

Conducción segura de una plataforma móvil en configuración diferencial, basado en técnica de control predictivo

Safe driving of a mobile platform in a differential configuration, based on predictive control technique

Danilo Hernando Moreno González¹
Leonardo Enrique Solaque Guzmán²
Adriana Riveros Guevara³

¹Estudiante Ing. en Mecatrónica, Universidad Militar Nueva Granada, Bogotá, Colombia. Email: u1802929@unimilitar.edu.co

²Docente Programa Ing. en Mecatrónica, Universidad Militar Nueva Granada, Bogotá, Colombia. Email: leonardo.guzman@unimilitar.edu.co

³Docente Programa Ing. en Mecatrónica, Universidad Militar Nueva Granada, Bogotá, Colombia. Email: tmp.adriana.riveros@unimilitar.edu.co

 OPEN ACCESS



Copyright:

©2021. La revista *Ingenierías USBmed* proporciona acceso abierto a todos sus contenidos bajo los términos de la licencia creative commons Atribución no comercial SinDerivar 4.0 Internacional (CC BY-NC-ND 4.0)

Tipo de artículo: Investigación.

Recibido: 21-10-2019.

Revisado: 06-04-2020.

Aprobado: 19-06-2020.

Doi: 10.21500/20275846.4366

Referenciar así:

D. H. Moreno González, L. E. Solaque Guzmán and A. Riveros Guevara, "Conducción segura de una plataforma móvil en configuración diferencial, basado en técnica de control predictivo," *Ingenierías USBMed*, vol. 12, n.º 2, pp. 33-40, 2021.

Disponibilidad de datos:

todos los datos relevantes están dentro del artículo, así como los archivos de soporte de información.

Conflicto de intereses:

los autores han declarado que no hay conflicto de intereses.

Editor: Andrés Felipe Hernández.
Universidad de San Buenaventura, Medellín, Colombia.

Resumen. Dentro del proceso de conducción segura, particularmente en el campo del agro donde interviene una plataforma robótica, se hace necesario salvaguardar tanto la estructura como su entorno. En la literatura se encuentran aplicaciones orientadas a "self-driving", donde el objetivo es conseguir guiar y controlar sistemas como automóviles o robots inmersos en ambientes dinámicos. El actual documento presenta una opción para que un robot en configuración diferencial se desplace de manera autónoma y segura dentro de un cultivo, previniendo colisiones o daños a la estructura o su entorno. Haciendo uso de la teoría de control predictivo se realiza el diseño del control y su implementación en la plataforma. La ley de control es calculada mediante el planteamiento de una función objetivo que asocia errores de seguimiento, restricciones dinámicas, y las saturaciones de las señales de control (velocidades máxima y mínima logradas por el robot). El error cuadrático medio y su varianza cuando el robot sigue una trayectoria de referencia fue el criterio usado para verificar el desempeño del control sobre el sistema.

Palabras Clave. MPC, control, Robot, ROS.

Abstract. Within the safe driving process, particularly in the field of agriculture where a robotic platform intervenes, it is necessary to safeguard both the structure and its environment. In the literature, there are applications oriented to "self-driving", where the objective is to be able to guide and control systems such as cars or robots immersed in dynamic environments. The current document presents an option for a robot in the differential configuration to move autonomously and safely within a crop, preventing collisions, damage to the structure, or its environment. Using predictive control theory, the control design and its implementation in the platform are carried out. The control law is calculated by proposing an objective function that associates tracking errors, dynamic restrictions, and the saturations of the control signals (the maximum and minimum speeds achieved by the robot). The mean square error and its variance when the robot follows a reference trajectory was the criterion used to verify the performance of the control over the system.

Keywords. Predictive control, Robots, MPC, ROS.

I. Introducción

En los últimos años, los robots han traído grandes beneficios, permitiendo mejorar la calidad de las tareas cotidianas; quizá el reto más grande es garantizar la seguridad de sus desplazamientos en los entornos para los cuales fueron diseñados. Gracias al desarrollo de nuevos algoritmos que garantizan en gran parte un movimiento seguro, se han realizado mejoras en los vehículos autónomos, conocidos en la literatura como de conducción segura (en inglés “self-driving”), que están dotados de elementos que les permiten distinguir y navegar el ambiente que los rodea. Este tipo de robots está tomando valor en áreas donde no era usual su uso, como en el campo de la agricultura, en Han, Yang, Kim y Seo [1], se presenta un robot modular diseñado para labores de agricultura en ambientes dinámicos externos, donde se propone todo el modelamiento dinámico y un algoritmo de conducción autónoma del mismo, puesto a prueba mediante simulaciones en Matlab-simulink. En Barbosa et al. [2], se propone un robot autónomo que mediante múltiples cámaras realiza la navegación, el monitoreo y análisis de insectos, enfermedades y maleza en cultivos de plantas de soya y algodón.

