

Jornadas de Automática

Diseño de controladores H-infinito para el seguimiento de trayectorias con robots móviles con ruedas

Nuñez, Ricardo^{a,*}, Nicoletta, Armando^b, Armesto, Leopoldo^c, González, Antonio^d, Sala, Antonio^d

^a Escuela Téc. Superior de Ingeniería Aeroespacial y Diseño Industrial, Universitat Politècnica de València, C/Camino de Vera s/n, 46022 Valencia, España.

^b Department of Industrial Engineering, University of Naples Federico II, C. so Umberto I 40, 80138, Naples, Italia.

^c Instituto de Diseño y Fabricación, Universitat Politècnica de València, C/Camino de Vera s/n, 46022 Valencia, España.

^d Instituto de Automática e Informática Industrial, Universitat Politècnica de València, C/Camino de Vera s/n, 46022 Valencia, España.

To cite this article: Nuñez, et. al. 2024. Design of H-infinity controllers for trajectory tracking for wheeled mobile non-holonomic robots. *Jornadas de Automática*, 45. <https://doi.org/10.17979/ja-cea.2024.45.10838>

Resumen

En este trabajo se presenta el desarrollo de una estructura de control de dos grados de libertad (feedforward+feedback) para seguimiento de trayectorias en una plataforma robótica TurtleBot3 Waffle Pi en la que se combina linealización por realimentación, control por punto descentrado y control H-infinito (en varias configuraciones). Los resultados demuestran la viabilidad de la propuesta de cara a futuros desarrollos en entornos más complejos.

Palabras clave: Control óptimo, Rechazo de perturbaciones, Navegación y control de guiado, Seguimiento.

Design of H-infinity controllers for trajectory tracking with wheeled mobile robots

Abstract

This work presents the development of a two-degree-of-freedom (feedforward+feedback) control structure for trajectory tracking on a TurtleBot robotic platform that combines feedback linearization, offset point control, and H-infinity control (in various configurations). The results demonstrate the viability of the proposal for future developments in more complex environments.

Keywords: Optimal control, Disturbance rejection, Guidance navigation and control, Tracking.

1. Introducción

En el control cinemático de robots, muchas veces se plantean estrategias sencillas como el control proporcional y se llevan con éxito a la práctica. No obstante, en entornos más complejos existen perturbaciones, ruido de medida, dinámica no modelada, errores paramétricos, etc (Goswami, 2018), (Díaz, 2023), además de la posible interacción entre varios robots en sistemas multi-agente. Las estrategias “sencillas” en estos contextos podrían no ser suficientes y, por tanto, merece la pena presentar un primer enfoque basado en H-infinito (Flamm, 1990) en la aplicación de control cinemático de robots.

El control \mathcal{H}_∞ es un enfoque de control óptimo que basa su metodología en la modulación de la respuesta en frecuencia de bucle cerrado de una planta. Por su relación con el teorema de pequeña ganancia, es la base del control robusto lineal y las metodologías con incertidumbre estructurada como μ -síntesis que derivan de él.

Este trabajo presenta un enfoque preliminar con intenciones fundamentalmente tutoriales, intentando hacer comprender la estructura global del problema y las distintas consideraciones a llevar a cabo para hacer funcionar ese tipo de reguladores. No obstante, el objetivo a medio plazo es aplicar las ideas a entornos más complicados, como es el caso de sistemas robóticos multi-agentes (González, 2018).

Este trabajo discute una estructura de 2 grados de libertad (feedforward+feedback) en el que la parte feedforward es calculada a partir de la geometría de la trayectoria, y la parte feedback se implementa mediante un control \mathcal{H}_∞ sobre un integrador (en cada coordenada de los ejes X, Y de movimiento); el estado de dicho integrador se considera la posición de un punto descentrado y se hace la adecuada linealización por realimentación para calcular velocidades de rueda en un robot tipo TurtleBot3 Waffle Pi.

La estructura del trabajo es como sigue: la sección 2 discute el problema a resolver y la metodología básica de diseño (diagrama de bloques, control por punto descentrado); la sección 3 presenta la planta generalizada y los pesos necesarios para diseñar los controladores \mathcal{H}_∞ objeto de experimentación; la sección 4 presenta los detalles del prototipo TurtleBot3 utilizado en los experimentos cuyos resultados se discuten en la sección 5; una sección de conclusiones cierra este trabajo.

