Recibido 30 Nov. 2022 *ReCIBE, Año 12 No.1, May. 2023*

Aceptado 01 Dec. 2022

Control Neuro Proporcional-Derivativo para la formación Líder-Seguidor entrenado con el Filtro de Kalman Extendido

Neuro Proportional-Derivative Control for Leader-Follower Formation Trained with the Extended Kalman Filter

Dr. José de Jesús Hernández Barragán[1]

Dra. Alma Yolanda Alanís García[1]

Dr. Erasmo Gabriel Martínez Soltero[1]

Dr. Jorge Daniel Ríos Arrañaga[1]

[1]Departamento de Innovación Basada en la Información y el Conocimiento, Centro Universitario de Ciencias Exactas e Ingenierías, Universidad de Guadalajara, Guadalajara 44430, México

Resumen: Este trabajo presenta un esquema control neuro adaptativo para la formación Líder-Seguidor de robots móviles diferenciales. El esquema está basado en un control neuro proporcional derivativo entrenado con filtro de Kalman extendido. El controlador propuesto ofrece un ajuste de ganancias adaptativo en línea, lo cual es ideal para lidiar con problemas de perturbaciones, ruido y dinámicas no modeladas. El esquema propuesto es implementado a nivel simulación y experimental usando robots Turtlebot3® y la plataforma Sistema Operativo Robótico (ROS). Además, el desempeño del controlador propuesto es comparado con un controlador para la formación Líder-Seguidor del estado del arte. Los resultados muestran que la propuesta de control tiene un mejor tiempo de convergencia, menor error de formación y menor error de seguimiento.

Palabras clave: Formación, Control PD Adaptativo, Filtro de Kalman Extendido, Robot móvil, Turtlebot3.

1. Introducción

La interacción entre un sistema robótico con otros sistemas robóticos o humanos de manera cooperativa para realizar una tarea es crucial para resolver problemas de vigilancia, exploración, reconocimiento, monitoreo ambiental, y manipulación cooperativa. Esta metodología es nombrada sistemas de robots de red (Sanfeliu, Hagita , & Saffiotti, 2008), (Kamel, Yu, & Zhang, 2020), (Wang, y otros, 2016). Realizar tareas con más de un equipo trae ventajas como: multitarea, tolerancia a fallas, rentabilidad, flexibilidad y distribución. Un sistema cooperativo multi agente puede estar compuesto por robots que operan coordinados para realizar una tarea. La técnica que garantiza la cooperación entre robots se denomina control cooperativo. El control de formación para múltiples agentes se considera un caso especial de control cooperativo (Kamel, Yu, & Zhang, 2020). El objetivo del control para la formación es mantener cierto patrón o forma durante la tarea de ejecución. Este patrón define una formación, que prácticamente son distancias relativas entre robots (Oh, Park, & Ahn, 2015). Existe una gran variedad de estrategias para el control de formación. Entre las más comunes se encuentran métodos basados en estructuras virtuales, esquemas de Líder-Seguidor, basados en grafos, y campos de potencial artificial (Kamel, Yu, & Zhang, 2020). En el esquema de Líder-Seguidor, uno de los robots se asigna como líder, mientras que los otros robots son seguidores (Wang, y otros, 2016). La ley de control se aplica a cada seguidor de manera independiente al líder. El propósito es que el robot seguidor mantenga una posición de formación con respecto al líder. En este trabajo, la estrategia de control de formación propuesta se enfoca en el esquema de Líder-Seguidor.

A continuación, mencionamos algunas de las estrategias del estado del arte sobre el control para la formación Líder-Seguidor. En (Lu, et al., 2019), se introduce un controlador que lidia con limitaciones de distancia en la comunicación entre robots móviles diferenciales. Luego en (Xuan-Mung & Hong, 2019), se propone una técnica de control robusta ante la presencia de perturbaciones, durante la tarea para la formación de drones. Después en (Hassan & Hammuda, 2020), se propone un esquema de control con base en observadores que permiten estimar el estado de los robots con alta precisión, lo que mejora la estrategia de control. Por otra parte, en (Dai, He, Chen, & Jin, 2020), se propone una estrategia de control distribuido utilizando solo información local entre los robots. Esta estrategia involucra un modelo dinámico para la formación. Finalmente, se propone un esquema de control para la formación de robots móviles tipo coche, especialmente en el caso del modelo Ackerman (Alfaro & Morán, 2020). Aunque estas estrategias demuestran su capacidad para lograr y mantener una formación, muchos de los trabajos reportan solo resultados de simulación. En una aplicación real, se deben tomar en consideración perturbaciones externas, ruido de medición y dinámicas no modeladas. El desempeño de estas estrategias se puede ver afectado en la práctica. Esto implica un ajuste de ganancias experimentales para mejorar el desempeño de las estrategias de control. Para resolver este

inconveniente, en este trabajo se plantea una estrategia de control adaptativo para la formación Líder-Seguidor.

Por otro lado, en la actualidad la presencia e importancia de la inteligencia artificial no se puede ignorar, ha permitido un avance en la sociedad en relativamente poco tiempo y sigue avanzando al encontrarse en áreas clave de la industria, la educación, la ciencia y el entretenimiento (Bryson, 2019). Día a día se proponen soluciones basadas en las diversas metodologías de inteligencia artificial, dentro de control de sistemas, sin duda a pesar de sus desventajas conocidas los controladores P, PD, y PID siguen siendo de los controladores más utilizados. Los controladores PID neuronales proponen metodologías para mejorar su desempeño del PID clásico. Existen técnicas de PID adaptativas entrenadas con base en gradientes (Hernandez-Alvarado, Garcia-Valdovinos, Salgado-Jimenez, Gómez-Espinosa, & Fonseca-Navarro, 2016), (Tang, Wang, Gu, & Gu, 2020), (Zhong, Zhu, Zhao, Han, & Zhang, 2020). Aunque se ha demostrado que los controladores PID entrenados con el Filtro de Kalman Extendido muestran superioridad con respecto a tasas de aprendizaje y tiempos de convergencias más rápidos (Hernandez-Barragan, et al., 2020), (Hernandez-Barragan, et al., 2021). En este artículo, la estrategia de control adaptativo para la formación Líder-Seguidor se basa en un controlador PD de una sola neurona, entrenada con el filtro de Kalman extendido.

El aporte de este trabajo está en el diseño de un esquema control neuro adaptativo para la formación Líder-Seguidor de robots móviles diferenciales. El esquema de control propuesto se basa en un controlador neuro proporcional derivativo (NPD, por sus siglas en inglés, *Neural Proporcional-Derivative*) que es entrenado con el filtro de Kalman extendido. La aportación principal del controlador NPD es evitar el ajuste de ganancias del controlador, ante perturbaciones exteriores, ruido y dinámicas no modeladas, especialmente para el caso de una aplicación práctica. Además, el controlador NPD se inspira y compara con un controlador del estado del arte para la formación Líder-Seguidor, tanto en pruebas de simulación como en pruebas de experimentación.

El trabajo está organizado de la siguiente manera: en la sección 2, se presenta el modelo cinemático del robot móvil diferencial, la cinemática para la formación Líder-Seguidor y la estrategia de control que toma como base el diseño del controlador propuesto NPD. Después, en la sección 3 se describe el diseño del controlador NPD, desde el diseño de la neurona y su entrenamiento, hasta la descripción del controlador adaptativo para la formación Líder-Seguidor. En la sección 4 se presentan los resultados de la simulación y la experimentación. La discusión sobre los resultados se muestra en la sección 5. Finalmente, la conclusión y el trabajo futuro se presenta en la sección 6.

