

Controlador PI² para las velocidades de un robot agrícola evaluado usando Hardware en el lazo (HIL)

PI² controller for the velocities of an agricultural robot evaluated using Hardware in the loop (HIL)

Guillermo Sánchez Herrera¹
Leonardo Enrique Solaque Guzmán²
Adriana Riveros Guevara³

¹Universidad Militar Nueva Granada, Bogotá, Colombia. Email: guillermo.sanchez@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: 08-24-2020.

Revisado: 26-11-2020.

Aprobado: 24-03-2021.

Doi: 10.21500/20275846.5071

Referenciar así:

G. Sánchez Herrera, L. E. Solaque Guzmán and A. Riveros Guevara, "Controlador PI² para las velocidades de un robot agrícola evaluado usando Hardware en el lazo (HIL)," *Ingenierías USBMed*, vol. 12, n.º 2, pp. 41-48, 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. La agricultura es considerada como una de las actividades económicas, sociales y ambientales más esenciales para el ser humano debido principalmente a que por medio de esta se obtienen los alimentos. Con el objetivo de aumentar la producción y de mejorar el estado de los alimentos e incluso la salud de los agricultores se han empezado a usar diferentes robots, en lo que se conoce actualmente como agricultura de precisión. Con este fin se desarrolló el robot para agricultura Ceres, diseñado y construido para asistir principalmente cultivos de hortalizas. En este documento, se muestran los diferentes pasos que se desarrollaron para lograr el seguimiento de las velocidades lineales y angulares por parte del robot mediante el uso de un controlador dinámico. En primer lugar, se muestra la modelación matemática del robot y la identificación de los diferentes parámetros del modelo. Posteriormente, se encuentra la linealización del modelo y dado el acoplamiento entre velocidad lineal y angular, el diseño de desacopladores dinámicos en un punto de operación determinado por las referencias a seguir. Finalmente, se utiliza un controlador PI con doble acción integral y se programa en un sistema embebido verificando su desempeño mediante pruebas de hardware en el lazo (HIL). Los resultados muestran tiempos de estabilización y seguimientos adecuados para futuras integraciones con la planta real y controladores de trayectoria.

Palabras Clave. Robótica, control, modelado, identificación, Hardware in the loop.

Abstract. Agriculture is considered as one of the most essential economic, social, and environmental activities for human beings, mainly because food is obtained through it. To increase production and improve the state of food and even the health of farmers, different robots have started to be used, in what is currently known as precision agriculture. For this purpose, the Ceres agricultural robot was designed and built to mainly assist vegetable crops. In this document, the different steps that were developed to achieve the tracking of linear and angular velocities by the robot using a dynamic controller are shown. First, the mathematical modeling of the robot and the identification of the model parameters are shown. Subsequently, the linearization of the model is found, and given the coupling between linear and angular velocities, the design of dynamic decouplers at an operating point is determined by the references to follow. Finally, a PI controller with an integral double action is used and it is programmed in an embedded system to verify the controller performance using hardware in the loop (HIL) tests. The results show stabilization times and an adequate input following for future integrations with the real plant and trajectory controllers.

Keywords. Robotic, Control, Modeling, Identification, Hardware in the loop.

I. Introducción

El uso de robots móviles ha crecido en los últimos años, abarcando temáticas como la atención médica, atención al usuario, entre otras [1]. Sin embargo, un área clave para la supervivencia de la raza humana es la agricultura, [2] la cual enfrenta grandes retos como la producción eficiente, el aumento de la producción asociado al crecimiento poblacional y el cambio climático [3].

En la agricultura, la robótica ha realizado avances importantes, no obstante, las dificultades del terreno, los cambios frecuentes de temperatura, humedad y la irregularidad del terreno, hacen de los robots agrícolas un desafío de ingeniería [4], con grandes retos desde el diseño, la integración, el sensado, la locomoción, la navegación y finalmente la ejecución de labores agrícolas en campo.

Los robots agrícolas pueden ser estudiados desde dos perspectivas: En primer lugar, la locomoción que permita el desplazamiento del vehículo en el cultivo de forma segura. Para lograr ese objetivo se deben tener en cuenta factores como la irregularidad del terreno, los sensores a utilizar y los algoritmos para conseguir el objetivo; en segundo lugar, el robot debe realizar la tarea o tareas principales sobre el cultivo (cosechar, fumigar, remover maleza, entre otras) [5].

La locomoción de robots móviles implica el seguimiento de comandos de desplazamiento por parte del robot, estos comandos se han abordado desde diferentes enfoques, el más común es el seguimiento de coordenadas GPS, en Ball et al. [6] se realiza la modificación a un vehículo pequeño, el cual es dotado de varios GPS redundantes y cámaras que permiten realizar el seguimiento de las líneas del cultivo detectando obstáculos presentes en el entorno. No obstante, iniciativas comerciales como Thorvald [7] también han implementado algoritmos de navegación configurables según la estructura solicitada por el cliente, permitiendo el seguimiento de surcos y la adecuación de herramientas para remoción de maleza, toma de muestras o recolección de frutas, según las necesidades. Esta última plataforma cuenta con integración completa con el sistema operativo para robótica ROS, permitiendo realizar simulaciones, registro de datos e inclusive añadir funcionalidades con este software libre [8].