2. Metodología de Diseño

El objetivo de esta sección es detallar la metodología de diseño propuesta para el control de robots no-holonómicos móviles con ruedas para el seguimiento de trayectorias.

El problema se descompone, por un lado, en la generación de una señal de control compuesta por dos componentes, la señal de prealimentación (o *feedforward* por su terminología en inglés) y la señal de retroalimentación (o *feedback* por su terminología en inglés). La primera de las señales generará las acciones de control que deberían aplicarse ante la ausencia de perturbaciones y ruido de medida para seguir una trayectoria determinada, mientras que la segunda de las señales se encargará de corregir las desviaciones con respecto a la trayectoria a seguir como consecuencia de las perturbaciones y ruido de medida.

En la Figura 1 se observa la estructura de prealimentación y realimentación propuesta y a continuación se detallarán cada uno de los elementos de dicha figura.

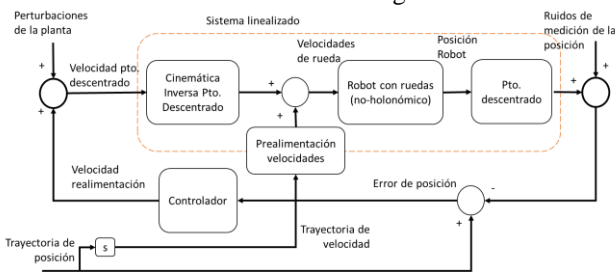


Figura 1: Estructura de prealimentación de control de punto descentrado con linealización de la realimentación de un robot móvil con ruedas (no-holonómico).

2.1. Control por punto descentrado

Dado un robot no-holonómico móvil con ruedas sus ecuaciones cinemáticas de movimiento (sin deslizamiento de sus ruedas) vienen definidas por:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \mathbf{J}_r(\theta) \cdot \mathbf{v}$$

siendo, x , y y θ las posiciones de las coordenadas X e Y y la orientación del robot, respectivamente; y v y ω son las velocidades lineales y angulares de la base. El objetivo del control, en el caso del seguimiento de trayectorias, es controlar la posición x e y del robot, mientras que la orientación vendrá, indirectamente, especificada por la propia trayectoria.

Dada una referencia de posición en las coordenadas X e Y definidas por expresiones que dependen del tiempo $x^*(t)$ y $y^*(t)$ y sus correspondientes derivadas $\dot{x}^*(t)$, $\dot{y}^*(t)$, $\ddot{x}^*(t)$ e $\ddot{y}^*(t)$, las velocidades de prealimentación lineal y angular de la base de un robot no-holonómico pueden obtenerse, respectivamente, a partir de las siguientes expresiones:

$$v^{ff}(t) = \sqrt{\dot{x}^{*2}(t) + \dot{y}^{*2}(t)}$$

$$\omega^{ff}(t) = \frac{\dot{y}^*(t)\dot{x}^*(t) - \ddot{x}^*(t)y^*(t)}{(v^{ff}(t))^2}$$

En consecuencia, las velocidades de la rueda izquierda y derecha para un robot diferencial se calculan a partir de:

$$\omega_L^{ff}(t) = v^{ff}(t) - b\omega^{ff}(t)$$

$$\omega_R^{ff}(t) = v^{ff}(t) + b\omega^{ff}(t)$$

siendo b la mitad de la separación entre las ruedas.

Para obtener las señales de realimentación, se linealizará primero el sistema para realizar un control por punto descentrado, que nos permitirán controlar, en este caso, un robot no-holonómico móvil con ruedas, arbitrariamente en cualquier dirección X-Y a partir de comandos de velocidad de rueda. Por tanto, desde un punto de vista del diseño el objetivo será controlar la posición del punto descentrado, en cuyo caso se tratará como dos integradores desacoplados, uno para cada una de las direcciones a controlar del punto descentrado, de manera que el controlador será el encargado de compensar las perturbaciones del sistema y ruidos de medición.