Uno de los problemas que conlleva trabajar en un entorno externo, es estimar la posición actual del vehículo, como es estudiado en Wang y Lambert [3]. Así mismo, para lograr una percepción más completa del entorno, se suelen utilizar múltiples sensores, cámaras y/o unidades inerciales que llevan al robot a poseer un alto grado de procesamiento para la toma de decisiones durante la ruta, que va desde un adecuado seguimiento a una trayectoria hasta poseer alta capacidad de maniobrabilidad en su entorno, tal como se puede evidenciar en Reke et al. [4], permitiendo el desarrollo de nuevos robots, con gran capacidad de autonomía, como es el resultado de Min et al. [5], donde hacen uso de un auto eléctrico comercial añadiendo cámaras, láser 3D, sistemas de localización y toda una arquitectura de software que le permite reconocer personas y señales de tránsito logrando una navegación segura en un entorno virtual.

La Universidad Militar Nueva Granada, desde el 2016, ha adelantado la construcción e instrumentación de una plataforma robótica dedicada al agro, llamada CERES, nombre inspirado en la mitología romana (Figura 1). Esta plataforma es propulsada por motores eléctricos, aloja tolvas y tanques para nutrientes, agua y plaguicida; instrumentos como IMU, GPS, cámara multiespectral, térmica y estéreo y sensores láser 3D y 2D que permiten su navegación y detallar el estado del cultivo. Tiene capacidad de carga útil de 100 kg de abono sólido y 20 litros de plaguicida; las labores de fumigación, fertilización y deshierbado pueden realizarse gracias a actuadores lineales (en la parte inferior central) en los 3 ejes que permiten la realización de tareas

puntuales en un volumen de 130 cm de frente x 90 cm de fondo y 50 cm de alto con prestaciones para remover maleza, aplicar nutrientes sólidos y plaguicidas, e incluso hidratar la planta de ser necesario. En el marco de ese desarrollo, se han planteado contribuciones desde el procesamiento de imágenes para remoción de maleza presentados en Pulido, Solaque y Velasco [6], identificación de estrés hídrico, planificación del movimiento de la herramienta efectora y el seguimiento de trayectorias por la plataforma completa desarrollados en Solaque, Sánchez y Riveros [7], implicando estas labores técnicas de control, programación algorítmica y visión de máquina.



Figura 1. Robot CERES Fuente: Fotografía propia

En la actualidad, la programación de muchos desarrollos de sistemas robóticos se viene haciendo sobre el Sistema Operativo Robot – **ROS (Robot Operating System)**. Este programa provee la funcionalidad de un sistema operativo que engloba paquetes heterogéneos [8]. El robot CERES fue dimensionado para operar con **ROS**, dada su arquitectura, modularidad, variedad de drivers para diferentes sensores y actuadores, soporte continuo por la comunidad y ser de fuente abierta. *The Autoware Foundation* [9], desarrolla un proyecto cuya primera fase desarrolló un robot autónomo, basado en **ROS**, el cual cuenta con algoritmos de SLAM, para su localización y navegación en entornos conocidos o desconocidos, planeación y seguimiento de trayectorias mediante técnicas de inteligencia artificial. A diferencia de este proyecto y como primer acercamiento a un vehículo “self-driving”, en este artículo se detalla el desarrollo de un seguimiento seguro a una trayectoria basado en una técnica de control predictivo.

El presente documento está organizado así: La sección 1 presenta la descripción de la plataforma y su modelo en configuración diferencial; La sección 2 presenta la teoría de modelo de control predictivo **MPC (Model Predictive Control)** para el aseguramiento de conducción segura, pruebas en nivel simulación, y la proyección para realizar la integración a los sistemas de procesamiento embarcado; Por último, se presentan las conclusiones.

II. Desarrollo del artículo

A. Modelo matemático del robot CERES

Para la descripción del movimiento del sistema se tienen diferentes marcos de referencia. Con relación al marco de referencia global R_o se da la posición-orientación del robot. El marco de referencia cuerpo R_b es el asociado al centro de gravedad del robot y sobre el que se plantean todas las ecuaciones dinámicas del robot.

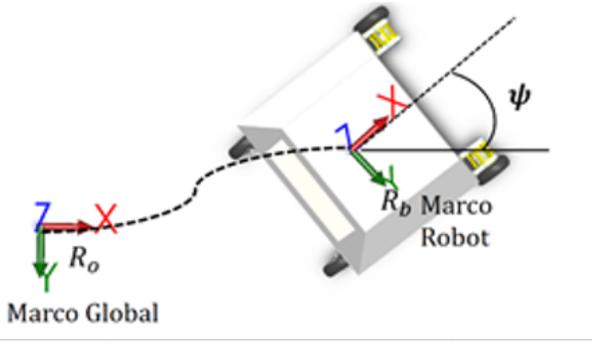


Figura 2. Marco de referencia global

Si se considera que el robot se desplaza a una velocidad traslacional u , con una velocidad rotacional ω medidas en el marco local R_b , estas pueden ser llevadas al sistema coordinado global R_o mediante una matriz de transformación. Denotando P como la posición (x, y) , y la orientación ψ del robot en el marco referencial R_o , la relación que vincula las velocidades descritas en $R_b(u, \omega)$, es presentada por la Ecuación (1).

$$\dot{p} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} c\psi & 0 \\ s\psi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} = R_0^b \begin{bmatrix} u \\ \omega \end{bmatrix}, \quad (1)$$

siendo $s\psi = \sin \psi$ y $c\psi = \cos \psi$.