2. Preliminares

2.1 Modelo cinemático del robot móvil

Los robots de tipo diferencial son equipos robóticos ampliamente utilizados en la industria junto con la metodología de sistema de robots en red en las áreas mencionadas en el párrafo anterior, estos equipos se caracterizan por contar con dos ruedas que son motorizadas de manera independiente, su tracción se ve afectado por el tipo de ruedas y terreno, por lo que se puede contar con diferentes modelos de robots en esta clasificación, como los modelo tipo oruga y el de dos ruedas (Siegwart, Nourbakhsh, & Scaramuzza, 2011), (Ben-Ari & Mondada, 2017).

El diagrama del robot móvil diferencial considerado en este trabajo se muestra en la Figura 1, donde la posición (x, y) y la orientación θ del robot están expresados en el marco de referencia global $\{o\}$. El marco de referencia local $\{b\}$ esta adherido a la plataforma móvil, justo en el centro de las ruedas. Además, v es una velocidad lineal y w una velocidad angular, ambas definidas en el marco de referencia local $\{b\}$.

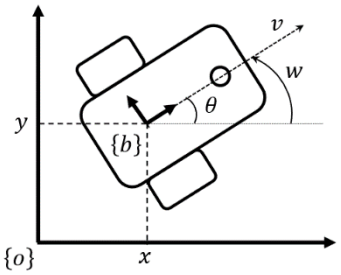

Figura 1: Diagrama del robot móvil diferencial.

El modelo cinemático del robot móvil está definido en (*1*), donde las velocidades $(\dot{x}, \dot{y}, \dot{\theta})$ describen el movimiento del robot con respecto al marco de referencia global $\{o\}$. Este modelo expresa la cinemática del robot monociclo donde las velocidades (v, w) definen las variables de control (Klancar, Zdesar, Blazic, & Skrjanc, 2017). En este trabajo, se considera controlar el robot móvil diferencial con las variables de control del modelo monociclo.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{1}$$

2.2 Modelo cinemático para la formación Líder-Seguidor

El sistema de formación utilizado en este trabajo esta descrito en la Figura 2: Esquema de formación Líder-Seguidor. La formación consiste en un robot móvil líder i y un robot móvil seguidor j. Las velocidades (v_i, w_i) y (v_j, w_j) expresan las velocidades lineales y angulares para cada robot, medidas en los marcos de referencia locales $\{i\}$ y $\{j\}$, respectivamente. Así mismo, se define la posición $\mathbf{r}_i = [x_i \quad y_i]^T$ y orientación θ_i del robot líder con respecto al marco de referencia global. De manera similar, definimos la posición $\mathbf{r}_j = [x_j \quad y_j]^T$ y orientación θ_j del robot seguidor.

La posición $\mathbf{r}_{ij} = [x_{ij} \quad y_{ij}]^T$ esta medida desde el marco de referencia del líder $\{i\}$ y se puede expresar como en (*2*).

$$\mathbf{r}_{ij} = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ -\sin(\theta_i) & \cos(\theta_i) \end{bmatrix} (\mathbf{r}_j - \mathbf{r}_i) \tag{2}$$

Además, la orientación θ_{ij} del robot seguidor se puede expresar desde el marco de referencia $\{i\}$ como $\theta_{ij} = \theta_j - \theta_i$.

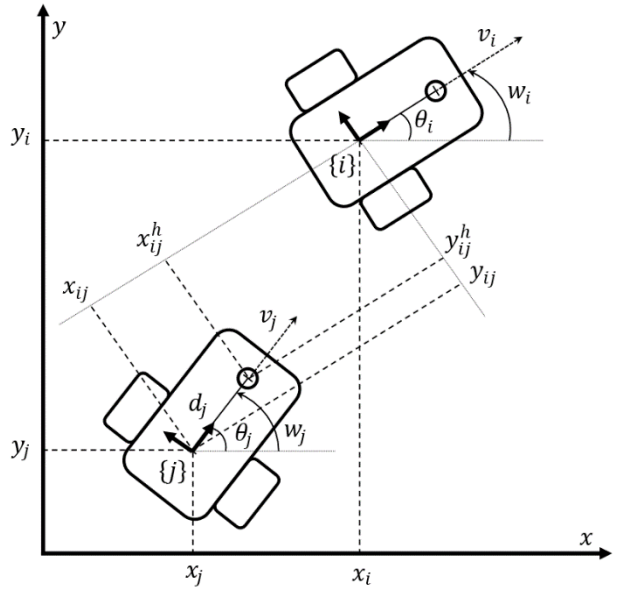

Figura 2: Esquema de formación Líder-Seguidor.

Por propósitos del control de formación, se define un punto $\mathbf{r}_{ij}^h = \begin{bmatrix} x_{ij}^h & y_{ij}^h \end{bmatrix}^T$ localizado a una distancia d_j desde el marco de referencia local $\{j\}$ del seguidor. Este punto se puede calcular como se expresa en (*3*).

$$\mathbf{r}_{ij}^h = \mathbf{r}_{ij} + d_j \begin{bmatrix} \cos(\theta_{ij}) \\ \sin(\theta_{ij}) \end{bmatrix} \tag{3}$$

Finalmente, derivando $\mathbf{r}_{ij}^h$ y la orientación θ_{ij} con respecto al tiempo se obtiene el modelo cinemático

(*4*), donde $\mathbf{G}_{ij}$ se define en (*5*) (Liang, et al., 2016) (Liang, Wang, Liu, Liu, & Chen, 2020).

$$\begin{bmatrix} \dot{x}_{ij}^h \\ \dot{y}_{ij}^h \end{bmatrix} = \mathbf{G}_{ij} \begin{bmatrix} v_j \\ w_j \end{bmatrix} + \begin{bmatrix} -v_i \\ 0 \end{bmatrix} + w_i \begin{bmatrix} y_{ij}^h \\ -x_{ij}^h \end{bmatrix}$$
$$\dot{\theta}_{ij} = w_j - w_i \tag{4}$$

$$\mathbf{G}_{ij} = \begin{bmatrix} \cos(\theta_{ij}) & -d_j \sin(\theta_{ij}) \\ \sin(\theta_{ij}) & d_j \cos(\theta_{ij}) \end{bmatrix} \tag{5}$$

Para más detalles sobre la cinemática para la formación Líder-Seguidor, se recomienda consultar (Liang, Wang, Liu, Liu, & Chen, 2020).

2.3 Control para la formación Líder-Seguidor

En (Liang, Wang, Liu, Liu, & Chen, 2020), se propone el diseño de un controlador para la formación Líder-Seguidor. En este trabajo, se hace referencia a esta estrategia como el controlador de formación [Liang 2020]. En esta estrategia, se define una posición de referencia $\mathbf{r}_{ij}^* = [x_{ij}^* \quad y_{ij}^*]^T$ relativa al marco de referencia del líder $\{i\}$. En otras palabras, el vector $\mathbf{r}_{ij}^*$ define una formación deseada. El propósito del controlador es calcular las velocidades (v_j, w_j) del robot seguidor para llevar la posición actual $\mathbf{r}_{ij}^h$ a la posición de referencia $\mathbf{r}_{ij}^*$.