En este caso, lo más común es ubicar dicho punto a una distancia e en la dirección de movimiento de avance. Por tanto, su posición es:

$$\mathbf{x}_e = \begin{bmatrix} x_e \\ y_e \end{bmatrix} = \begin{bmatrix} x + e \cdot \cos \theta \\ y + e \cdot \sin \theta \end{bmatrix}$$

El parámetro e tiene influencia en el control de las desviaciones perpendiculares a la trayectoria. En este trabajo este parámetro se ha fijado a la distancia entre el escáner láser del robot y el eje de las ruedas, siendo esta la variable controlada en nuestro diseño posterior. La velocidad de dicho punto es:

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \end{bmatrix} = \begin{bmatrix} 1 & 0 & -e \cdot \sin \theta \\ 0 & 1 & e \cdot \cos \theta \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \mathbf{J}_e(\theta) \cdot \dot{\mathbf{x}}$$

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \end{bmatrix} = \begin{bmatrix} \cos \theta & -e \cdot \sin \theta \\ \sin \theta & e \cdot \cos \theta \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \mathbf{J}_e(\theta) \cdot \mathbf{J}_r(\theta) \cdot \mathbf{v}$$

Se puede por tanto comprobar que para $e \neq 0$ la jacobiana $(\mathbf{J}_e(\theta) \cdot \mathbf{J}_r(\theta))^{-1}$ es invertible, y por tanto, el sistema linealizado resaltado en la Figura 1 se comporta como:

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \end{bmatrix} = \begin{bmatrix} 1/s & 0 \\ 0 & 1/s \end{bmatrix} \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \end{bmatrix}$$

3. Diseño de controladores \mathcal{H}_∞

El sistema a controlar puede por tanto modelarse como un integrador, por tratarse de un controlador cinemático que debe eliminar las perturbaciones y filtrar los ruidos de medida en base al siguiente esquema de planta generalizada:

$$\begin{bmatrix} e \\ u \\ \tilde{e} \end{bmatrix} = \begin{bmatrix} \frac{1}{s} & 0 & \frac{1}{s} \\ 0 & 0 & 1 \\ -\frac{1}{s} & 1 & -\frac{1}{s} \end{bmatrix} \cdot \begin{bmatrix} d \\ n \\ u \end{bmatrix}$$

siendo u la acción de control correspondiente a la velocidad deseada del punto descentrado, d la perturbación de entrada al proceso (en forma de velocidad del punto descentrado diferente a la teóricamente comandada), n el ruido de medida, y tanto e como \tilde{e} el error de posición y el error de posición medido (cambiado de signo para que el control salga con realimentación negativa, como suele ser habitual) respectivamente. Dado que el control en las coordenadas X e Y está desacoplado, diseñaremos un controlador para cada una de las coordenadas.

Junto con la planta generalizada, diseñaremos una serie de filtros (pesos en frecuencia) de la perturbación, ruido de medida, error y acción de control, que nos permitirán obtener la planta generalizada ponderada, y que definirán la respuesta en frecuencia de las diferentes señales a considerar.

Los filtros para las entradas generalizadas son:

$$G_d(s) = \frac{0.033s + 0.033}{10s + 1}$$

$$G_n(s) = 0.0333$$

porque se ha considerado una amplitud máxima de perturbaciones a baja frecuencia de 3.3cm/s de la diferencia entre la velocidad comandada y la realmente adquirida por el robot, y diez veces menos a altas frecuencias. Véase el diagrama de Bode de este filtro en la Figura 2 (izquierda). El ruido máximo esperado en la medida de posición es de 3.3cm.

Los filtros para las salidas generalizadas son la inversa de estas “plantillas” que reflejarán el tamaño máximo de error y variable controlada deseados cuando las perturbaciones tengan un tamaño peor-caso dados por $G_d(s)$ y $G_n(s)$:

$$G_e(s) = 0.05$$

$$G_u(s) = \frac{0.1 \cdot \left(\frac{\pi}{T_s} s + 1 \right)}{\frac{\pi}{T_s} s + 1}$$

siendo $T_s = 0.125s$ el periodo de muestreo utilizado para la discretización del filtro en su implementación por computador. El polo del filtro de la entrada se ha escogido de esta manera para que disminuya su actividad a partir de la frecuencia de un 50% de la frecuencia de Nyquist. De esta forma, la implementación discreta de este diseño no se verá afectada por efectos de alta frecuencia o por aliasing debidos

al muestreo. El diagrama de Bode del filtro de la entrada se muestra en la Figura 2 (derecha). El error máximo admisible será de 5cm en el peor de los casos.