La simulación HIL (*Hardware in the Loop*), es una herramienta usada ampliamente en el área de control que permite verificar la ejecución del algoritmo en condiciones semi reales ejecutándose en el hardware a utilizar en la implementación final. Este hardware es intercomunicado con una planta simulada para verificar su desempeño y tiempos de ejecución. Estas simulaciones son usadas ampliamente en el sector automovilístico con lo cual se reducen costos y tiempos de desarrollo [9].

En este artículo se detalla el proceso de diseño y simulación de controladores PI cuadrado de locomoción

para un robot agrícola, el cual cuenta con herramientas para la remoción de maleza, fertilización y fumigación, teniendo una carga útil total de 150 kg; previo al proceso de diseño se detalla la modelación matemática del robot y el ajuste de los parámetros del modelo. Por último, se realiza la programación y simulación del controlador diseñado haciendo uso de una arquitectura de simulación *Hardware in the Loop*, donde se simula la planta mientras el controlador es ejecutado en el hardware final.

II. Procedimiento

En esta sección se detalla el proceso realizado desde la modelación matemática hasta el diseño de los controladores desacoplados, para el robot agrícola con el fin de dotar de locomoción al vehículo.

El robot Ceres cuenta con un volumen de trabajo de: 0.585 m^3 , para realizar labores de cuidado del cultivo, con un rango de movimiento de 1.3 m, 0.9 m y 0.5 m en los ejes X , Y , y Z respectivamente. En la Figura 1 se puede ver el robot construido, el cual posee una capacidad de carga en tolva de 100 Kg de abono sólido, y una capacidad en tanques de 20 litros de plaguicida o nutrientes. El robot es propulsado por motores eléctricos de 48 V y 5 KW e instrumentado con IMU, GPS, cámaras multiespectral, térmica y estéreo, sensores láser 3D y 2D. La asistencia al estado de salud se realiza mediante un sistema alojado en la parte inferior del robot que posee actuadores lineales en los tres ejes para la realización de tareas puntuales tales como aspersión de plaguicida, remoción de maleza o aspersión de nutrientes.



Figura 1. Robot eléctrico (48 V) Ceres, con capacidad de carga de 100 Kg de abono sólido y 20 litros de plaguicida

Previo al diseño de los controladores de locomoción, se define un modelo dinámico a partir de las características geométricas y dinámicas del vehículo. Los controladores obtenidos son probados en simulación de Hardware en lazo, mediante el uso de Matlab y ROS.

A. Modelado cinemático

El primer análisis realizado es el cinemático, donde se define como punto de referencia el centro de masa del robot, al cual se le asigna la coordenada C, sin embargo, debido a la geometría del robot, otro punto de interés para el cálculo de la posición del robot es el centro del eje de movimiento (A), el cual pasa por las dos ruedas de locomoción (en este caso las frontales), por su parte el ángulo (θ) indica la orientación del robot en el marco de referencia R.

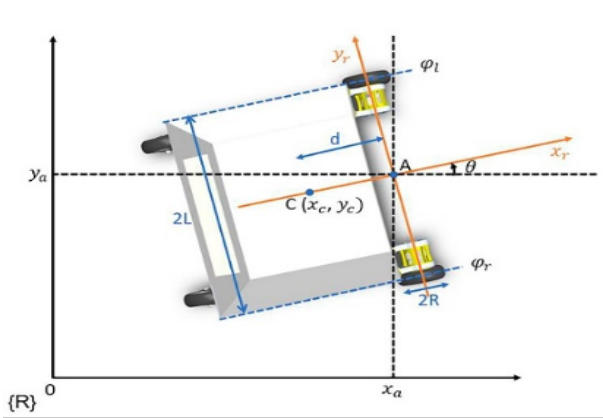


Figura 2. Marcos referencia robot Ceres

Las ecuaciones cinemáticas de un robot móvil diferencial tienen en cuenta su característica no-holónómica (la dirección de avance siempre es en el sentido de las ruedas), las ecuaciones principales se presentan a continuación, note la distancia d (entre el centro de tracción y el centro de gravedad), la cual cambia la cinemática del robot.

$$\begin{aligned} \dot{x}_c &= v \cos(\theta) - d\omega \sin(\theta) \\ \dot{y}_c &= v \sin(\theta) - d\omega \cos(\theta) \\ \dot{\theta} &= \omega \end{aligned} \quad (1)$$

Estas ecuaciones permiten obtener la ubicación en el plano XY en función de la velocidad lineal (v) y angular (ω). Sin embargo, para propósitos de control es de interés conocer las ecuaciones anteriores en términos de las velocidades angulares de cada una de las llantas, lo cual se consigue mediante las relaciones:

$$\begin{aligned} v &= \frac{v_{rr} + v_{rl}}{2} = R \frac{\dot{\varphi}_r + \dot{\varphi}_l}{2} \\ \omega &= \frac{v_{rr} - v_{rl}}{2L} = R \frac{\dot{\varphi}_r - \dot{\varphi}_l}{2L} \end{aligned} \quad (2)$$

Siendo v_{rx} las velocidades tangenciales de la rueda derecha e izquierda y $\dot{\varphi}_x$ la velocidad angular de cada una de las ruedas de tracción.