Al calcular la inversa de esta matriz, se puede ir del marco de referencia global al sistema coordinado local. Matemáticamente $\begin{bmatrix} u \\ \omega \end{bmatrix}^T = (R_0^b)^{-1} \dot{p}$. Es de anotar que el sistema es no holonómico, lo que implica que se cumple $\dot{x}s\psi = \dot{y}c\psi$, lo que refleja restricciones dinámicas.

Desde la teoría de modelado de sistemas dinámicos por Newton-Euler, se puede plantear el modelado dinámico de la plataforma.

$$M\ddot{x} = \sum F \quad (2)$$

$$I\ddot{\psi} = \sum \tau \quad (3)$$

donde M es la masa del sistema, I la inercia del robot, F la fuerza y τ el par (la fuerza y el par son generados por el contacto entre las llantas y el suelo).

La Figura 3 ilustra la convención asumida para la plataforma, donde se presentan: Las fuerzas generadas

por las llantas (proporcionales al par creado por cada motor $F_{r,l} = \tau/r$, donde r es el radio de la llanta y τ el par generado por el motor), las fuerzas, el desplazamiento traslacional (x), la posición angular (ψ), el centro de gravedad del robot denotado como CG.

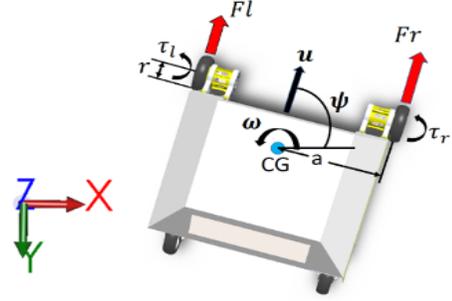


Figura 3. Marco de referencia local

Tomando el CG como punto de referencia para realizar la sumatoria de torques, se puede mostrar que el modelo que representa la dinámica de la plataforma está dado por las ecuaciones (4) y (5).

$$\beta_{11}\ddot{\theta}_r + \beta_{12}\ddot{\theta}_l + \beta\dot{\theta}_r = \tau_r \quad (4)$$

$$\beta_{21}\ddot{\theta}_r + \beta_{22}\ddot{\theta}_l + \beta\dot{\theta}_l = \tau_l \quad (5)$$

Donde

$$\beta_{11} = \beta_{22} = \left[\frac{Mr^2}{4} + \frac{I_{CG}r^2}{4a^2} + I_{llanta} \right]$$

$$\beta_{12} = \beta_{21} = \left[\frac{Mr^2}{4} - \frac{I_{CG}r^2}{4a^2} \right]$$

Solaque G, G. Sánchez y A. Riveros [10], presentan en mayor detalle la elaboración del modelo y la construcción de los controles de bajo nivel.

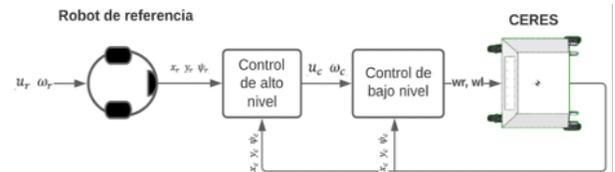


Figura 4. Jerarquización controladores de CERES: el robot de referencia es usado para generar la trayectoria de referencia en un espacio 2D

B. Controles de bajo y alto nivel

El control del desplazamiento del robot en el espacio 2D se plantea sobre una estructura jerárquica, donde los controles que aseguran la velocidad traslacional (u) y la velocidad rotacional (ω) son denominados controles de bajo nivel. Para hacer seguimiento a una trayectoria se diseñan controles de alto nivel y es precisamente aquí donde entra el control predictivo MPC,

quien asegura que la estructura cumpla la misión. De esta manera, la salida del control de alto nivel, son las referencias del control de bajo nivel, tal como lo ilustra la Figura 4.

1) Controles de bajo nivel

Para controlar la plataforma y llevarlo a un sistema donde solo se recibe una referencia de velocidad traslacional (u) y una referencia de velocidad rotacional ($\dot{\psi}$), se plantean las leyes de control de bajo nivel siguientes:

$$F_u(t) = k1_r X_u(t) + k2_r \frac{dx_u(t)}{dt} + k_{ir} \int e_u(t) dt \quad (6)$$

$$F_u(t) = k1_r X_\psi(t) + k2_l \frac{dx_\psi(t)}{dt} + k_{il} \int e_\psi(t) dt \quad (7)$$

De esta manera el sistema se puede ver como un modelo cinemático simple, similar al planteado en (1). Cabe resaltar que este modelo debe ser respetado en sus dinámicas, es decir que el sistema tendrá unos tiempos de reacción mínimos para responder y mantener la validez del modelo cinemático, el cual es usado por controles de alto nivel.

