. A., Mut, V. A., Ortiz, O. A., Scaglia, G. J., 2016. A nonlinear trajectory tracking controller for mobile robots with velocity limitation via parameters regulation. Robotica 34 (11), 2546-2565.
- Serrano, M. E., Godoy, S. A., Romoli, S., Scaglia, G. J., 2017. A numerical approximation-based controller for mobile robots with velocity limitation. Asian Journal of Control 19 (6), 2165-2177.
- Serrano, M. E., Scaglia, G. J., Godoy, S. A., Mut, V., Ortiz, O. A., 2013. Trajectory tracking of underactuated surface vessels: A linear algebra approach. IEEE Transactions on Control Systems Technology 22 (3), 1103-1111.
- Sira-Ramírez, H., López-Urbe, C., Velasco-Villa, M., 2013. Linear observer-based active disturbance rejection control of the omnidirectional mobile robot. Asian Journal of Control 15 (1), 51-63.
- Sun, Z., Xia, Y., Dai, L., Liu, K., Ma, D., 2017. Disturbance rejection mpc for tracking of wheeled mobile robot. IEEE/ASME Transactions On Mechatronics 22 (6), 2576-2587.
- Tian, Y., Sarkar, N., 2014. Control of a mobile robot subject to wheel slip. Journal of Intelligent and Robotic Systems 74, 915-929.
- Tzafestas, S. G., 2013. Introduction to mobile robot control. Elsevier.
- Wang, K., 2012. Near-optimal tracking control of a nonholonomic mobile robot with uncertainties. International Journal of Advanced Robotic Systems 9 (3), 66.
- Wit, C. C. d., Khenouf, H., Samson, C., Jordalen, O. J., 1993. Nonlinear control design for mobile robots. En: Recent trends in mobile robots. World Scientific, pp. 121-156.
- Wu, Q., Qi, J., Wu, C., Yang, M., 2020. Design of ugv trajectory tracking controller in ugv bay cooperation. En: 2020 39th Chinese Control Conference (CCC). pp. 3685-3694. DOI: 10.23919/CCC50068.2020.9188688
- Yang, C., Ma, H., Fu, M., Yang, C., Ma, H., Fu, M., 2016. Robot kinematics and dynamics modeling. Advanced technologies in modern robotic applications, 2, 48.
- Yang, Y., Yan, X., Sirlantzis, K., Howells, G., 2019. Application of sliding mode trajectory tracking control design for two-wheeled mobile robots. En: 2019 NASA/ESA Conference on Adaptive Hardware and Systems (AHS).pp. 109-114.DOI: 10.1109/AHS.2019.00012
- Ye, J., 2008a. Adaptive control of nonlinear pid-based analog neural networks for a nonholonomic mobile robot. Neurocomputing 71 (7-9), 1561-1565.
- Ye, J., 2008b. Tracking control for nonholonomic mobile robots: Integrating the analog neural network into the backstepping technique. Neurocomputing 71 (16-18), 3373-3378.
- Yu, H., Xie, T., Paszczynski, S., Wilamowski, B. M., 2011. Advantages of radial basis function networks for dynamic system design. IEEE Transactions on Industrial Electronics 58 (12), 5438-5450.
- Yue, M., Tang, F., Liu, B., Yao, B., 2012. Trajectory-tracking control of a nonholonomic mobile robot: Backstepping kinematics into dynamics with uncertain disturbances. Applied Artificial Intelligence 26 (10), 952-966.
- Zhang, L., Wang, B., Guan, E., Liu, X., Saqib, M., Zhao, Y., 2024. Adaptive skid-steering control approach for robots on uncertain inclined planes with redundant load-bearing mobility. Biomimetics 9 (2), 64.
- Zhang, M., Miao, Z., Li, N., 2022. Research on trajectory tracking control of wheeled mobile robot by ilc. En: 2022 3rd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE). pp. 33-36. DOI:10.1109/ICBAIE56435.2022.9985861

Apéndice A. Análisis de Estabilidad

Teorema 1: Considerando el sistema definido por (2), y el modelo estimador (9). El observador de errores neuronales basado en funciones RBF es SQUUB Prueba: Una función candidata de Laypunov $\mathcal{L}_{n,i}$ función definida positiva se define por:

$$\mathcal{L}_{n,i} = \tilde{E}_{n,i}^2 + \gamma^{-1}(\mathbf{w}_{n-1,i}^T \tilde{\mathbf{w}}_{n-1,i}) \quad (A.1)$$