B. Modelado dinámico

Partiendo de las dimensiones del robot y la carga útil máxima, es de vital importancia tener en cuenta

los efectos dinámicos del movimiento, por tanto, una aproximación Euler-Lagrange es usada [10], [11].

Debido al enfoque energético de esta teoría, se definen las energías potencial y cinética del robot. En el primer caso, debido a que no se presentan cambios de altura del robot, la energía potencial se asume como cero. La energía cinética, se asume partiendo de tres divisiones, en primer lugar, la energía de la estructura del robot, la energía cinética (T_c) en (3), energía cinética de la rueda y actuador derecho (T_{wr}) y, por último, la energía cinética del conjunto rueda actuador izquierdo (T_{wl}).

$$\begin{aligned} T_c &= \frac{1}{2} m_c v_c^2 + \frac{1}{2} I_c \dot{\theta}^2 \\ T_{wr} &= \frac{1}{2} m_w v_{wr}^2 + \frac{1}{2} I_m \dot{\theta}^2 + \frac{1}{2} I_w \dot{\varphi}_r^2 \\ T_{wl} &= \frac{1}{2} m_w v_{wl}^2 + \frac{1}{2} I_m \dot{\theta}^2 + \frac{1}{2} I_w \dot{\varphi}_l^2 \end{aligned} \quad (3)$$

Donde m_c es la masa del robot sin las ruedas y actuadores, m_w es la masa de la rueda y el motor e I_c es el momento de inercia del robot con respecto al eje vertical del centro de masa. I_m el momento de inercia del motor con respecto al diámetro de la rueda e I_w el momento de inercia de la rueda con el actuador, respecto al eje de la rueda. Adicionalmente, θ es la posición angular del robot, φ_r es la posición angular de la rueda derecha, φ_l posición angular de la rueda izquierda, v_c la velocidad del cuerpo del robot, las velocidades v_{wr} y v_{wl} corresponden con las velocidades de la rueda y el actuador.

Las masas de los elementos son obtenidas de las hojas de datos y el modelo CAD (estos son mostrados en la Tabla 1) y las velocidades de los conjuntos rueda motor se deducen asumiendo un cuerpo rígido y conociendo la velocidad del centro de gravedad del robot y las respectivas distancias a cada uno de los conjuntos.

Lo anterior permite llegar al modelo dinámico que se muestra en (4) y (5), reemplazando en las derivadas de las velocidades angular y lineal.

$$\begin{aligned} \dot{v} &= \frac{-1}{2I_w + mR^2} (c_3 v - R\tau_r - R\tau_l + c_4 v + Lc_3 \omega - Lc_4 \omega \\ &\dots + Rc_1 \text{sign}\left(\frac{v+L\omega}{R}\right) + Rc_2 \text{sign}\left(\frac{v-L\omega}{R}\right) \\ &+ dm_c R^2 \omega^2) \end{aligned} \quad (4)$$

$$\begin{aligned} \dot{\omega} &= \frac{1}{I + \frac{2I_w L^2}{R^2}} \left[dm_c v \omega + \frac{L}{R} (\tau_r - \tau_l) \dots \right. \\ &- \frac{L}{R} \left(c_1 \text{sign}\left(\frac{v+l\omega}{R}\right) + \frac{c_3(v+L\omega)}{R} - \frac{c_4(v-L\omega)}{R} \right. \\ &\left. \left. - c_2 \text{sign}\left(\frac{v-L\omega}{R}\right) \right) \right] \end{aligned} \quad (5)$$

Tabla 1. Coeficientes del sistema

Símbolo	Coefficiente	Valor	Unidades
m_c	Masa del vehículo sin ruedas ni motores	554.46	Kg
m_w	Masa del conjunto rueda y motor	40	Kg
I_c	Momento de inercia del robot con respecto al eje vertical del centro de masa	474.06	Kg m ²
I_m	Momento de inercia de la rueda con actuador con respecto al diámetro de la rueda	4.59	Kg m ²
I_w	Momento de inercia de la rueda con actuador con respecto al eje de la rueda	34	Kg m ²
d	Distancia entre el centro de masa y el centro del eje que une las ruedas	0.9	m
R	Radio de la rueda	0.28	m
L	Mitad de la distancia entre ruedas	1.1275	m

Los parámetros mostrados en la Tabla 2 son calculados haciendo uso de técnicas de identificación paramétrica con datos capturados de pruebas experimentales, en estas pruebas se registraron los de voltajes de entrada y las velocidades de salida del robot. Los coeficientes hallados fueron: la constante de proporcionalidad del motor izquierdo y derecho (k_1 y k_2), el coeficiente de fricción de Coulomb del motor derecho e izquierdo (c_1 y c_2) y el coeficiente de fricción viscosa del motor derecho e izquierdo (c_3 y c_4).