2) Control de alto nivel

La idea del control de alto nivel es usar la técnica de control predictivo MPC tal que asegure el seguimiento a la trayectoria y al mismo tiempo evaluar las restricciones dinámicas de la plataforma y las saturaciones de señales de control para los motores. Así, la conducción segura, parte del principio de tener sensores capaces de identificar la proximidad de elementos y hacer que lo esquivé mientras cumple la misión.

La técnica MPC se basa entonces, en el planteamiento de una función objetivo que integra restricciones dinámicas, saturaciones en las señales de control a los motores, errores de seguimiento a la trayectoria referencia y las distancias a los objetos encontrados durante la ejecución de la misión. Así, para encontrar solución se acude a la teoría de optimización [11]. Para el caso del robot CERES, se plantea el modelo dinámico del robot en tiempo discreto, quien será el modelo de predicción y permite estimar la salida en cada instante t , logrando llevarlo a un horizonte finito denotado como N_p ($y(t+k|t)$ desde $k=0, \dots, N_p$). Al hacer evolucionar el modelo, se pueden ver los cambios en el modelo asociados a la señal de control en el futuro Δu , que también se ve con un horizonte finito denotado como N_c ($\Delta u(t+k)$ desde $k=0, \dots, N_c-1$).

De otra parte, el optimizador busca que, mediante una función de costo sujeta a una serie de restricciones, genere incrementos que permitan que la variable controlada llegue a ser lo más cercana posible al modelo de predicción de referencia con el menor costo de energía, lo que se plantea como un problema de programación

cuadrática (QP) cuya forma matricial no cuadrática general para MPC es planteado en la ecuación (8):

$$J = (Y - W)^T R (Y - W) + U^T Q U \quad (8)$$

donde W corresponde a un vector columna de las referencias deseadas, Y es el vector de las salidas estimadas por el modelo de predicción y las matrices R y Q son matrices diagonales correspondientes a los pesos de ponderación de los estados y los incrementos de control, respectivamente. Para llevar la forma representada en (8) a una forma cuadrática hay que reemplazar el modelo de estimación de la forma $y = F\Delta u + f$, donde la primera parte corresponde a la respuesta forzada y la segunda a la respuesta libre. Quedando la ecuación (8) como una función de costo en forma cuadrática (9).

$$J = \frac{1}{2} u^T H u + F_0 u \quad (9)$$

donde H corresponde a la matriz Hessiana representada por $H = 2\phi^T Q \phi + R$, la cual describe la parte cuadrática de la función de costo; F_0 describe la parte lineal y está descrita como $F_0 = (F x_c)^T Q \phi$, [12]. El cálculo de las matrices ϕ y F puede profundizarse en [13].

Al describir el modelo de CERES como un robot de cuerpo rígido que se mueve dentro del plano con restricciones no holonómicas, se plantea un modelo cinemático descrito por (1), cuyas variables de estado se representan como $x_c = [x_c \ y_c \ \psi_c]$ y entradas de control $u_c = [u_c \ \omega_c]$. La trayectoria de referencia es generada previamente para mantener las restricciones dinámicas en los giros del robot y consta de 5 elementos: posición en eje x (x_r), posición en eje y (y_r), ángulo de orientación (ψ_r), velocidad lineal (u_r) y velocidad angular (ω_r) – ver figura 5:

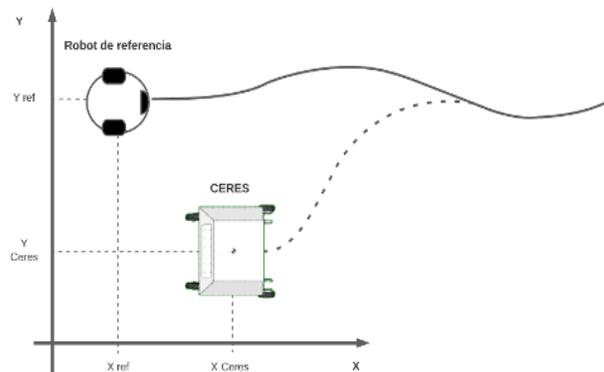


Figura 5. Modelo de referencia vs modelo CERES

La teoría de Taylor (linealización de primer orden) es usada para encontrar la representación lineal de la ecuación (1), en el punto de operación (x_r, u_r) , como se muestra en la ecuación (11).

$$\begin{aligned} \dot{x} = f(x_r, u_r) + \frac{\partial(x_c, u_c)}{\partial x_c} \Big|_{\substack{x=x_r \\ u=u_r}} (x_c - x_r) \\ + \frac{\partial(x_c, u_c)}{\partial u_c} \Big|_{\substack{x=x_r \\ u=u_r}} (u_c - u_r) \end{aligned} \quad (10)$$

Partiendo de la dinámica que el sistema tiene y la trayectoria referencia, se plantea un sistema que representa los errores del sistema para alcanzar la trayectoria referencia, quedando representado por la ecuación (11). Cabe resaltar que este modelo está representado en forma discreta y usando el método de Euler hacia adelante, con un tiempo de muestreo T_s [12].