Ahora, haciendo la diferencia en tiempo discreto, $\Delta\mathcal{L}_{n,i}$ se escribe de la siguiente manera:

$$\begin{aligned} \Delta\mathcal{L}_{n,i} &= \mathcal{L}_{n+1,i} - \mathcal{L}_{n,i} = \dots \\ &= \tilde{E}_{n+1,i}^2 - \tilde{E}_{n,i}^2 + \gamma^{-1}(\tilde{\mathbf{w}}_{n,i}^T \tilde{\mathbf{w}}_{n,i} - \tilde{\mathbf{w}}_{n-1,i}^T \tilde{\mathbf{w}}_{n-1,i}) \end{aligned} \quad (A.2)$$

Definiendo el segundo término de la ecuación anterior como:

$$\Delta \mathbf{w} = \gamma^{-1}(\tilde{\mathbf{w}}_{n,i}^T \tilde{\mathbf{w}}_{n,i} - \tilde{\mathbf{w}}_{n-1,i}^T \tilde{\mathbf{w}}_{n-1,i}) \quad (\text{A.3})$$

Organizando (A.2) y considerando que $\Delta \tilde{E}_{n,i} = \tilde{E}_{n+1,i} - \tilde{E}_{n,i}$, obtenemos:

$$\begin{aligned} \Delta \mathcal{L}_{n,i} &= \tilde{E}_{n+1,i}^2 - \tilde{E}_{n,i}^2 + \Delta \mathbf{w} = \dots \\ &= (\tilde{E}_{n,i} + \Delta \tilde{E}_{n,i})^2 - \tilde{E}_{n,i}^2 + \Delta \mathbf{w} = \dots \\ &= 2\tilde{E}_{n,i} \Delta \tilde{E}_{n,i} + \Delta \tilde{E}_{n,i}^2 + \Delta \mathbf{w} \end{aligned} \quad (\text{A.4})$$

Entonces, reemplazando (14) en (A.4), $\Delta \mathcal{L}_{n,i}$ se convierte en:

$$\begin{aligned} \Delta \mathcal{L}_{n,i} &= 2\tilde{E}_{n,i}(\tilde{E}_{n+1,i} - \tilde{E}_{n,i}) + \Delta \tilde{E}_{n,i}^2 + \Delta \mathbf{w} = \dots \\ &= 2\tilde{E}_{n,i}(\tilde{\omega}_i^T \boldsymbol{\varphi}(\hat{\mathbf{E}}) - \lambda_i \tilde{E}_{n,i} + \bar{\varepsilon}_i - \tilde{E}_{n,i}) + \Delta \tilde{E}_{n,i}^2 + \Delta \mathbf{w} \end{aligned} \quad (\text{A.5})$$

Si ahora consideramos $\Delta \tilde{\mathbf{w}}_{n,i} = \tilde{\mathbf{w}}_{n,i} - \tilde{\mathbf{w}}_{n-1,i}$, nosotros reescribimos $\Delta \mathbf{w}$ por conveniencia se expresa como:

$$\begin{aligned} \Delta \mathbf{w} &= \gamma^{-1}(\tilde{\mathbf{w}}_{n,i}^T \tilde{\mathbf{w}}_{n,i} - (\tilde{\mathbf{w}}_{n,i} - \Delta \tilde{\mathbf{w}}_{n,i})^T (\tilde{\mathbf{w}}_{n,i} - \Delta \tilde{\mathbf{w}}_{n,i})) \\ &= 2\gamma^{-1}(\tilde{\mathbf{w}}_{n,i}^T \Delta \tilde{\mathbf{w}}_{n,i}) - \gamma^{-1}(\Delta \tilde{\mathbf{w}}_{n,i}^T \Delta \tilde{\mathbf{w}}_{n,i}) \end{aligned} \quad (\text{A.6})$$