Tabla 2. Parámetros identificados del sistema

Símbolo	Valor	Desviación estándar	Unidades
k_1	575.211	10.2158	Nm/V
k_2	594.24	12.9673	Nm/V
c_1	895.012	21.1102	Nms
c_2	879.152	16.9086	Nms
c_3	57.7522	1.69425	Nms
c_4	57.4151	1.2875	Nms

C. Linealización del modelo y desacople de dinámicas

Teniendo en cuenta que el sistema tiene dos entradas (v_l y v_r) y dos salidas (v y ω), y que a su vez estas se encuentran acopladas, se decide realizar un desacoplador de estados para tratar el sistema MIMO como dos sistemas SISO, lo que facilita el diseño y la implementación del controlador [12].

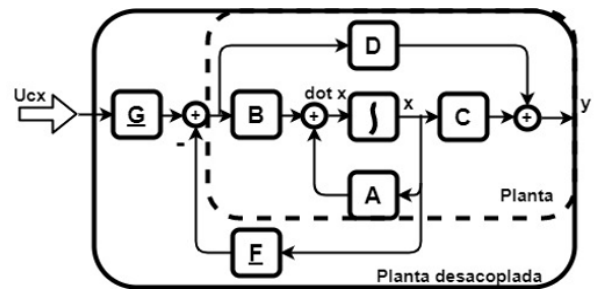
En busca de obtener funciones de transferencia desacopladas entre sí, se realiza una linealización del sistema en un punto de operación, estos puntos se eligen considerando inicialmente los puntos de equilibrio del sistema, los cuales se obtienen cuando las derivadas son cero en el espacio de estados.

El punto de operación permite linealizar el sistema por medio del jacobiano, obteniendo las ecuaciones mostradas a continuación:

$$\begin{aligned} \begin{bmatrix} \dot{v} \\ \dot{\omega} \end{bmatrix} &= \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} + \begin{bmatrix} e & e \\ -f & f \end{bmatrix} \begin{bmatrix} v_l \\ v_r \end{bmatrix} \\ \begin{bmatrix} v \\ \omega \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_l \\ v_r \end{bmatrix} \end{aligned} \quad (6)$$

Donde $a = -0.98$, $b = -0.66\bar{\omega} - 0.0032$, $c = 0.23\bar{\omega} - 0.0023$, $d = 0.23\bar{v} - 0.87$, $e = 1.4$ y $f = 1.1$.

Las anteriores matrices indican el acople de las variables, sin embargo, con el fin de simplificar el controlador, se decide separar este sistema de forma que el voltaje del motor izquierdo controle la velocidad lineal y el voltaje del motor derecho la velocidad angular. Para este desacople se hace uso de matrices adicionales que permitan considerar la planta desacoplada, en la Figura 3 se muestra el diagrama de bloques del sistema, en donde las matrices F y G son las matrices de desacople, estas matrices están representadas por las ecuaciones (??) y (11) y dependen de los puntos de operación definidos para la velocidad lineal (\bar{v}) y para la velocidad angular ($\bar{\omega}$).


Figura 3. Diagrama de bloques del desacoplador de estados

Aplicando las fórmulas propuestas por Espinoza [13] para obtener las matrices de desacople, se debe verificar la no singularidad de la matriz E , la cual se presenta en (7), comprobada la no singularidad se procede al cálculo de las matrices \underline{B} , \underline{C} , \underline{E} y \underline{G} como se muestra en (8), (9), (10) y (11).

$$E = \begin{bmatrix} C_1 A^{v_1-1} B \\ C_2 A^{v_2-1} B \\ \vdots \\ C_q A^{v_q-1} B \end{bmatrix} \quad (7)$$

$$\underline{B} = \underline{E} = \begin{bmatrix} 1.391 & 1.391 \\ -1.102 & 1.102 \end{bmatrix} \quad (8)$$