La estrategia de control de formación [Liang 2020] define como en (*6*), donde $\mathbf{K}_{ij}$ es una matriz simétrica definida positiva y $\Delta\mathbf{r}_{ij}^{*} = \mathbf{r}_{ij}^{h} - \mathbf{r}_{ij}^{*}$ es el error de formación. La matriz $\mathbf{K}_{ij}$ define las ganancias del control de formación. Para poder implementar el controlador, se asume que las posiciones y orientaciones relativas de cada robot son accesibles. También se asume que las velocidades del robot líder (v_i, w_i) son medibles.

$$\begin{bmatrix} v_j \\ w_j \end{bmatrix} = \mathbf{G}_{ij}^{-1}\left\{-\mathbf{K}_{ij}\Delta\mathbf{r}_{ij}^{*} - \begin{bmatrix} -v_i \\ 0 \end{bmatrix} - w_i \begin{bmatrix} y_{ij}^{h} \\ -x_{ij}^{h} \end{bmatrix}\right\} \tag{6}$$

En el esquema de control (*6*), el cálculo de las velocidades del robot (v_j, w_j), no solo dependen de los ajustes de la matriz de ganancias $\mathbf{K}_{ij}$, también depende de las velocidades del robot líder (v_i, w_i). Es decir, valores altos para las velocidades (v_i, w_i) provocan valores altos para (v_j, w_j). En este caso, las ganancias de $\mathbf{K}_{ij}$ no deberían de ser grandes para no calcular velocidades excesivas para el robot seguidor. Por otra parte, valores bajos para (v_i, w_i) provocan valores bajos para (v_j, w_j). En este otro caso, las ganancias de $\mathbf{K}_{ij}$ deberían de ser mas grandes para llevar los errores $\Delta\mathbf{r}_{ij}^{*}$ a un vecindario cercano a 0. La desventaja de esta estrategia de control es que la ganancia $\mathbf{K}_{ij}$ es fija.

3. Descripción del Controlador Neuro Proporcional-Derivativo para la formación Líder-Seguidor

3.1 Controlador Neuro Proporcional-Derivativo

El esquema del controlador NPD se muestra en la Figura 3. El controlador NPD está compuesto por una neurona de dos entradas y una salida.

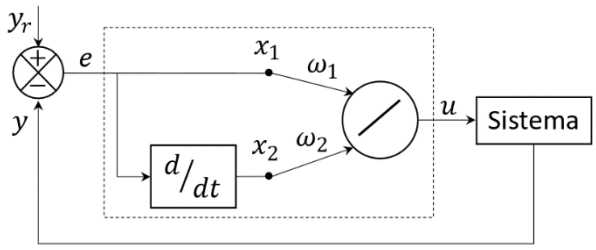

Figura 3: Controlador Neuro Proporcional-Derivativo.

El error $e(k)$ es la diferencia (*7*) entre una referencia $y_r(k)$ y la salida del sistema $y(k)$ en el instante de tiempo k. Las entradas $x_1(k)$ y $x_2(k)$ están definidas como el error (*8*) y la derivada del error (*9*), respectivamente en tiempo discreto (Hernandez-Barragan, et al., 2021). Los pesos de la neurona representan las ganancias del control, donde $\omega_1(k)$ y $\omega_2(k)$ son la ganancia proporcional y la ganancia derivativa, respectivamente. Estos pesos se ajustan en línea utilizando el algoritmo FKE. Finalmente, la salida $u(k)$ representa la acción de control (*10*).

$$e(k) = y_r(k) - y(k) \tag{7}$$

$$x_1(k) = e(k) \tag{8}$$

$$x_2(k) = e(k) - e(k-1) \tag{9}$$

$$u(k) = \omega_1(k)x_1(k) + \omega_2(k)x_2(k) \tag{10}$$

3.2 Algoritmo de entrenamiento con base en el Filtro de Kalman Extendido

El algoritmo de entrenamiento con base en el Filtro de Kalman Extendido (FKE) proporciona tasas de aprendizaje y tiempo de convergencia más rápidos que los algoritmos de entrenamiento con base en gradientes (Hernandez-Barragan, et al., 2020), (Hernandez-Barragan, et al., 2021). Además, el uso del algoritmo FKE es ideal para aplicaciones en línea y en tiempo real.

El FKE tiene como objetivo encontrar los pesos óptimos de la neurona que minimizan el error de predicción (Sanchez & Alanis, Redes neuronales: conceptos fundamentales y aplicaciones a control automático., 2006). Los pesos de la neurona se ajustan utilizando las reglas (*11*), (*12*) y (*13*) de actualización en el instante de tiempo k.

$$\mathbf{K}(k) = \mathbf{P}(k)\mathbf{H}(k)[\mathbf{R}(k) + \mathbf{H}^T(k)\mathbf{P}(k)\mathbf{H}(k)]^{-1} \tag{11}$$

$$\boldsymbol{\omega}(k+1) = \boldsymbol{\omega}(k) + \eta\mathbf{K}(k)e(k) \tag{12}$$

$$\mathbf{P}(k+1) = \mathbf{P}(k) - \mathbf{K}(k)\mathbf{H}^T(k)\mathbf{P}(k) + \mathbf{Q}(k) \tag{13}$$

donde

- $\boldsymbol{\omega}(k)$ es el estado que representa los pesos de la neurona $\boldsymbol{\omega}(k) = [\omega_1(k) \quad \omega_2(k)]^T$.
- $\mathbf{K}(k) \in \mathbb{R}^{2\times1}$ es el vector de ganancias de Kalman.
- $\mathbf{P}(k) \in \mathbb{R}^{2\times2}$ es la matriz de covarianza de error de predicción.
- $\mathbf{Q}(k) \in \mathbb{R}^{2\times2}$ es la matriz de covarianza del ruido del proceso.
- $\mathbf{R}(k) \in \mathbb{R}$ representa el error de covarianza de medición.
- $\mathbf{H}(k) \in \mathbb{R}^{2\times1}$ contiene las derivadas parciales de la salida de la red $u(k)$ con respecto a cada de uno los pesos.
- η es un factor de aprendizaje.

Los parámetros de Kalman η, $\mathbf{P}$, $\mathbf{Q}$ y $\mathbf{R}$ se ajustan de manera experimental. El factor de aprendizaje η se ajusta con la intención de minimizar el error $e(k)$. Las matrices $\mathbf{P}$ y $\mathbf{Q}$ se inicializan con valores en la diagonal principal, mientras que $\mathbf{R}$ es un escalar. La matriz $\mathbf{Q}$ se ajusta para lidiar con el ruido del proceso y $\mathbf{R}$ con el ruido de medición. Finalmente, el vector $\mathbf{H}(k)$ está definido como (*14*).

$$\mathbf{H}(k) = \begin{bmatrix} \dfrac{\partial u(k)}{\partial \omega_1(k)} \\ \dfrac{\partial u(k)}{\partial \omega_2(k)} \end{bmatrix} = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \tag{14}$$

Mas detalles sobre el entrenamiento de redes neuronales con base en el algoritmo FKE se pueden encontrar en (Sanchez, Alanís, & Loukianov, Discrete-time high order neural control, 2008).

Durante el entrenamiento de la neurona, los pesos se podrían ajustar con valores negativos. Así mismo, los pesos pueden crecer sin límite. En este trabajo, recomendamos mantener los pesos de la neurona acotados. Es importante asegurar (*15*), donde ω_1^{max} y ω_2^{max} definen los límites máximos de ajuste de la parte proporcional y derivativa, respectivamente.

$$0 \leq \omega_1 \leq \omega_1^{max}, \qquad 0 \leq \omega_2 \leq \omega_2^{max} \tag{15}$$

3.3 Controlador Neuro Proporcional-Derivativo para la formación Líder-Seguidor

En este trabajo, se propone utilizar el enfoque de control NPD para dotar al controlador (*6*) con la capacidad de adaptar sus ganancias en línea. Se pretende sustituir solo el termino $\mathbf{K}_{ij}\Delta\mathbf{r}_{ij}^{*}$ por dos controladores NPD. Las demás ecuaciones del controlador permanecen sin modificación.