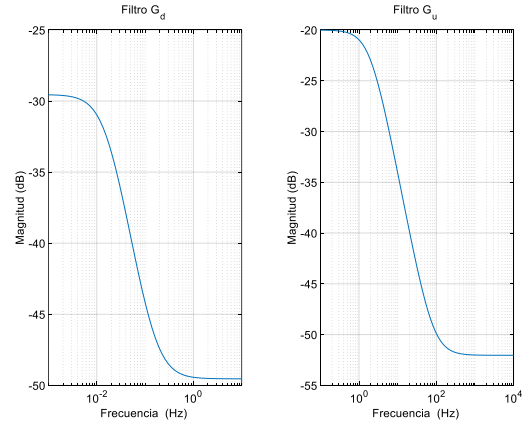


Figura 2: Respuesta en frecuencia de los filtros $G_d(s)$ y $G_u(s)$.

Para el diseño de estos filtros se han considerado varios aspectos:

- Todos los filtros deben cumplir con la condición $\|G(\omega)\|_\infty \neq 0$, para asegurarnos que el controlador/observador diseñado no tenga ganancias infinitas a alguna de sus frecuencias como consecuencia una señal de ganancia nula, lo cual no es deseable en la práctica.
- Los filtros usados para la entrada de la planta generalizada (perturbación y ruido de medida) se han escogido en base al contenido frecuencial esperado en la práctica experimental, y que deben ser modificados si se aprecia que los controladores excitan señales no deseadas a determinadas frecuencia.
- Se han buscado filtros sencillos para no subir el orden de la planta generalizada ponderada en exceso. Por ejemplo el filtro del ruido de medida se ha hecho constante porque al limitar la alta frecuencia del actuador se consigue plantear adecuadamente el problema para el caso SISO.

En base a estas especificaciones, hemos diseñado tres controladores \mathcal{H}_∞ para comparar sus prestaciones:

- Controlador proporcional con filtro paso-bajo: $G_c(s) = \frac{10K_p}{s+10}$, habiéndose escogido la frecuencia de corte de $10 \text{ rad} \cdot s^{-1}$ por considerarse la frecuencia a partir de la cual se desea filtrar el ruido de medida.
- Controlador PID: $G_c(s) = K_p + \frac{K_i}{s} + \frac{K_d s}{T_f s + 1}$, siendo en este caso el optimizador el que debe escoger las tres constantes del controlador, así como la frecuencia del filtro paso-bajo.
- Controlador de orden completo.

Obviamente, la metodología \mathcal{H}_∞ no es la única opción para diseño de PID (las dos primeras opciones), pero dado que se van a comparar con esta métrica, se diseñarán con la metodología \mathcal{H}_∞ , mediante el comando de Matlab `hinfstruct` que permite escoger la estructura del controlador,

mientras que el controlador de orden completo se ha diseñado con el comando `hinfscyn`, que es el único que garantiza que la solución sea la óptima.

La siguiente tabla muestra la norma infinito de los diferentes controladores diseñados. Todos ellos por debajo de 1, lo que garantiza que, frente a perturbaciones y ruidos unitarios, la señal del error y la acción de control tienen un valor por debajo de 1, en magnitudes “escaladas” y frente a perturbaciones de tamaño máximo $G_d(s)$ y $G_n(s)$ un error y acción de control menor de $G_e(s)$ y $G_u(s)$ en magnitudes físicas originales.

Tabla 1: Norma infinito

Controlador	γ
P (filtro)	0.7649
PID	0.7535
Orden completo	0.7038

Tabla 2: Controladores diseñados

Controlador	
P (filtro)	$G_c(s) = \frac{19.8}{s + 10}$
PID	$G_c(s) = 1.42 + \frac{0.0992}{s} - \frac{0.442s}{0.325s + 1}$
Orden completo	$A = \begin{bmatrix} -14.38 & -18.16 & -0.6821 \\ 1.349 & -46.13 & 0.06692 \\ 0.009605 & -61.44 & -0.08782 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ -65.18 \\ -87.02 \end{bmatrix}$, $C = \begin{bmatrix} 0.9537 \\ -0.03546 \\ -0.001332 \end{bmatrix}^T$, $D = 0$

La Figura 3, muestra el diagrama de Bode en bucle cerrado de los controladores. Observamos que el controlador PID es el único que añade un integrador, pero sin embargo su ganancia entre $0.02 \text{ rad} \cdot \text{s}^{-1}$ y $0.3 \text{ rad} \cdot \text{s}^{-1}$ es de 2.1, mientras que el resto de los controladores son capaces de ofrecer una ganancia de 4 hasta $1 \text{ rad} \cdot \text{s}^{-1}$. A partir de esa frecuencia, los controladores atenúan la señal de control para no excitar señales de alta frecuencia. El controlador de orden completo no tiene un integrador, pero dado que su ganancia es elevada a bajas frecuencias, en la práctica generará un error de posición más pequeño que el controlador proporcional. A alta frecuencia todos los controladores filtran las señales, incluso un controlador PID con una respuesta en frecuencia lo suficientemente pequeña para atenuar este tipo de señales.