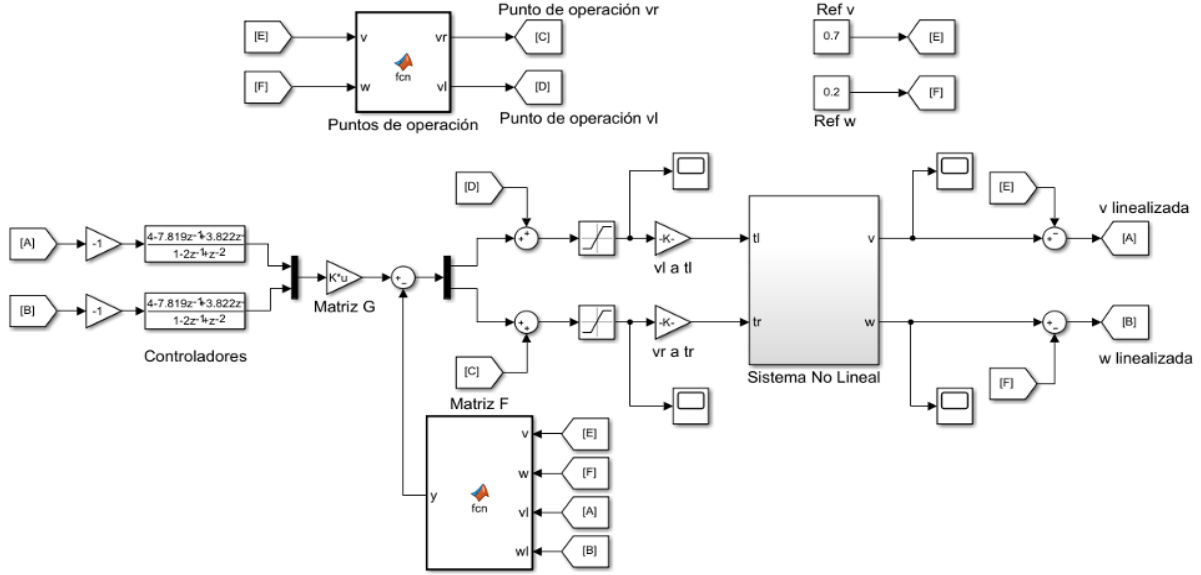


Figura 4. Esquema de simulación del controlador

$$\underline{C} = \begin{bmatrix} C_1 A^{v_1} \\ C_2 A^{v_2} \\ \vdots \\ C_q A^{v_q} \end{bmatrix} \quad (9)$$

$$\underline{F} = \underline{B}^{-1} \underline{C} = \begin{bmatrix} -0.1\bar{w} - 0.3 & -0.2\bar{w} - 0.01\bar{v} + 0.4 \\ 0.1\bar{w} - 0.3 & -0.2\bar{w} + 0.1\bar{v} - 0.4 \end{bmatrix} \quad (10)$$

$$\underline{G} = \underline{B}^{-1} = \begin{bmatrix} 0.3596 & -0.4537 \\ 0.3596 & 0.4537 \end{bmatrix} \quad (11)$$

Por último, se define la función de transferencia del lazo cerrado resultante, la cual se muestra en la ecuación (12):

$$H_{CL} = \begin{bmatrix} s^{-v_1} & 0 \\ 0 & s^{-v_2} \end{bmatrix} = \begin{bmatrix} s^{-1} & 0 \\ 0 & s^{-1} \end{bmatrix} \quad (12)$$

De lo anterior, se tiene que el sistema resultante es un integrador puro.

D. Controlador PI Cuadrado

Con el fin de asegurar error en estado estable igual a cero para una entrada parábola y considerando que se tiene una función de transferencia que describe un sistema tipo 1, se escoge un controlador tipo PI cuadrado o controlador proporcional con doble acción integral, para que realice el seguimiento a este tipo de entrada [14]. Así, la función de transferencia que describe al controlador es presentada en (13).

$$C(s) = \frac{K_p s^2 + K_i s + K_{i2}}{s^2} \quad (13)$$

Como parámetros de diseño se establece un tiempo de establecimiento igual a 7 segundos, tiempo similar a la respuesta del sistema en lazo abierto, y un coeficiente de amortiguamiento cercano a 1 con el fin de obtener una respuesta críticamente amortiguada. La determinación de los valores de las acciones de control se realiza por la técnica de asignación de polos, obteniendo los siguientes valores: $K_p = 4$, $K_i = 3.5918$, $K_{i2} = 0.9329$.

Finalmente, el desempeño del controlador es verificado en Matlab, comprobando la linealización y desacople en el punto de operación acorde a la referencia dada. El esquema de simulación se muestra en la Figura 4.

III. Simulaciones Hardware en el lazo

Con el fin de realizar la simulación HIL en condiciones reales, se debe tener en cuenta que el robot Ceres y sus sensores operan con el sistema operativo para robótica ROS. Los controladores de velocidades diseñados previamente son discretizados con un tiempo de muestreo de $t = 0.05$ s y programados en un Arduino Mega el cual se conecta por comunicación serial al computador principal que corre el nodo principal de ROS. En esta subsección se describe el software, hardware y el flujo de datos utilizado para la prueba de los controladores diseñados.

El *middleware* ROS provee una capa de comunicación entre sistemas operativos Linux y sistemas embebidos (Ejemplo: Computador y uno o varios microcontroladores). Como características principales brinda comunicación de igual a igual (peer to peer), es programado en C++ y en Python, permite comunicar los programas en esos lenguajes en un mismo proyecto. ROS cuenta con herramientas muy útiles para facilitar la visualización, supervisión de tiempos de ejecución

y dato. Finalmente, ROS es de código abierto bajo la licencia BSD y cuenta con una amplia comunidad, lo cual facilita la corrección de errores y el soporte. Los detalles de su filosofía pueden ser consultados en [8] y los tutoriales oficiales están disponibles en www.ros.org.