Dada la formación deseada $\mathbf{r}_{ij}^{*}$ y la posición actual del robot seguidor $\mathbf{r}_{ij}^{h}$, se puede utilizar el error de formación $\Delta\mathbf{r}_{ij}^{*} = \mathbf{r}_{ij}^{h} - \mathbf{r}_{ij}^{*} = \mathbf{e}$ como un error en el instante de tiempo k definido como $\mathbf{e}(k) = [e_x(k) \quad e_y(k)]^T$. Dado que el vector de error $\mathbf{e}(k)$ está compuesto por dos errores $e_x(k)$ y $e_y(k)$, es necesario utilizar dos controladores NPD. La salida de cada controlador adaptativo define una acción de control $\mathbf{u}(k)$ tal que $\mathbf{u}(k) = [u_x(k) \quad u_y(k)]^T$.

La estrategia de control se puede reescribir como (*16*).

$$\begin{bmatrix} v_j \\ w_j \end{bmatrix} = \mathbf{G}_{ij}^{-1}\left\{-\mathbf{u}(k) - \begin{bmatrix} -v_i \\ 0 \end{bmatrix} - w_i \begin{bmatrix} y_{ij}^{h} \\ -x_{ij}^{h} \end{bmatrix}\right\} \tag{16}$$

Durante las siguientes secciones, el desempeño del controlador NPD propuesto (*16*) se comparará con el controlador [Liang 2020] (6).

4. Resultados

En esta sección se muestra el desempeño del controlador NPD propuesto para la formación Líder-Seguidor con base en simulaciones y experimentos. También consideramos incluir una comparación del controlador NPD propuesto (*16*) con el controlador [Liang 2020] (6). Los detalles para las pruebas de simulación y experimentación se muestran a continuación.

Se incluye una prueba de control en nivel simulación y otra prueba a nivel experimental. En cada prueba, el robot móvil líder i sigue una trayectoria variante en el tiempo de manera independiente al robot móvil seguidor. El propósito del control de formación es calcular la acción de control (v_j, w_j) para el robot seguidor j con la intención de que mantenga una formación deseada $\mathbf{r}_{ij}^{*}$ con respecto al robot líder i.

En cada prueba, la posición inicial del robot líder esta selecciona como $\mathbf{r}_i = [0 \quad 0]^T$ y la orientación $\theta_i = 0$. Para el robot seguidor, su posición inicial es $\mathbf{r}_j = [-0.9 \quad -0.3]^T$ con orientación $\theta_j = 0$. Además, se consideró utilizar $d_j = 0.08$ para el robot seguidor y para la formación deseada $\mathbf{r}_{ij}^{*} = [-0.6 \quad 0]^T$.

Para la matriz de ganancias $\mathbf{K}_{ij}$ del controlador clásico, se utilizó (17).

$$\mathbf{K}_{ij} = \begin{bmatrix} 0.3 & 0 \\ 0 & 0.3 \end{bmatrix} \tag{17}$$

En el caso del control NPD, el factor de aprendizaje se estableció como $\eta = 0.05$. Los pesos iniciales de la red se consideraron como $\boldsymbol{\omega} = [0.3 \quad 0]^T$. Los pesos máximos para limitar las ganancias del control NPD se seleccionaron como $\omega_1^{max} = \omega_2^{max} = 2.0$. Estos parámetros se configuraron de manera experimental. Finalmente, parámetros $\mathbf{R}$, $\mathbf{P}$ y $\mathbf{Q}$ de Kalman se establecieron como (*18*), (*19*) y (*20*), respectivamente, de acuerdo con (Hernandez-Barragan, et al., 2020), (Hernandez-Barragan, et al., 2021).

$$\mathbf{R} = 0.001 \tag{18}$$

$$\mathbf{P} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{19}$$

$$\mathbf{Q} = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix} \tag{20}$$

Para comparar el desempeño de las estrategias de control para la formación líder seguidor, se muestran los resultados de los errores de formación $\Delta\mathbf{r}_{ij}^{*} = \mathbf{r}_{ij}^{h} - \mathbf{r}_{ij}^{*} = [\Delta x_{ij} \quad \Delta y_{ij}]^T$ en gráficas.

También, se considera incluir gráficas para comparar el seguimiento de formación expresado desde el marco de referencia global. Es decir, las gráficas muestran la formación de referencia $\mathbf{r}_j^*$ y la posición actual $\mathbf{r}_j^h$. La posición $\mathbf{r}_{ij}^h$ se puede expresar desde el marco de referencia global como (*21*).

$$\mathbf{r}_j^h = \begin{bmatrix} x_j^h \\ y_j^h \end{bmatrix} = \mathbf{r}_j + d_j \begin{bmatrix} \cos(\theta_j) \\ \sin(\theta_j) \end{bmatrix} \tag{21}$$

Además, la posición deseada $\mathbf{r}_{ij}^*$ se puede calcular desde el marco de referencia global con $\mathbf{r}_j^*$ definido como (*22*).

$$\mathbf{r}_j^* = \begin{bmatrix} x_j^* \\ y_j^* \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \mathbf{r}_{ij}^* + \mathbf{r}_i \tag{22}$$

Finalmente, la acción de control (v_j, w_j) de ambas estrategias también se considera reportar con graficas.

Las especificaciones de equipo de cómputo utilizado para las pruebas es procesador Intel Core i7®-4770 (Intel Core i7, es una marca registrada de Intel Corporation, EE. UU.) con CPU a 3.4 GHz y 16 GB de RAM. El sistema operativo Windows® 10 (Windows, es una marca registrada de Microsoft Corporation, EE. UU.) y el software Matlab® R2021a (Matlab, es una marca registrada de Mathworks, EE. UU.). Los Turtlebot3® corren bajo Ubuntu® 18.04 (Ubuntu, es una marca registrada de Canonical Ltd., Reino Unido) y ROS Kinetic (ROS, es una marca registrada de Open Robotics, EE.UU.).

4.1 Resultados de simulación

Para la prueba de simulación, se cuenta con todas las mediciones necesarias, tanto posiciones y orientaciones relativas y globales, así como las velocidades del robot móvil líder (v_i, w_i). Los resultados de la prueba se muestran a continuación.

Los errores de formación $\Delta\mathbf{r}_{ij}^*$ para las estrategias de formación comparadas se muestran en la Figura 4. La Figura 4 (a) muestra que el controlador [Liang 2020] converge a un vecindario cercano a cero después de 15 segundos de simulación. También se puede observar que el error Δy_{ij} converge un poco más rápido que Δx_{ij}. Así mismo, la Figura 4 (b) muestra que la estrategia de control propuesta NPD converge antes de 5 segundos. Además, ambos errores Δx_{ij} y Δy_{ij} convergen de manera similar. Este efecto se debe notar en la grafica de seguimiento de formación. De acuerdo con estos resultados, el controlador NPD reporta un tiempo de convergencia menor comparado con el controlador [Liang 2020].