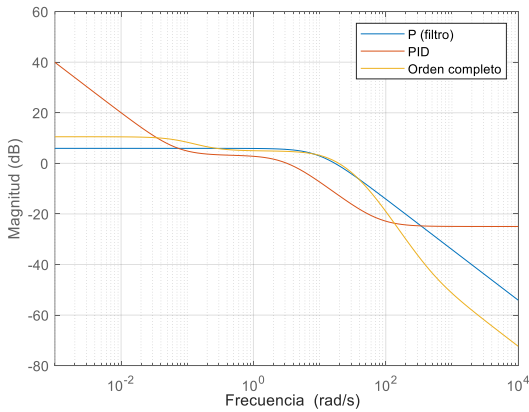


Figura 3: Respuesta en frecuencia de los diferentes controladores diseñados.

En la Figura 4 se observa el comportamiento dinámico frente a escalón de perturbación y ruido en la señal del error y en la acción de control.

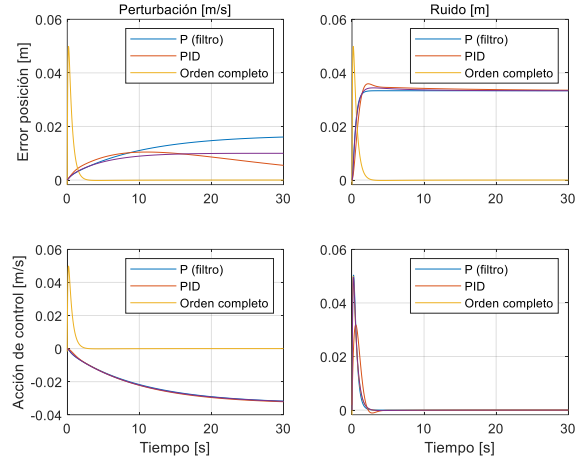


Figura 4: Respuesta frente a escalón de los diferentes controladores frente a perturbación del sistema y ruido de medida.

Con objeto de comprobar que el diseño en bucle cerrado cumple con las especificaciones exigidas en los filtros, se muestra en la Figura 5 el diagrama de Bode en bucle cerrado de los diferentes controladores:

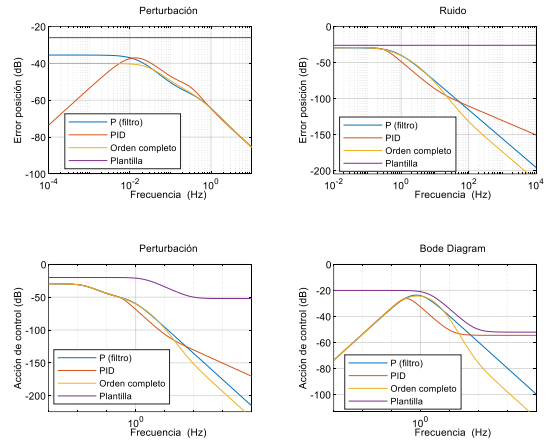


Figura 5: Respuesta en frecuencia del sistema lineal en bucle cerrado.

4. Plataforma robótica TurtleBot3

En este trabajo, se explica cómo controlar el robot TurtleBot3 Waffle Pi (Amsters, 2020) para implementar un control cinemático para el seguimiento de trayectorias, tarea esencial en una gran variedad de aplicaciones de robótica móvil, desde la exploración autónoma a la construcción de mapas en entornos de interior.

El robot dispone de un láser de telemetría (LIDAR) LDS-02 2D y de motores inteligentes XL430-W210 que permiten el control de bajo nivel de velocidad de giro, proporcionando, mediante codificadores ópticos, el giro realizado por las ruedas como sistema de odometría. Además, el robot incorpora una IMU de 6-ejes que mide las aceleraciones y velocidades angulares.