En la Figura 5 se puede observar el hardware con el que cuenta el robot para la locomoción, los motores eléctricos de 48v son actuados mediante una señal analógica de 0 a 5 V, la IMU (Unidad de medida inercial) mediante la librería del fabricante es integrada a ROS, permitiendo acceder a los estados del robot vía comunicación serial.

La simulación del sistema propuesto se hace sobre la base de tener el moldeo en un PC del robot Ceres en Matlab y mediante comunicación serial, se enlaza con el Arduino, quien aloja los controles presentados en la sección II-D.



Figura 5. Hardware de locomoción del robot Ceres

Las referencias de velocidad lineal y velocidad angular son generadas desde ROS y sobre él, se hace el flujo de datos y lanzamiento de funciones (nodos), de manera que el Arduino realice el cálculo de las señales de control para los motores del robot, que son enviadas en forma de voltaje (referencias para los manejadores de potencia de los motores que reciben señales en voltaje). Con el objetivo de cerrar el lazo de control, se reciben estas señales en el PC y se aplican al modelo del robot que corre en simulación. El modelo retorna la velocidad alcanzada tanto lineal como trotacional, y son las que se envían al Arduino para generar la señal de error, cerrando el lazo para el respectivo cálculo de la señal de control. En la Figura 6 se muestra el esquema de simulación utilizado.

IV. Resultados

En las siguientes figuras se muestra el resultado comparativo entre desempeño ideal esperado en la simulación realizada en Matlab y el desempeño obtenido con el controlador programado en el sistema embebido. Las pruebas se realizan a un 33% y un 66% de la velocidad máxima de trabajo del robot agrícola Ceres.

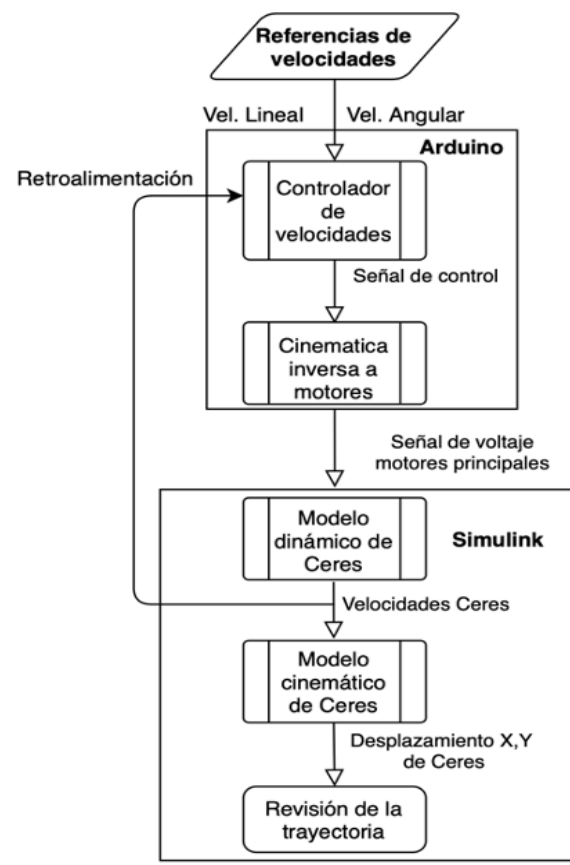


Figura 6. Diagrama de flujo del procedimiento para la simulación usando Hardware in the Loop

La Figura 7 y la Figura 8 muestran la respuesta a escalones asíncronos en las velocidades lineal y angular, mostrando un acoplamiento reducido evidenciado en 3.3 segundos para la velocidad lineal y 1.5 segundos para la velocidad angular.

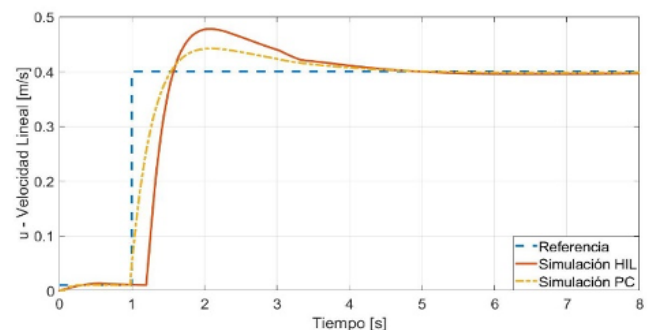


Figura 7. Respuesta a un escalón de amplitud 0.4 m/s en la velocidad lineal

Los voltajes ideales y obtenidos en la simulación HIL son mostrados en la Figura 9, en este caso el acoplamiento se hace más evidente debido a la cinemática inversa aplicada antes del envío de la señal a los motores, sin embargo, la señal no se satura para los rangos a utilizar en el robot Ceres.