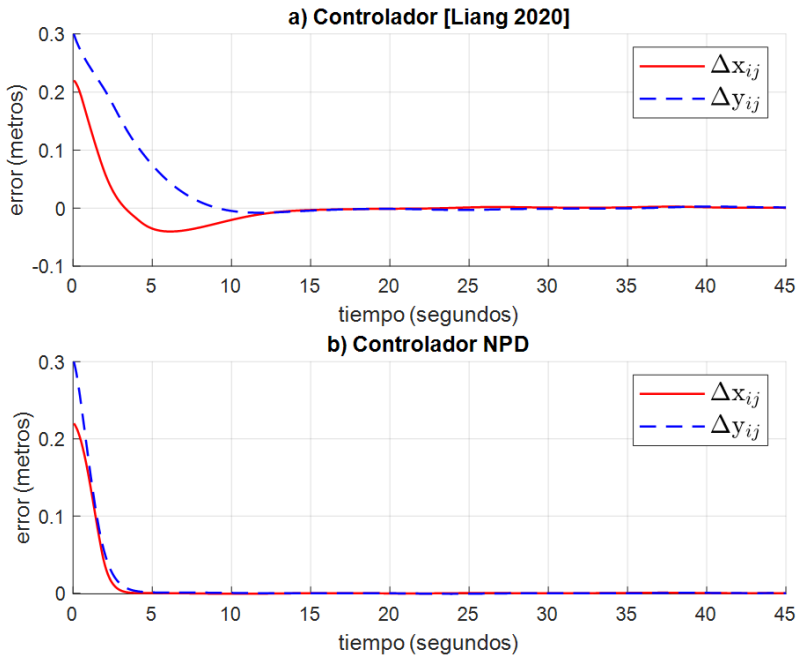

Figura 4: Errores de formación $\Delta \mathbf{r}_{ij}^{*}$. a) muestra los resultados de controlador [Liang 2020], mientras que b) muestra los resultados del control propuesto.

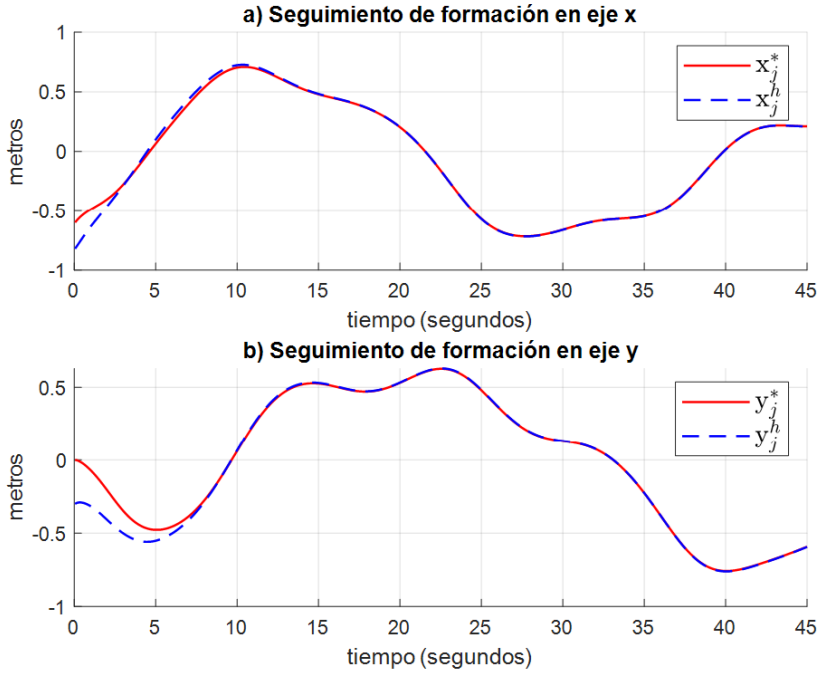

Figura 5: Seguimiento de formación del controlador [Liang 2020].

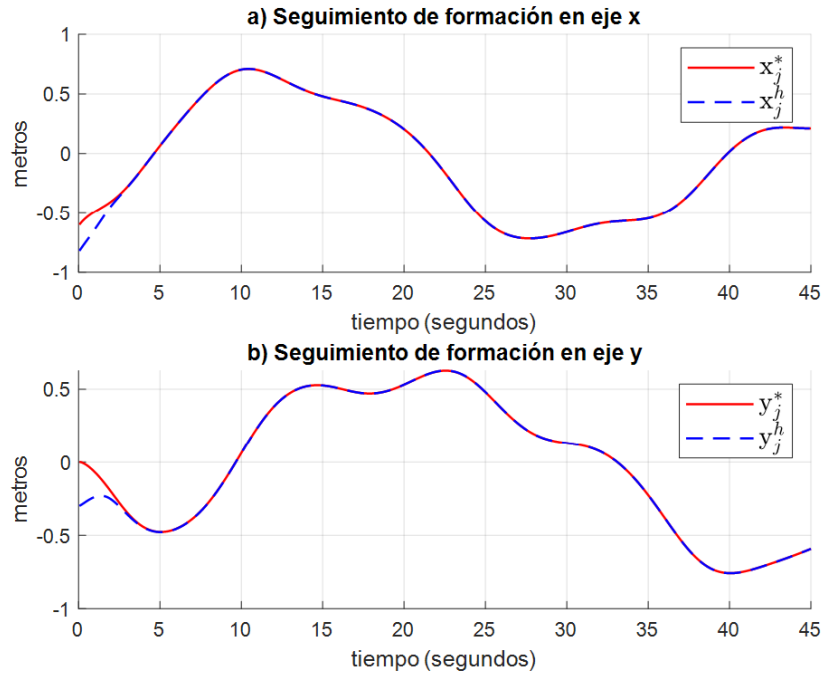

Figura 6: Seguimiento de formación del controlador propuesto NPD.

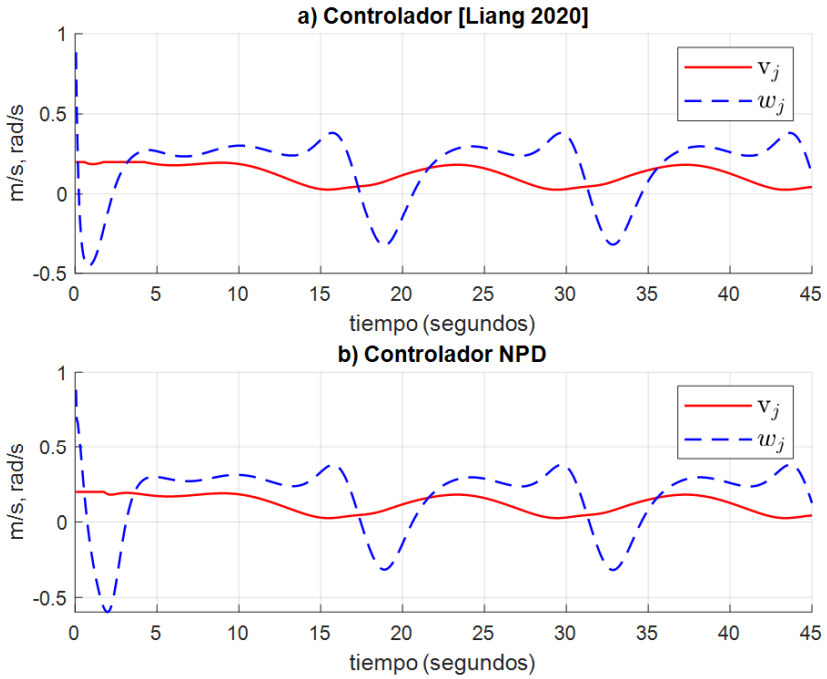

Figura 7: Acción de control para robot móvil seguidor j. a) muestra los resultados de controlador clásico, mientras que b) muestra los resultados del control propuesto. m: metros, rad: radianes, s: segundos.

La Figura 5 reporta los resultados del seguimiento de formación para el caso del controlador [Liang 2020]. Como se esperaba al analizar los errores de formación, después de los 15 segundos la referencia y la posición actual son similares. Además, se corrige más rápido el seguimiento en el eje y. Por otra parte, la Figura 6 muestra los resultados del seguimiento de formación para el caso del controlador NPD. Debido a la rápida convergencia del error, la posición actual alcanza la referencia antes de 5 segundos. Ambas estrategias de control demuestran un seguimiento de formación adecuado, sin embargo, el controlador propuesto NPD resuelve el seguimiento en menor tiempo.