Entonces, (A.6) se reemplaza en (A.5) donde, reordenando los términos, obtenemos:

$$\begin{aligned} \Delta \mathcal{L}_{n,i} &= 2\tilde{E}_{n,i}(\tilde{\mathbf{w}}_{n,i}^T \boldsymbol{\varphi}(\hat{\mathbf{E}}) - \lambda_i \tilde{E}_{n,i} + \bar{\varepsilon}_i - \tilde{E}_{n,i}) + \Delta \tilde{E}_{n,i}^2 + \\ &+ 2\gamma^{-1}(\tilde{\mathbf{w}}_{n,i}^T \Delta \tilde{\mathbf{w}}_{n,i}) - \gamma^{-1}(\Delta \tilde{\mathbf{w}}_{n,i}^T \Delta \tilde{\mathbf{w}}_{n,i}) = \dots \\ &= (2\tilde{\mathbf{w}}_{n,i}^T [\tilde{E}_{n,i} \boldsymbol{\varphi}(\hat{\mathbf{E}}) + \gamma^{-1} \Delta \tilde{\mathbf{w}}_{n,i}]) - \dots \\ &- 2\lambda_i \tilde{E}_{n,i} + 2\tilde{E}_{n,i} \bar{\varepsilon}_i - 2\tilde{E}_{n,i}^2 + \Delta \tilde{E}_{n,i}^2 - \gamma^{-1}(\Delta \tilde{\mathbf{w}}_{n,i}^T \Delta \tilde{\mathbf{w}}_{n,i}) \end{aligned} \quad (\text{A.7})$$

Sustituyendo la ley de ajuste (12) $\Delta \tilde{\mathbf{w}}$, en (A.7), $\Delta \mathcal{L}_{n,i}$ queda como:

$$\begin{aligned} \Delta \mathcal{L}_{n,i} &= -2(\lambda_i + 1)\tilde{E}_{n,i}^2 + 2\tilde{E}_{n,i} \bar{\varepsilon}_i + \Delta \tilde{E}_{n,i}^2 - \dots \\ &- \gamma^{-1}(\Delta \tilde{\mathbf{w}}_{n,i}^T \Delta \tilde{\mathbf{w}}_{n,i}) \end{aligned} \quad (\text{A.8})$$

En (A.8) el incremento del modelo del error $\Delta \tilde{E}_{n,i}$ es desconocido, y se puede aproximar mediante la siguiente ecuación:

$$\Delta \tilde{E}_{n,i} = \left(\frac{\partial \tilde{E}_{n+1,i}}{\partial \tilde{\mathbf{w}}_{n,i}} \right)^T \Delta \tilde{\mathbf{w}}_{n,i} \quad (\text{A.9})$$

La derivada parcial del modelo de error depende solo de los pesos de la red neuronal y se puede reescribir como:

$$\Delta \tilde{E}_{n,i} = -\left(\frac{\partial \hat{E}_{n+1,i}}{\partial \hat{\mathbf{w}}_{n,i}} \right)^T \Delta \tilde{\mathbf{w}}_{n,i} \quad (\text{A.10})$$

Cambiar los valores de los pesos según (12), y considerando que \mathbf{w}^* es constante, los vectores de peso ideal se requieren solo con fines analíticos. reorganizando, $\Delta \hat{\mathbf{w}}_{n,i}$ se puede escribir como:

$$\begin{aligned} \Delta \hat{\mathbf{w}}_{n,i} &= \hat{\mathbf{w}}_{n,i} - \hat{\mathbf{w}}_{n-1,i} = (\mathbf{w}^* - \hat{\mathbf{w}}_{n,i}) - (\mathbf{w}^* - \hat{\mathbf{w}}_{n-1,i}) \\ &= -\hat{\mathbf{w}}_{n,i} + \hat{\mathbf{w}}_{n-1,i} = -\Delta \hat{\mathbf{w}}_{n,i} \end{aligned} \quad (\text{A.11})$$

Entonces, el incremento del modelo de error (A.9) se puede escribir como:

$$\Delta \tilde{E}_{n,i} = -\left(\frac{\partial \hat{E}_{n+1,i}}{\partial \hat{\mathbf{w}}_{n,i}} \right)^T (\gamma \tilde{E}_{n,i} \boldsymbol{\varphi}(\hat{\mathbf{E}})) \quad (\text{A.12})$$

Considerando el valor de la derivada parcial de la red neuronal (9) y reemplazándolo en (A.12), nosotros llegamos a:

$$\Delta \tilde{E}_{n,i} = -(\gamma \tilde{E}_{n,i} \boldsymbol{\varphi}(\hat{\mathbf{E}})^T \boldsymbol{\varphi}(\hat{\mathbf{E}})) \quad (\text{A.13})$$