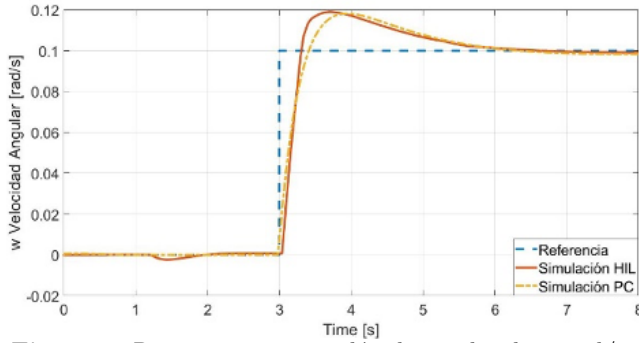


Figura 8. Respuesta a un escalón de amplitud 0.1 rad/s en la velocidad angular

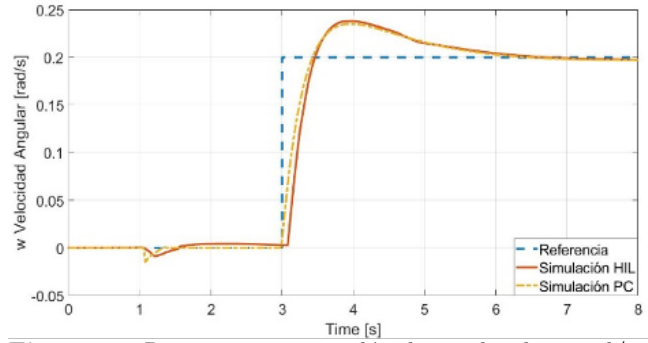


Figura 11. Respuesta a un escalón de amplitud 0.2 rad/s en la velocidad angular

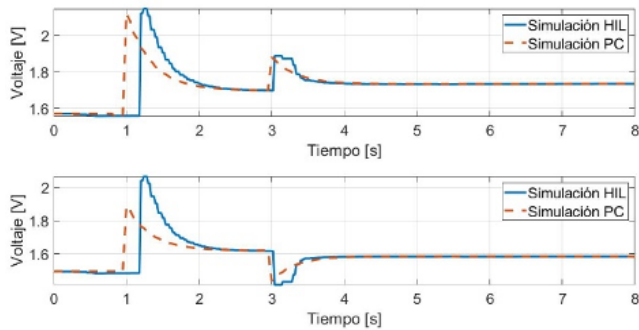


Figura 9. Voltajes calculados por el controlador para la señal de control de los motores de locomoción

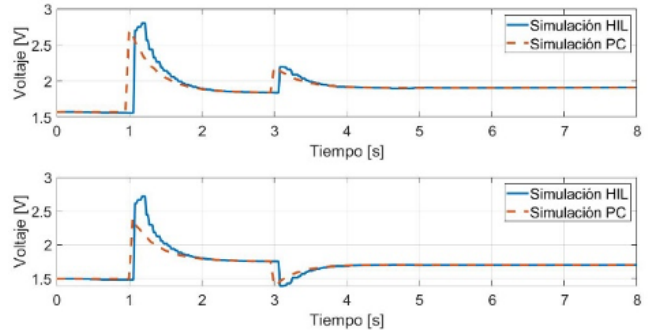


Figura 12. Voltajes calculados por el controlador para la señal de control de los motores de locomoción

Las pruebas se repiten para el doble de las velocidades mostradas anteriormente, la Figura 10 muestra las velocidades lineales obtenidas con un seguimiento adecuado. De manera similar la Figura 11 muestra las velocidades angulares obtenidas en las pruebas.

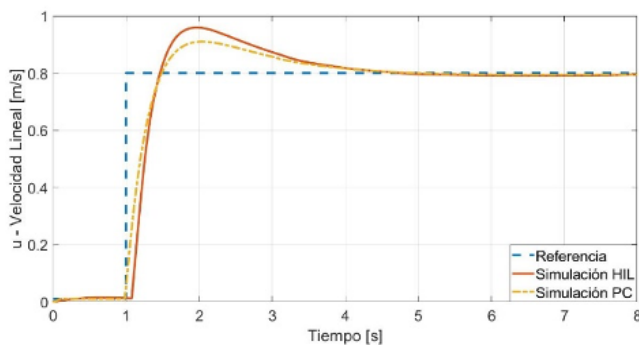


Figura 10. Respuesta a un escalón de amplitud 0.8 m/s en la velocidad lineal

Por último, los voltajes obtenidos para la segunda prueba son mostrados en la Figura 12, con un aumento de su valor absoluto y valor final, no obstante, estos valores se mantienen alejados del valor máximo de 5V permitido por los motores de locomoción.

V. Conclusiones

Este artículo presentó la determinación de un modelo dinámico para un robot en configuración diferencial, como lo es CERES. A partir de este modelo se identificaron sus parámetros, llegando a mostrar que, para su representación lineal, se tienen acoples de dinámicas. Lo anterior se enfrentó mediante el diseño de desacopladores dinámicos según el punto de operación, los cuales se implementaron y proporcionaron modelos directos entre las variables de velocidad traslacional y rotacional, siendo estas últimas los objetivos de control.

Como control, se implementó un control tipo PI con doble acción integral para llegar a hacer seguimiento a señales tipo parábola. Para este diseño se tomaron como parámetros el tiempo de estabilización similar al de lazo abierto de la plataforma y una respuesta críticamente amortiguada.