Para terminar con los resultados de simulación, la acción de control para el robot móvil seguidor se reporta en la Figura 7. Las velocidades que se aplican al robot seguidor son similares para ambas estrategias de control. En el caso del controlador propuesto NPD de la Figura 7 (b), la velocidad angular w_j es menor a -0.5 rad/s. Al contrario, la velocidad angular w_j es mayor a -0.5 rad/s para el caso del controlador clásico de la Figura 7 (a). Esto indica que el controlador NPD calculó una acción de control mayor para corregir el seguimiento en menor tiempo. Después de los 10 segundos no se nota una diferencia significativa entre las velocidades de ambos controladores. Finalmente, las velocidades lineales y angulares reportadas son adecuadas para su aplicación en el Turtlebot3®.

4.2 Resultados de experimentación

Para la prueba de experimentación, se utilizaron dos robots móviles Turtlebot3®. La plataforma móvil Turtlebot3 es un robot diferencial para educación e investigación de bajo costo (ROBOTIS, 2022). Además, tiene muchas herramientas de software y hardware libre (ROBOTIS, 2022). Uno de los modelos de Turtlebot3® es el Waffle Pi. Entre sus componentes principales se encuentran: una tarjeta Raspberry Pi 3, dos actuadores XL430-W20, un módulo IMU con giroscopio y acelerómetro de 3 ejes, un LiDAR LDS-01, una tarjeta openCR (32-bit ARM Cortex M7) para el control de los actuadores, una cámara RGB modelo PI, entre otros componentes. Los robots Turltebot3 utilizados se muestran en la Figura 8.

Figura 8: Plataforma móvil Turtlebot3® Waffle Pi.

Para implementar los esquemas de control, se utilizó la API de la plataforma ROS. ROS ofrece diversas paqueterías para acceder a los Turtlebot3®. Por medio de estos paquetes podemos

enviar las velocidades lineales v y angulares w a cada robot. Además, es posible acceder a la odometría de cada robot para medir la posición y orientación desde un marco de referencia global. También se cuenta con paquetes para medir las velocidades lineales y angulares actuales. Finalmente, los TurtleBot3® son controlados desde una PC remota por medio de ROS.

Es importante mencionar que las velocidades v y w se acotan antes de aplicarlas al TurtleBot3® para evitar dañar o desgastar los actuadores. Las acotaciones de la velocidad lineal son $-0.2 \leq v \leq 0.2$ y para la velocidad angular $-0.9 \leq w \leq 0.9$. Los resultados de la prueba se muestran a continuación.

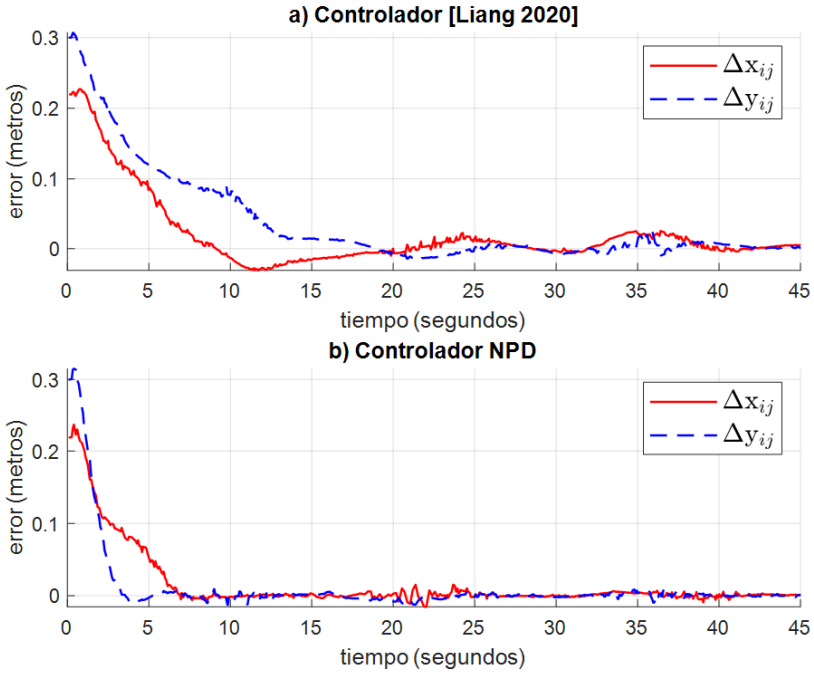

Figura 9: Errores de formación $\Delta \boldsymbol{r}_{ij}^{*}$. a) muestra los resultados de controlador clásico, mientras que b) muestra los resultados del control propuesto.

La Figura 9 muestra los errores de formación $\Delta \mathbf{r}_{ij}^{*}$ para las estrategias de formación comparadas. La Figura 9 (a) reporta el caso del controlador [Liang 2020], donde el error de formación converge alrededor de 20 segundos y se mantiene acotado. Pero, el error de formación es mucho mayor al error de formación de las simulaciones. Lo que sugiere que es necesario ajustar las ganancias del controlador para reducir los errores de formación. Así mismo, la Figura 9 (b) reporta el caso del controlador propuesto NPD. Como se esperaba por los resultados de las simulaciones, el error de convergencia es más rápido que el controlador clásico. En este caso, el error de formación converge a un vecindario cercano a 0 poco después de 5 segundos. Los errores de formación de simulación y experimentación son parecidos. Esto indica que la neurona adapta los pesos para compensar cualquier perturbación, ruido o dinámica no modelada, tratando de tener un desempeño similar al de simulación. Esto sugiere que no es necesario ajustar ningún parámetro a menos que se desee mejorar el desempeño del controlador NPD.

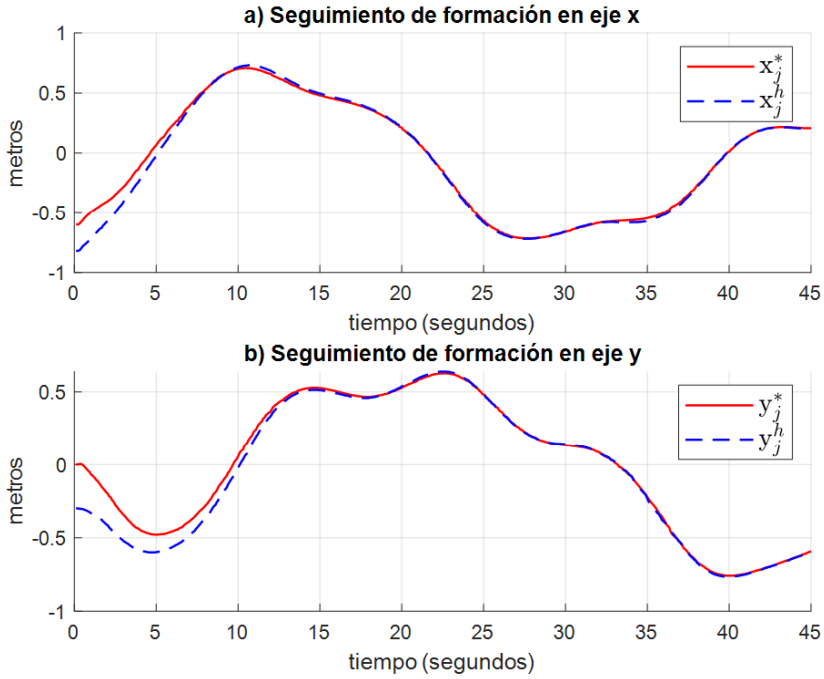

Figura 10: Seguimiento de formación del controlador clásico.