El robot dispone de una Raspberry Pi 4 que permite que el robot pueda ser controlado con ROS2 desde un computador abordo Ubuntu. El uso del estándar DDS (Data Distribution Services) ofrece un mecanismo de transferencia de la información más fiable y que escala mejor a sistemas multi-agentes, en el caso de querer controlar varios robots.

A continuación, se describe la arquitectura de control de los nodos de ROS que se han usado para el control cinemático del robot TurtleBot3 y cuya relación se muestra en la Figura 6. De forma simplificada, sólo se muestran los nodos principales que intervienen en este proceso de localización y control.

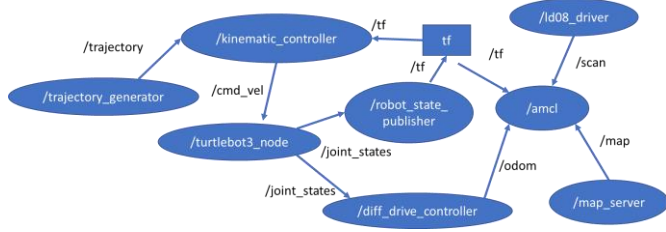


Figura 6: Arquitectura de control de nodos para el control cinemático.

El nodo de ROS2 “/turtlebot3_node” se suscribe al tópico “/cmd_vel” para recibir comandos de movimiento con mensajes del tipo Twist, que a su vez envía al nodo “/diff_drive_controller” las velocidades de las ruedas. Dicho nodo es el que gestiona la comunicación con la placa de control de bajo nivel OpenCR1.0 de bajo nivel que es la encargada de enviar las señales de control para los motores y leer la posición de las ruedas y cuya posición es publicada a través del tópico “/odom”. Por otro lado, el nodo “/ld08_driver” utiliza el controlador del LIDAR para proporcionar los datos de telemetría a través del tópico “/scan” (del tipo LaserScan). Para estimar la posición del robot se usa el algoritmo de localización del robots AMCL (Adaptive Monte-Carlo Localization) (Zhang, 2012), al cual se le debe de proporcionar un mapa a través del nodo “/map_server”. Dicho mapa ha sido construido previamente con un algoritmo de SLAM (Simultaneous Localization and Mapping) (Durrant, 2006) basado en el paquete “turtlebot3_cartographer” (Darby, 2024).

El nodo “/kinematic_controller” recibe la posición del robot a través de los diferentes sistemas de transformación que se publican en el tópico “/tf”, y que relacionan la posición del robot con respecto al sistema de odometría y del sistema de odometría con respecto el mapa y recibe la trayectoria a seguir a través de un mensaje específico con la información necesaria de la posición y velocidad de la trayectoria a seguir.

5. Validación de los controladores

Con el fin de validar los controladores se ha implementado el control cinemático para que siga una trayectoria periódica con la siguiente expresión matemática:

$$\begin{bmatrix} x^*(t) \\ y^*(t) \end{bmatrix} = \begin{bmatrix} \cos \phi_0 & -\sin \phi_0 \\ \sin \phi_0 & \cos \phi_0 \end{bmatrix} \cdot \begin{bmatrix} A_x \sin \frac{2\pi t}{T} \\ 4\pi t \\ A_y \sin \frac{4\pi t}{T} \end{bmatrix}$$

siendo $T = 120s$ el periodo de la trayectoria, $A_x = 1.5$ y $A_y = 0.6$ son las longitudes de los ejes principales de la

trayectoria, estando girada, por conveniencia, $\phi_0 = -\text{atan}\left(\frac{2A_y}{A_x}\right)$.

Como detalle de implementación, mencionar que los controladores de la Tabla 2 se han diseñado en realidad en discreto, al periodo de muestreo $T_s = 0.125s$, habiéndose discretizado la planta con el método de discretización ZOH y los filtros han sido discretizados con Tustin.

En la Figura 7 se puede apreciar que todos los controladores diseñados tienen un buen comportamiento frente al seguimiento de la trayectoria.