$$\begin{aligned} \bar{x}_k(k+1) &= \bar{A}_k(k)\bar{x}_k(k) + \bar{B}_k(k)\bar{u}_k(k) \\ \hat{y}(k+1) &= \bar{C}_k(k)\bar{x}_k(k) \end{aligned} \quad (11)$$

donde:

$$\bar{A}_k(k) = \begin{bmatrix} 1 & 0 & -u_k(k) \operatorname{sen}(\psi_r(k))T_s \\ 0 & 1 & u_k(k) \operatorname{cos}(\psi_r(k))T_s \\ 0 & 0 & 1 \end{bmatrix} \quad (12)$$

$$\bar{B}_k(k) = \begin{bmatrix} \operatorname{cos}(\psi_r(k))T_s & 0 \\ \operatorname{sen}(\psi_r(k))T_s & 0 \\ 0 & T_s \end{bmatrix} \quad (13)$$

$$\bar{C}_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \quad (14)$$

$$\bar{x}_k = \begin{bmatrix} x_c(k) - x_r(k) \\ y_c(k) - y_r(k) \\ \psi_c(k) - \psi_r(k) \end{bmatrix} \quad (15)$$

$$\bar{u}_k = \begin{bmatrix} u_c(k) - u_r(k) \\ \omega_c(k) - \omega_r(k) \end{bmatrix} \quad (16)$$

Teniendo planteada la dinámica del error, se procede a integrar las restricciones físicas de los motores eléctricos con los que cuenta CERES, a la función objetivo de la ecuación (9). Dichas restricciones se plantean en forma de inecuaciones, como se presenta en la ecuación (17).

$$u_{\min} \leq u_c \leq u_{\max} \quad (17)$$

Debido a la linealización realizada en el modelo, es necesario traducir las restricciones a cambios en la señal de control (16), por lo cual deben ser expresadas como sigue:

$$\Delta u_{\min} \leq \Delta u_c(k) \leq \Delta u_{\max} \quad (18)$$

En función de ir a una forma cuadrática, es necesario separarlas en un límite inferior y un límite superior con signo opuesto, que de forma matricial toma la siguiente forma:

$$\begin{bmatrix} I \\ -I \end{bmatrix} \Delta u_c(k_i) \leq \begin{bmatrix} \Delta u_{\max} \\ -\Delta u_{\min} \end{bmatrix} \quad (19)$$

El valor de las restricciones está ligado a los valores nominales de máxima y mínima velocidad de los motores, que se reflejan en las acciones de control, con velocidad lineal (u) mínima y máxima de $1 \pm 1 \frac{m}{s}$ y velocidad angular (ω) mínima y máxima $\pm 1 \frac{rad}{s}$.

C. Resultados y simulación

Los resultados de la simulación aplicando el controlador de alto nivel tipo MPC al modelo de CERES son mostrados en esta sección. El robot inicia en la posición $x_r = [0 \ 0 \ 0]^T$ y se toma el modelo de CERES discretizado con un tiempo de muestreo de $T_s = 0.1$ s. El paquete **quadprog** de Matlab fue el encargado de resolver el problema de optimización, donde las matrices de ponderación R y Q fueron asumidas con valor unitario, asignando la misma importancia tanto al controlador como a los estados y, se asignaron los valores de $N_p = 5$ y $N_c = 3$.

La **primera prueba** trata de aplicar solo una velocidad traslacional de magnitud $0.75 \frac{m}{s}$ al modelo de referencia y desplazar al robot real CERES a una condición inicial de $x_c = [0 \ 1 \ \frac{\pi}{4}]^T$, con el fin de evaluar el control adecuado por parte del algoritmo MPC en 2 de las 3 variables de estado de CERES, así como las restricciones en la señal de control, los resultados se evidencian a continuación:

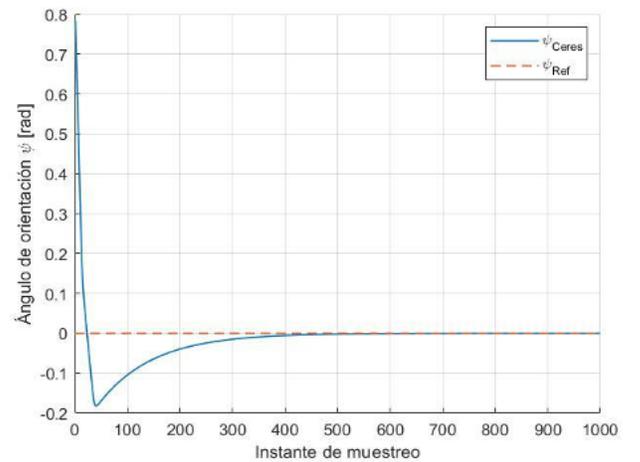


Figura 6. Comportamiento ángulo de orientación ψ en la prueba 1

Las figuras 6 y 7 presentan como el control MPC modifica el comportamiento de cada uno de los estados de CERES, haciendo que el robot converja asintóticamente a la trayectoria de referencia.