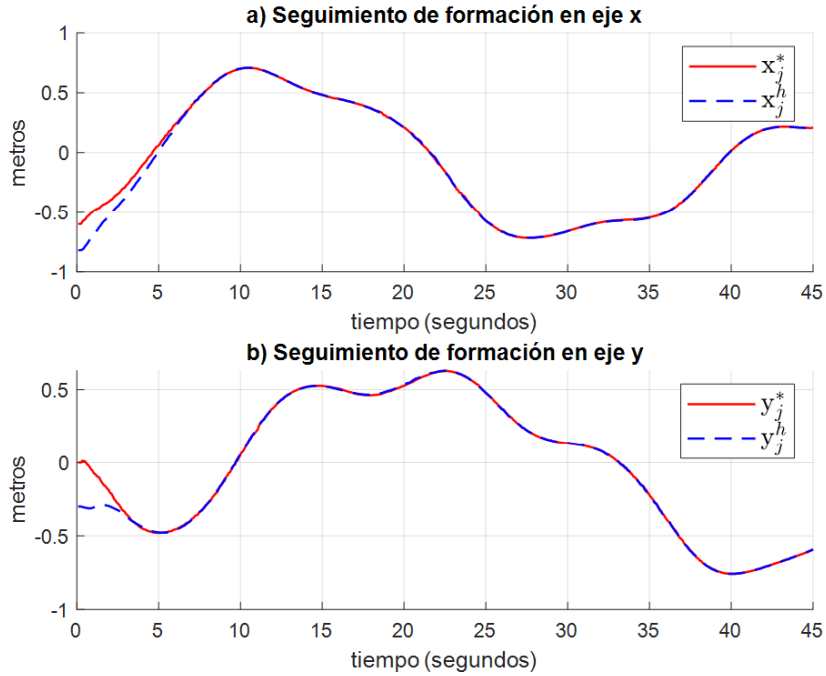

Figura 11: Seguimiento de formación del controlador propuesto NPD.

La Figura 10 muestra los resultados de seguimiento de formación del controlador [Liang 2020]. Aunque el error de formación converge después de 20 segundos, es posible notar el error de seguimiento, sobre todo en el eje x. Mientras tanto, la Figura 11 reporta los resultados de seguimiento de formación del controlador NPD. Como se esperaba, el seguimiento de formación es

muy parecido al seguimiento en la simulación. Poco después de 5 segundos, la posición actual alcanza la referencia. De acuerdo con estos resultados, el controlador propuesto NPD no solo resuelve el seguimiento en menor tiempo, sino que también demuestra un desempeño similar al de simulación, sin necesidad de un ajuste.

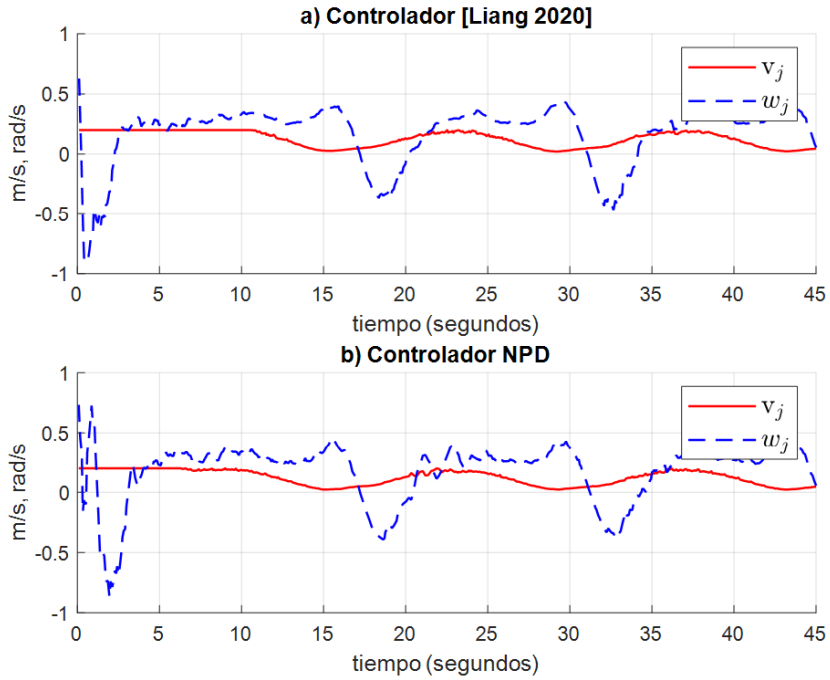

Figura 12: Acción de control para robot móvil seguidor j. a) muestra los resultados de controlador clásico, mientras que b) muestra los resultados del control propuesto. m: metros, rad: radianes, s: segundos.

Para terminar con los resultados de experimentación, la acción de control para el robot móvil seguidor se reporta en la Figura 12. Aunque la acción de control en ambos controladores tiene un patrón similar al de las simulaciones, se nota la presencia de ruido en las mediciones y perturbaciones. De acuerdo con los resultados del controlador [Liang 2020] en la Figura 12 (a), se puede observar que la acción de control v_j se mantuvo acotada hasta los primeros 10 segundos. Por otra parte, los resultados del controlador NPD en la Figura 12 (b), la acción de control v_j se mantuvo acotada solo en los primeros 6 segundos. Esto demuestra que el robot seguidor alcanzo más rápido la referencia con el controlador NPD. Los resultados del controlador NPD también muestran el efector de adaptación en la velocidad angular w_j ante la presencia de perturbaciones.

5. Discusión

De acuerdo con los resultados de la prueba de simulación, tanto el controlador [Liang 2020] como el controlador propuesto NPD reportan un seguimiento de formación adecuado. Es decir, ambos controladores llevar la posición del seguidor actual $\mathbf{r}_{ij}^{h}$ a la posición de referencia $\mathbf{r}_{ij}^{*}$ con un error de formación $\Delta\mathbf{r}_{ij}^{*}$ que tiende a 0. Sin embargo, el controlador NPD reporta un tiempo de convergencia menor comparado con el controlador [Liang 2020].

Por otra parte, de acuerdo con los resultados de la prueba de experimentación, el desempeño del controlador [Liang 2020] empeoró debido a que reportó un error de formación $\Delta\mathbf{r}_{ij}^{*}$ mayor al obtenido en la simulación. Esto indica que el controlador [Liang 2020] requiere del ajuste de ganancias para mejor el seguimiento de formación. En contraste, el controlador propuesto NPD presentó errores de formación $\Delta\mathbf{r}_{ij}^{*}$ muy parecidos a los errores de formación en simulación. Esto significa que no se requiere ningún ajuste para el controlador NPD. Además, la acción de control de ambos esquemas de control muestra la presencia de ruido de medición y perturbaciones externas. Esto demuestra que el entrenamiento de la neurona ajusta sus pesos para compensar cualquier perturbación, ruido o dinámica no modelada, tratando de tener un desempeño similar al de simulación.

Las estrategias de control presentadas en este trabajo asumen que se tendrán disponibles la pose (posición y orientación) del robot móvil seguidor y el robot líder. Además, se asume que las velocidades del líder son medibles. En la práctica, no siempre es posible acceder a todas las mediciones. Por otra parte, la odometría mediante el uso de sensores abordo como codificadores ópticos o módulos IMU suelen carecer de precisión. Para mejorar la estimación de la pose de los robots móviles, es conveniente trabajar con otros esquemas como auto localización mediante el uso de cámaras y marcadores de entorno (Yu, et al., 2019). Otra estrategia útil para la estimación de la pose podría ser la técnica SLAM (Kolhatkar & Wagle, 2021). Se deja como trabajo futuro incluir algunas de estas estrategias para el cálculo de la pose y velocidades de cada robot, para el propósito de mejorar la precisión de la formación. Además de que es posible plantear una estrategia de control que no requiere del uso de las velocidades del robot líder.