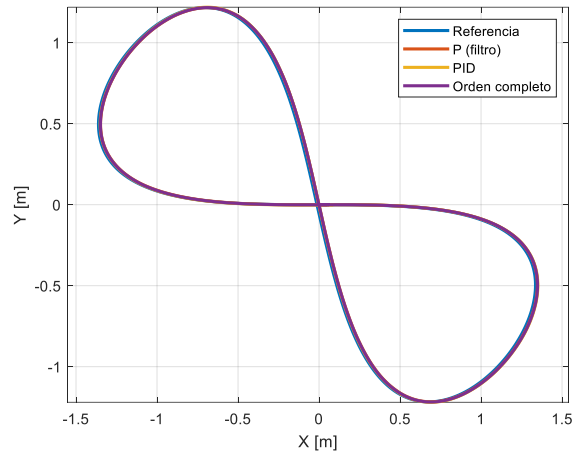


Figura 7: Trayectoria X-Y de la referencia y posición del robot (para los controladores \mathcal{H}_∞ con estructura P con filtro, PID y SS (de orden 4) y controlador \mathcal{H}_∞ de orden completo).

Las velocidades de rueda aplicadas, para el caso del controlador de orden completo se muestran en la Figura 8, donde se ha representado, por un lado, el componente de prealimentación y por otro lado la velocidad aplicada (suma del componente de prealimentación más el componente de realimentación). Se puede apreciar en la Figura que la velocidad comandada está por encima de la velocidad de prealimentación porque el control de velocidad que tienen los motores del robot tiene un error de posición debido a la fricción, ya que en sus valores por defecto tiene implementado un control proporcional para el control de velocidad de los motores.

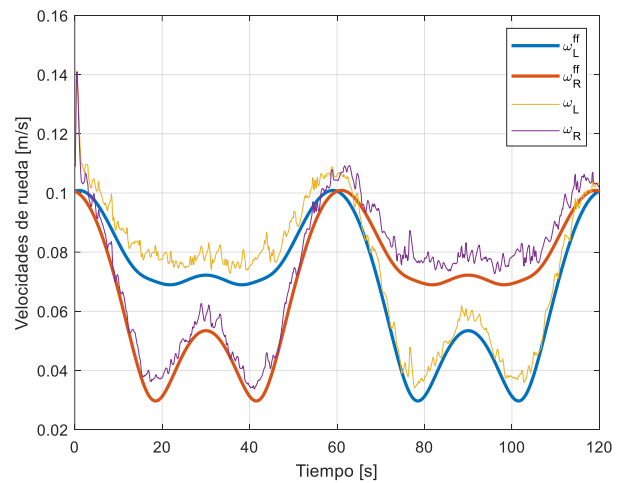


Figura 8: Velocidades de rueda aplicadas al robot con el controlador de orden completo. En trazado grueso se representan las velocidades prealimentadas (feed-forward) y en trazado fino se representan las velocidades comandadas.

En la Figura 9 se muestran los errores en las coordenadas X e Y de seguimiento de la trayectoria y los componentes de realimentación obtenidos por el controlador de orden completo.

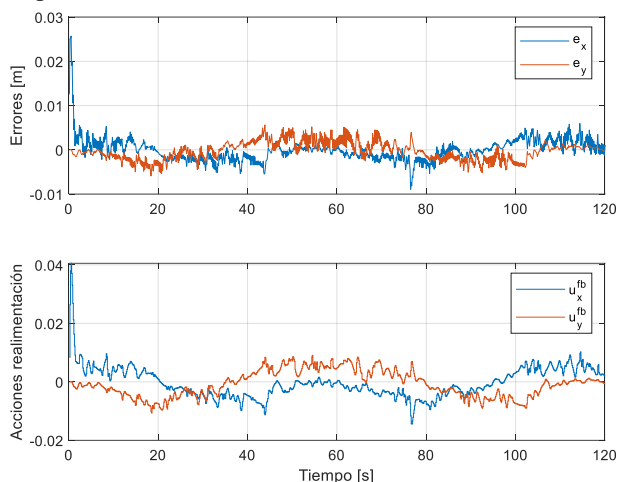


Figura 9: Errores de posición del controlador de orden completo y acciones de control (de punto descentrado) generadas por el controlador en la realimentación.

El espectro frecuencial de las señales de error en X de los diferentes controladores se muestra en la Figura 10, donde se aprecia que los controladores PID y de orden completo tienen un menor componente de error medio (error a frecuencia cero), pero en general tienen un comportamiento frecuencial similar.