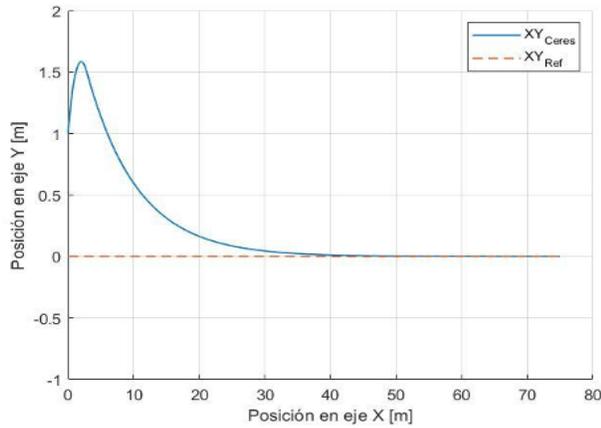


Figura 7. Trayectoria en el plano XY en la prueba 1

El algoritmo MPC va corrigiendo la posición en el eje Y y el ángulo de orientación de CERES, convergiendo a los mismos estados expuestos por la trayectoria de referencia como se propone en la Figura 5, lo que traduce en un seguimiento satisfactorio. Teniendo en cuenta las restricciones no holonómicas del modelo y solo un desplazamiento en la condición inicial en el eje Y, se espera que su trayectoria en el plano XY no tenga cambio alguno en X, pero tenga las mismas características de Y como se observa en la Figura 7.

Nótese que en la Figura 8, se muestran las señales de control que se emplearon para llegar a dicho seguimiento de trayectoria, donde tanto la velocidad lineal (u) y velocidad angular (ω), respetan los límites de las restricciones impuestas en la función de costo (9), por lo cual, la función de costo propuesta y optimizada por programación cuadrática converge a un valor de cero. Estos valores de velocidad lineal (u) y velocidad angular (ω) constituyen las entradas del control de bajo nivel para el seguimiento a la trayectoria, manteniendo la estructura jerarquizada de controladores de la Figura 4.

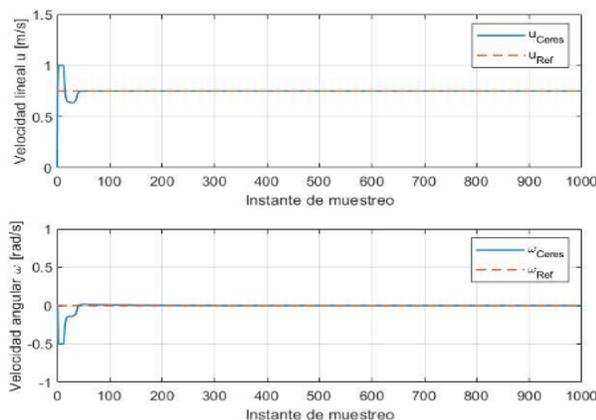


Figura 8. Señales de control MPC en la prueba 1

La **segunda prueba** tiene como objetivo evaluar el comportamiento del algoritmo MPC con una trayectoria mucho más elaborada, la cual tiene forma de espiral, donde entra en juego el control de las 3 variables de estado con las que cuenta CERES. Se mantienen como condiciones iniciales en $x_r = [0 \ 0 \ 0]^T$ para la trayectoria de referencia, pero CERES se desplaza a una condición inicial de $x_c = [0.5 \ 1 \ \frac{\pi}{10}]^T$ como se evidencia en el instante de muestreo inicial, en la Figura 9 y Figura 10.

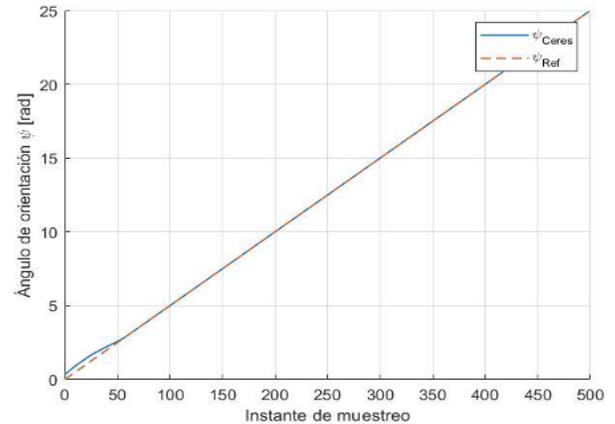


Figura 9. Comportamiento ángulo de orientación ψ en la prueba 2

El control MPC corrige la posición inicial x_c dada por $x_c = [0.5 \ 1 \ \frac{\pi}{10}]^T$ de CERES igualando asintóticamente su trayectoria con la de referencia, tal como se muestra en la Figura 10. Teniendo en cuenta la prueba previamente desarrollada las señales de control cumplen las restricciones de velocidad, por lo tanto, respetan la dinámica de la plataforma al ser aplicados a los controles de bajo nivel.