Para la prueba de los controladores, se usó la aproximación por simulaciones en hardware HIL, donde el modelo se simuló en un computador bajo el software Matlab. Se estableció el control en un sistema de microprocesador, tal como se implementa directamente sobre la plataforma real, mostrando un buen desempeño. Sin embargo, se puede observar de las gráficas de la sección ‘Resultados’ que los reguladores trabajando en el sistema a embarcar sobre la plataforma, propor-

cionan respuestas del tipo sub-amortiguadas y manteniendo la estabilidad de la plataforma. El uso de estas técnicas de simulación permite flexibilidad y disminución de errores de cara a pruebas con las plantas reales, aspectos de vital relevancia en sistemas mecatrónicos complejos y de alto costo como el robot Ceres.

VI. Trabajos futuros

Con los resultados y la experimentación realizada, deja abierta la puerta a seguir pruebas directas sobre la plataforma CERES y en el terreno de cultivos de hortalizas. El siguiente objetivo para el proyecto será evaluar el desempeño de los controladores tras implementarlos en la plataforma y realizar las modificaciones necesarias que permitan un funcionamiento adecuado en campo.

Posteriormente, se buscará asegurar la autonomía del robot tal que permita la realización de labores dirigidas al cuidado del cultivo, focalizadas en la detección de deficiencias en los cultivos y aplicación de medidas correctivas. Con el fin de afrontar los problemas de sanidad que pueda tener el cultivo, se podrán realizar cambios en los mecanismos del robot, que permitan un mejor funcionamiento.

Agradecimientos

Este trabajo es soportado por el proyecto INV-ING-2637, titulado “Aproximación a la Autonomía de una plataforma robótica dedicada a labores de agricultura de precisión”, financiado por la Vicerrectoría de Investigaciones de la Universidad Militar Nueva Granada en Bogotá-Colombia.

Referencias

- [1] J. Wirtz et al., “Brave new world: service robots in the frontline,” *Journal of Service Management*, vol. 29, n.º 5, pp. 907–931, 2018. <https://doi.org/10.1108/JOSM-04-2018-0119>.
- [2] S. Hendricks, “The food security continuum: a novel tool for understanding food insecurity as a range of experiences,” *Food Security*, n.º 7, pp. 609–619, 2015. <https://doi.org/10.1007/s12571-015-0457-6>.
- [3] M. Karkee y Q. Zhang, “Mechanization and automation technologies in specialty crop production,” *Resource Magazine*, vol. 19, n.º 5, pp. 16–17, 2012. <https://elibrary.asabe.org/abstract.asp?aid=42314>.
- [4] S. Hiremath et al., “Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter,” *Computers and Electronics in Agriculture*, vol. 100, pp. 41–50, 2014. <https://doi.org/10.1016/j.compag.2013.10.005>.
- [5] A. Bechar, “Robotics in horticultural field production,” *Stewart Postharvest Review*, vol. 6, n.º 3, pp. 1–11, 2010. <https://doi.org/10.2212/spr.2010.3.11>.
- [6] D. Ball et al., “Vision-based Obstacle Detection and Navigation for an Agricultural Robot,” *Journal of Field Robotics*, vol. 33, n.º 8, pp. 1107–1130, 2016. <https://doi.org/10.1002/rob.21644>.
- [7] L. Grimstad y P. J. From, “The Thorvald II Agricultural Robotic System,” *Robotics*, vol. 6, n.º 4, p. 24, 2017. <https://doi.org/10.3390/robotics6040024>.
- [8] M. Quigley, B. Gerkey y W. Smart, *Programming Robots with ROS: A Practical Introduction to the Robot Operating System*, OReilly Media, 2015.
- [9] J. Khan, “A Standardized Process Flow for Creating and Maintaining Component Level Hardware in the Loop Simulation Test Bench,” *SAE Technical Paper*, 2016. <https://doi.org/10.4271/2016-01-0052>.
- [10] R. Dhaouadi and A. Abu Hatab, “Dynamic Modelling of Differential-Drive Mobile Robots using Lagrange and Newton-Euler Methodologies: A Unified Framework,” *Advances in Robotics and Animation*, vol. 2, n.º 2, pp. 107–113, 2013. <https://doi.org/10.4172/2168-9695.1000107>.
- [11] G. M. Andaluz Ortiz, *Modelación, identificación y control de robots móviles*, QUITO/EPN, 2011.
- [12] J. R. Espinoza C, *Síntesis Sistemas de Control*, Universidad de Concepción, 2003.
- [13] J. R. Espinoza C, *Apuntes Sistemas de Control — 543 244*, Universidad de Concepción, 2020. http://www2.udec.cl/jose.espinoza/SDC/543244_SDC_2020_2_Apuntes.pdf.
- [14] E. Solera Saborío and V. M. Alfaro Ruíz, “Sintonización de controladores PI y PID utilizando modelos de polo doble más tiempo muerto,” *Ingeniería: Revista de la Universidad de Costa Rica*, vol. 16, n.º 2, págs. 23–31, 2006.