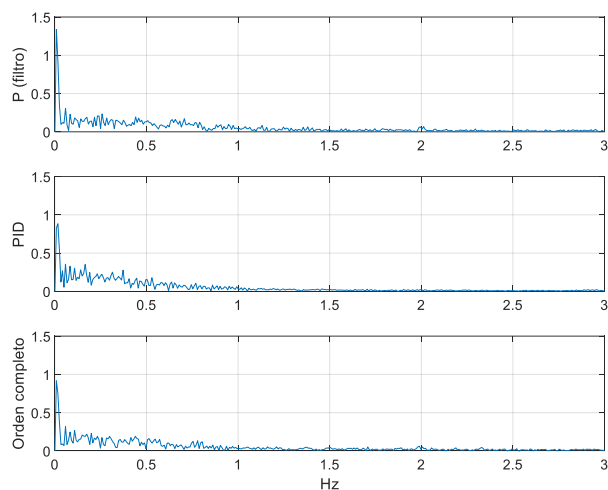


Figura 10: Transformada de Fourier de la señal del error de posición en X de los diferentes controladores.

La Tabla 3 muestra el error medio cuadrático de los diferentes controladores, donde se aprecia que el controlador de orden completo es el que obtiene un ligero menor error de los tres, pero todos en el mismo orden de magnitud.

Tabla 3: Valor de la media cuadrática del error de posición

Controlador	RMS(e_x)	RMS(e_y)
-------------	--------------	--------------

P (filtro)	0.0032	0.0027
PID	0.0031	0.0024
Orden completo	0.0028	0.0021

Se observa que el controlador de orden completo es el que menor valor RMS tiene del error, a cambio de tener una actividad de alta frecuencia mayor que otras opciones (una especie de efecto “cama de agua” que suele ser usual en control: para disminuir error en baja frecuencia hay que actuar “rápido” a bandas de frecuencia más alta.

3. Conclusiones

En este trabajo se han comparado distintas estrategias de diseño de controladores basándose en técnicas \mathcal{H}_∞ , desde el controlador \mathcal{H}_∞ óptimo de orden completo hasta otros más sencillos (P, PID) mediante \mathcal{H}_∞ estructurado. Se trata de un primer enfoque hacia estas metodologías para sentar las bases en trabajos futuros con incertidumbre de modelado, múltiples agentes, retardos, etc.

Agradecimientos

Este trabajo ha sido realizado gracias al apoyo del proyecto LOCPU PID2020-116585GB-I00 financiado/a por MCIN/AEI/10.13039/501100011033/, REACT-EU, Action IV.5, PhD programmes on green topics y Department of Industrial Engineering, University of Naples, Federico II.

Referencias

Amsters, R., Slaets, P., 2020. Turtlebot 3 as a robotics education platform. In Robotics in Education: Current Research and Innovations 10 (pp. 170-181). Springer International Publishing. https://link.springer.com/chapter/10.1007/978-3-030-26945-6_16

Darby Lim, Pyo, “turtlebot3_cartographer” (fecha de acceso 31 mayo 2024). url: https://index.ros.org/p/turtlebot3_cartographer/

Díaz, J. A., Estrada-Manzo, V., & Bernal, M., 2023. Diseño de Par Calculado Robusto No Lineal basado en Observación: una Solución por Medio de Desigualdades Matriciales Lineales. Revista Iberoamericana de Automática e Informática industrial (In Press). DOI: 10.4995/riai.2024.20765

Durrant-Whyte, H., Bailey, T., 2006. Simultaneous localization and mapping: part I. IEEE robotics & automation magazine, 13(2), 99-110. 10.1109/MRA.2006.1638022

González, A., Aragüés, R., López-Nicolás, G., Sagüés, C., 2018. Stability analysis of nonholonomic multiagent coordinate-free formation control subject to communication delays. International Journal of Robust and Nonlinear Control, 28 (14), 4121-4138. DOI: 10.1002/rnc.4225

Goswami, N. K., & Padhy, P. K., 2018. Sliding mode controller design for trajectory tracking of a non-holonomic mobile robot with disturbance. Computers & Electrical Engineering, 72, 307-323. DOI: 10.1016/j.compeleceng.2018.09.021

Flamm, D. S., Yang, H., 1990. H/sup infinity/-optimal mixed sensitivity for general distributed plants. In 29th IEEE Conference on Decision and Control (pp. 134-139). IEEE. DOI: 10.1109/CDC.1990.203561

Zhang, L., Zapata, R., Lepinay, P., 2012. Self-adaptive Monte Carlo localization for mobile robots using range finders. Robotica, 30(2), 229-244. DOI: DOI: 10.1017/S0263574711000567