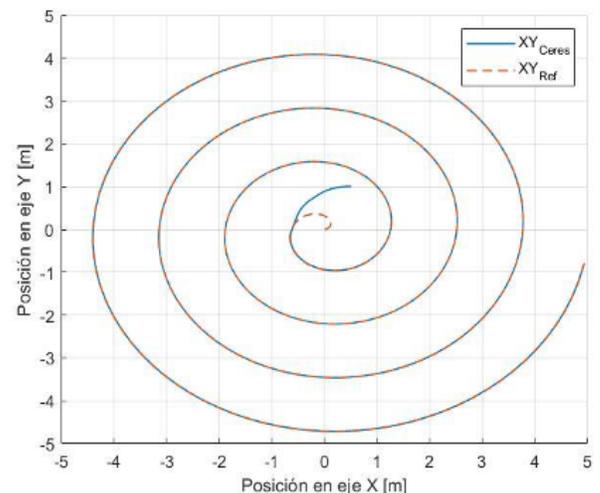


Figura 10. Trayectoria en el plano XY en la prueba 2

Para **evaluar** el comportamiento del sistema en las pruebas, se hallan los errores cuadráticos de las 3 variables de estado en cada instante de muestreo y se ilustra el resultado en las Figuras 11 y Figura 12, para la primera y segunda prueba respectivamente. Se evidencia que al aplicar el control GPC, el error cuadrático converge a cero, lo que se traduce a que el robot CERES sigue la trayectoria de referencia en posición XY y en ángulo de orientación.

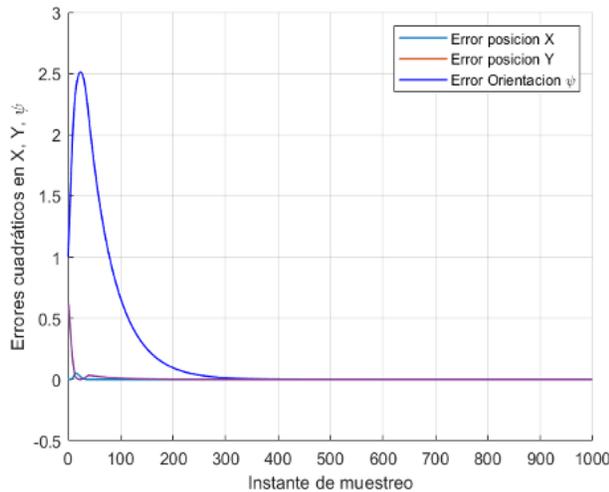


Figura 11. Errores cuadráticos en la prueba 1

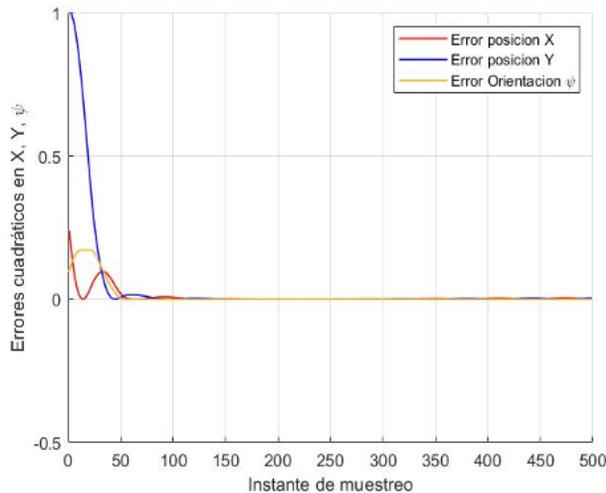


Figura 12. Errores cuadráticos en la prueba 2

La Tabla 1 y Tabla 2, resumen los errores cuadráticos medios y la varianza de cada variable de estado, tanto para la prueba 1 como para la prueba 2, con el fin de evaluar el comportamiento del error de seguimiento a la trayectoria en cada caso. De esta, se puede observar que son bajos los errores de seguimiento a lo largo de las trayectorias, considerando que las dimensiones del robot de 2 m × 2 m.

Tabla 1. Error cuadrático medio durante el seguimiento de la trayectoria en cada prueba

	Estados	Errores cuadráticos medios
Prueba 1	X	7.1485 E-04 m
	Y	0.1973 E-02 m
	ψ	0.0063 rad
Prueba 2	X	0.0083 m
	Y	0.0418 m
	ψ	0.0116 rad

Tabla 2. Varianza durante el seguimiento de la trayectoria en cada prueba

	Estados	Varianza
Prueba 1	X	6.9951 E-04 m ²
	Y	0,015 m ²
	ψ	0.0061 rad ²
Prueba 2	X	0.0083 m ²
	Y	0.0395 m ²
	ψ	0.0105 rad ²

III. Conclusiones

Dentro del proceso de aseguramiento de una conducción segura para la plataforma robótica CERES, se implementó un control jerárquico que permitió independizar controles de bajo nivel y alto nivel. Los primeros aseguran un control directo de los motores que la propulsan, recibiendo señales de referencia de velocidades traslacionales y orientación del robot. El control de alto nivel es centrado sobre la teoría de MPC.