El aumento en el error del modelo se define como:

$$\Delta \tilde{E}_{n,i} = -\tilde{\gamma} |\tilde{E}_{n,i}| \quad (\text{A.14})$$

Donde

$$\tilde{\gamma} = \gamma \|\boldsymbol{\varphi}(\hat{\mathbf{E}})\|^2 \quad (\text{A.15})$$

con el vector de funciones RBF, $\boldsymbol{\varphi}(\hat{\mathbf{E}})$, siendo una función acotada, i.e., $\|\boldsymbol{\varphi}(\hat{\mathbf{E}})\| \leq k$. El valor $\tilde{\gamma}$ es un factor de aprendizaje de la red neuronal $0 < \tilde{\gamma} < 1$ y puede definirse arbitrariamente. Expresando ecuación (A.14) en forma vectorial:

$$|\Delta \tilde{E}_{n,i}| \leq \tilde{\gamma} |\tilde{E}_{n,i}| \quad (\text{A.16})$$

Sustituyendo el valor del incremento del error del modelo en (A.8), la diferencia discreta de Lyapunov se define como:

$$\Delta \mathcal{L}_{n,i} = -2(\lambda_i + 1)\tilde{E}_{n,i}^2 + 2\tilde{E}_{n,i} \bar{\varepsilon}_i + \tilde{\gamma}^2 \tilde{E}_{n,i}^2 - \gamma^{-1}(\Delta \tilde{\mathbf{w}}_{n,i}^T \Delta \tilde{\mathbf{w}}_{n,i})$$

Reemplazando la regla de aprendizaje en la ecuación anterior y aplicando la regla, se obtiene:

$$\begin{aligned} \Delta \mathcal{L}_{n,i} &= -2(\lambda_i + 1)\tilde{E}_{n,i}^2 + 2\tilde{E}_{n,i} \bar{\varepsilon}_i + \tilde{\gamma}^2 \tilde{E}_{n,i}^2 - \dots \\ &- \gamma^{-1}(\gamma \tilde{E}_{n,i} \boldsymbol{\varphi}(\hat{\mathbf{E}}))^T (\gamma \tilde{E}_{n,i} \boldsymbol{\varphi}(\hat{\mathbf{E}})) \leq \dots \\ &\leq -2(\lambda_i + 1)|\tilde{E}_{n,i}|^2 + 2|\tilde{E}_{n,i}||\bar{\varepsilon}_i| + \tilde{\gamma}^2 |\tilde{E}_{n,i}|^2 - \dots \\ &- \gamma \|\boldsymbol{\varphi}(\hat{\mathbf{E}})\|^2 |\tilde{E}_{n,i}|^2 \leq \dots \\ &\leq -2(\lambda_i + 1)|\tilde{E}_{n,i}|^2 + 2|\tilde{E}_{n,i}||\bar{\varepsilon}_i| + \tilde{\gamma}^2 |\tilde{E}_{n,i}|^2 - \dots \\ &- \tilde{\gamma} |\tilde{E}_{n,i}|^2 \leq \dots \\ &\leq -2\left(\lambda_i - \frac{(1-\tilde{\gamma})\tilde{\gamma}}{2} + 1\right) |\tilde{E}_{n,i}|^2 + 2|\tilde{E}_{n,i}||\bar{\varepsilon}_i| \end{aligned} \quad (\text{A.17})$$

de la ecuación anterior podemos obtener la condición de estabilidad

$$\begin{aligned} \Delta \mathcal{L}_{n,i} &\leq -2\left(\lambda_i - \frac{(1-\tilde{\gamma})\tilde{\gamma}}{2} + 1\right) |\tilde{E}_{n,i}|^2 + \dots \\ &+ 2|\tilde{E}_{n,i}||\bar{\varepsilon}_i| \leq 0 \end{aligned} \quad (\text{A.18})$$

Teniendo en cuenta la Observación 1 se llega a:

$$|\tilde{E}_{n,i}| \geq \frac{\varepsilon_{max}}{\left(\lambda_i - \frac{(1-\tilde{\gamma})\tilde{\gamma}}{2} + 1\right)} \quad (\text{A.19})$$

La ecuación (A.19) demuestra que la estimación neuronal del sistema no lineal es semiglobal, finalmente uniformemente acotada (SGUUB) y es estable y que también se puede utilizar para construir una ley de control, que se ilustrará en la siguiente sección.