6. Conclusiones

Este trabajo presentó un controlador para la formación Líder-Seguidor con base en un controlador Neuro Proporcional-Derivativo (NPD). El controlador NPD hace uso del Filtro de Kalman Extendido (FKE) para ajustar sus pesos en línea. El desempeño del controlador propuesto se validó mediante simulaciones y experimentos. Además, se incluyó una comparación con un controlador para la formación Líder-Seguidor del estado del arte. Finalmente, dos robots móviles modelo Turtlebot3® Waffle Pi se utilizaron en la prueba de experimentación mediante la plataforma ROS.

De acuerdo con los resultados reportados, el controlador propuesto NPD presentó errores de formación similares, tanto en la prueba de experimentación y la prueba de simulación. Incluso cuando la acción de control de ambos esquemas de control muestra la presencia de ruido de medición y perturbaciones externas. Esto demuestra que el entrenamiento con base en el FKE compensa las perturbaciones, ruido o dinámica no modelada. En contraste, el error de formación del controlador del estado del arte empeoró en la prueba experimental, lo que indica que este controlador requiere del ajuste de ganancias para mejor el seguimiento de formación. Además, el controlador NPD reporta un tiempo de convergencia menor comparado con el controlador del estado del arte.

Podemos concluir que el esquema de control adaptativo NPD mejora el desempeño del controlador del estado del arte, con respecto a menor tiempo de convergencia del error de formación y capacidad de ajuste de ganancias en línea. Esto tiene la ventaja de evitar ajuste de ganancias, ante la presencia de ruido o perturbaciones externas.

Aunque este trabajo resuelve la formación Líder-Seguidor utilizando robots móviles diferenciales, como trabajo futuro resulta atractivo llevar este enfoque de formación, al uso de robots móviles omnidireccionales. Además, se pretender extender el uso del controlador NPD para resolver problemas de formación con base en consenso, utilizando cuatro o más robots Turtlebot3® Waffle Pi.

Referencias

Alfaro, A., & Morán, A. (2020). Leader-Follower Formation Control of Nonholonomic Mobile Robots. En 2020 IEEE ANDESCON, 1-6.

Ben-Ari, M., & Mondada, F. (2017). Elements of Robotics. Springer International Publishing.

Bryson, J. J. (2019). The Past Decade and Future of AI's Impact on Society. BBVA. Obtenido de Open Mind BBVA.

Dai, S.-L., He, S., Chen, X., & Jin, X. (2020). Adaptive Leader–Follower Formation Control of Nonholonomic Mobile Robots With Prescribed Transient and Steady-State Performance. IEEE Transactions on Industrial Informatics, 3662-3671.

Hassan, M. F., & Hammuda, M. (2020). Leader-follower formation control of mobile nonholonomic robots via a new observer-based controller. International Journal of Systems Science, 1243-1265.

Hernandez-Alvarado, R., Garcia-Valdovinos, L. G., Salgado-Jimenez, T., Gómez-Espinosa, A., & Fonseca-Navarro, F. (2016). Neural Network-Based Self-Tuning PID Control for Underwater Vehicles. Sensors.

Hernandez-Barragan, J., Rios, J. D., Alanis, A. Y., Lopez-Franco, C., Gomez-Avila, J., & Arana-Daniel, N. (2020). Adaptive single neuron anti-windup PID controller based on the extended Kalman filter algorithm. MDPI Electronics, 636.

Hernandez-Barragan, J., Rios, J. D., Gomez-Avila, J., Arana-Daniel, N., Lopez-Franco, C., & Alanis, A. Y. (2021). Adaptive neural PD controllers for mobile manipulator trajectory tracking. PeerJ Computer Science, 7, e393. doi:10.7717/peerj-cs.393

Kamel, M. A., Yu, X., & Zhang, Y. (2020). Formation control and coordination of multiple unmanned ground vehicles in normal and faulty situations: A review. Annual Reviews in Control, 49, 128-144. doi:https://doi.org/10.1016/j.arcontrol.2020.02.001

Klancar, G., Zdesar, A., Blazic, S., & Skrjanc, I. (2017). Wheeled mobile robotics: from fundamentals towards autonomous systems. Butterworth-Heinemann.

Kolhatkar, C., & Wagle, K. (2021). Review of SLAM algorithms for indoor mobile robot with LIDAR and RGB-D camera technology. Innovations in electrical and electronic engineering, 397-409.

Liang, X., Liu, Y.-H., Wang, H., Chen, W., Xing, K., & Liu, T. (2016). Leader-Following Formation Tracking Control of Mobile Robots Without Direct Position Measurements. IEEE Transactions on Automatic Control, 61(12), 4131-4137. doi:10.1109/TAC.2016.2547872

Liang, X., Wang, H., Liu, Y.-H., Liu, Z., & Chen, W. (2020). Leader-Following Formation Control of Nonholonomic Mobile Robots With Velocity Observers. IEEE/ASME Transactions on Mechatronics, 25(4), 1747-1755. doi:10.1109/TMECH.2020.2990991

Lu, Q., Miao, Z., Zhang, D., Ye, L. Y., Yang, S. X., & Su, C.-Y. (2019). Distributed Leader-follower Formation Control of Nonholonomic Mobile Robots. IFAC-PapersOnLine, 67-72.

Oh, K.-K., Park, M.-C., & Ahn, H.-S. (2015). A survey of multi-agent formation control. Automatica, 53, 424-440. doi:https://doi.org/10.1016/j.automatica.2014.10.022

Open Robotics. (2021). Why ROS? Obtenido de ROS: https://www.ros.org

ROBOTIS. (2022). Robotis Turtlebot3 Overview. Obtenido de Robotis Turtlebot3: https://emanual.robotis.com/docs/en/platform/turtlebot3/overview/

Sanchez, E. N., & Alanis, A. Y. (2006). Redes neuronales: conceptos fundamentales y aplicaciones a control automático. Cinvestav Unidad Guadalajara: Editorial Prentice Hall.

Sanchez, E. N., Alanís, A. Y., & Loukianov, A. G. (2008). Discrete-time high order neural control. Warsaw: Springer.

Sanfeliu, A., Hagita , N., & Saffiotti, A. (2008). Robotics and Autonomous SystemsRobotics and Autonomous Systems 56 (2008) 793–797. Network robot systems, 56(10), 793–797. doi:https://doi.org/10.1016/j.robot.2008.06.007

Siegwart, R., Nourbakhsh, I. R., & Scaramuzza, D. (2011). Introduction to Autonomous Mobile Robots. MIT Press.

Tang, W., Wang, L., Gu, J., & Gu, Y. (2020). Single Neural Adaptive PID Control for Small UAV Micro-Turbojet Engine. Sensors.

Wang, H., Guo, D., Liang, X., Chen, W., Hu, G., & Leang, K. K. (2016). Adaptive Vision-Based Leader-Follower Formation Control of Mobile Robots. IEEE Transactions on Industrial Electronics, 64(4). doi:10.1109/TIE.2016.2631514

Xuan-Mung, N., & Hong, S. K. (2019). Robust adaptive formation control of quadcopters based on a leader–follower approach. International Journal of Advanced Robotic Systems, 1-11.

Yu, H., Fu, Q., Yang, Z., Tan, L., Sun, W., & Sun, M. (2019). Robust Robot Pose Estimation for Challenging Scenes With an RGB-D Camera. IEEE Sensors Journal, 2217-2229.

Zhong, J., Zhu, Y., Zhao, C., Han, Z., & Zhang, X. (2020). Position tracking of a pneumatic-muscle-driven rehabilitation robot by a single neuron tuned PID controller. Complexity.