Los resultados arrojados por el control de alto nivel mostraron ser aceptables en función de las dinámicas impuestas por los reguladores de bajo nivel, permitiendo seguir una trayectoria de referencia y cumplir la misión. El tener dentro de la función objetivo integrados los errores de seguimiento y las saturaciones de las señales de control, permitieron ajustar controles apropiados para la plataforma CERES, pues de manera implícita se contempla la dinámica de la plataforma a cada evaluación de la función objetivo.

El análisis del error cuadrático medio sobre el seguimiento de la trayectoria, ponen de manifiesto que los controles llevan los errores a cero en comportamientos temporales aceptables en el marco de los tiempos gastados por el robot en alcanzar su estabilización.

La proyección de solución fue contemplada con sistemas basados en ROS, lo que permite extrapolar de manera directa e integrar, los resultados a la plataforma real.

IV. Trabajos futuros

Incorporar la evasión de obstáculos en la función de optimización, teniendo en cuenta los obstáculos que pueden presentarse durante una misión, vistos con el láser 3D con el que cuenta CERES, con el fin de penalizar y encontrar la mejor solución visto como un problema de programación cuadrática.

Probar sobre la plataforma real los controladores para el seguimiento de la trayectoria en diferentes tipos de terreno.

V. Agradecimientos

Este trabajo está soportado por el proyecto INV-ING-2990 titulado “Auto-conducción en la ejecución de misiones en escenarios externos, dinámicos y poco estructurados para un robot en configuración diferencial plataforma CERES – Agrobot.”, financiado por la Vicerrectoría de Investigaciones de la UMNG.

Referencias

- [1] J. B. Han, K. M. Yang, D. H. Kim y K. H. Seo, “A Modeling and Simulation based on the Multibody Dynamics for an Autonomous Agricultural Robot,” in *2019 7th International Conference on Control, Mechatronics and Automation (ICCMA)*, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 137–143. <https://doi.org/10.1109/ICCMA46720.2019.8988607>.
- [2] W. S. Barbosa, et al., “Design and Development of an Autonomous Mobile Robot for Inspection of Soy and Cotton Crops,” in *2019 12th International Conference on Developments in eSystems Engineering (DeSE)*, 2019, pp. 557–562. <https://doi.org/10.1109/DeSE.2019.00107>.
- [3] Z. Wang y A. Lambert, “ICSP based visual teach and repeat for outdoor car-like robot localization,” in *2018 10th Computer Science and Electronic Engineering (CEECE)*, 2018, pp. 60–65. <https://doi.org/10.1109/CEECE.2018.8674207>.
- [4] M. Reke, et al., “A Self-Driving Car Architecture in ROS2,” in *2020 International SAUPEC/RobMech/PRASA Conference*, 2020, pp. 1–6. <https://doi.org/10.1109/SAUPEC/RobMech/PRASA48453.2020.9041020>.
- [5] K. Min, et al., “SAE Level 3 Autonomous Driving Technology of the ETRI,” in *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 464–466. <https://doi.org/10.1109/ICTC46691.2019.8939765>.
- [6] “Weed recognition by SVM texture feature classification in outdoor vegetable crops images,” *Ingeniería e Investigación*, vol. 37, n.º 1, págs. 68–74, 2017. dirección: <https://doi.org/10.15446/ing.investig.v37n1.54703>.
- [7] L. E. Solaque G and G. Sanchez and A. Riveros, “Design and Test of a Path Tracking Controller to a High Capability Agricultural Vehicle,” in *Proceedings of the 5th International Conference on Mechatronics and Robotics Engineering*, New York, Association for Computing Machinery, 2019, pp. 60–64. <https://doi.org/10.1145/3314493.3314514>.
- [8] ROS.org, “Powering the worlds robots,” [Online]. Available: <https://www.ros.org>. [Último acceso: 14 marzo 2021].
- [9] “The Autoware Foundation – Open Source for Autonomous Driving,” [Online]. Available: <https://www.autoware.org/>. [Último acceso: 14 marzo 2021].
- [10] L. E. Solaque G and G. Sánchez and A. Riveros, “Diseño y pruebas con hardware en el lazo de controladores dinámicos para un robot agrícola,” *Revista UIS Ingenierías*, vol. 19, n.º 2, pp. 89–98, 2020. <https://doi.org/10.18273/revuin.v19n2-2020010>.
- [11] S. Rey Serrano, *Ingeniería Basada en Modelos de un Controlador MPC para un Robot Movil*. Madrid: Universidad Politécnica de Madrid, 2016. http://www.ece.ufrgs.br/fetter/mechrob04_553.pdf.
- [12] F. Kühne and W. Fetter Lages and J. M. Gomes da Silva Jr., “Model Predictive Control of a Mobile Robot Using Linearization,” in *Proceedings of mechatronics and robotics*, 2004, pp. 525–530. http://www.ece.ufrgs.br/fetter/mechrob04_553.pdf.
- [13] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB*. London: Springer, 2009. <https://doi.org/10.1007/978-1-84882-331-0>.