

APLICACIÓN DE CONTROLADORES DE VELOCIDAD EN LLANTAS HOLONÓMICAS, CONTROL DE POSICIONAMIENTO Y PROTOCOLOS DE COMUNICACIÓN DE UN ROBOT JUGADOR DE FUTBOL

SPEED CONTROLLERS APPLICATION IN HOLONOMIC WHEELS, POSITIONING CONTROL AND PROTOCOLS IN A FOOTBALL PLAYER ROBOT COMMUNICATION

MSc. Tania Liseth Acevedo Gauta*, MSc. Andrés Orlando Páez Melo*
MSc. Joaquín Andrés Valderrama Rincón**

* **Universidad de Pamplona**, Facultad de Ingenierías y Arquitectura,
Grupos de Investigación A&C, GIMS.

Km1 vía Bucaramanga, Pamplona, Norte de Santander, Colombia.

E-mail: {tlacevedo, [andres.paez2](mailto:andres.paez2@unipamplona.edu.co)}@unipamplona.edu.co.

Orcid: <https://orcid.org/0009-0007-7824-4514>

Orcid: <https://orcid.org/0000-0002-7118-7111>

** **Universidad Antonio Nariño**, Facultad de Ciencias
Cl. 58a Bis #3794, Bogotá, Cundinamarca, Colombia

E-mail: jandvalderrama@uam.edu.co.

Orcid: <https://orcid.org/0009-0001-6112-4387>

Resumen: En este trabajo, se presenta un sistema de control digital de posición para un robot holonómico, basado en un sistema de localización compuesto por una fusión multisensorial entre la odometría del móvil y un sistema de visión artificial externo. El robot tiene un sistema embebido que se encuentra dentro del robot, el software funciona basado en un sistema operativo en tiempo real (FreeRTOS). Se presenta también el diseño de un control de velocidad para cada uno de los motores que controlan las ruedas, y un protocolo de comunicación ad-hoc para la comunicación (backplane) del mismo.

Palabras clave: Control, motores, posición, comunicación, robot.

Abstract: This work presents a digital position control system for a holonomic robot, based on a location system composed of a multi-sensorial fusion between the mobile odometry and an external artificial vision system. The robot has an embedded system that is inside it and the software works based on a real-time operating system (FreeRTOS). The design of a speed control for each engine that controls the wheels is also showed off, as well as, an ad-hoc communication protocol for its communication (backplane).

Keywords: Control, motors, position, communication, robot.

1. INTRODUCCIÓN

La robótica móvil ha despertado el interés del mundo para lograr que las máquinas desarrollen ciertas tareas. Los robots de pasa tiempo y de competición son una de las ramas de las disciplinas

que engloba la robótica y que más entusiasma a los seguidores, generar sistemas autónomos para la ejecución de tareas es el campo de acción donde se enmarca la robocopa, una liga que promueve la competencia entre robots autónomos donde se

aplican los resultados de investigaciones en las áreas de control e inteligencia artificial entre otros.

La SSL (robots de pequeño tamaño), como área de desarrollo de este campeonato se centran en el problema de acoplamiento de sistemas multiagente y el control de variables, para este tipo de entornos dinámicos con un sistema híbrido centralizado, aplicado a un robot móvil o jugador y que cumpla con las reglas y medidas establecidas por la competencia (The Robocup Federation [Online], n.d.). Diferentes trabajos plantean tres procesos generales importantes en este tipo de aplicaciones, la adquisición de datos, implementación de inteligencia artificial y ejecución de movimientos (Cantillo Maldonado et al., 2018).

Para jugar fútbol con un equipo de robots, se requiere partir del diseño de los mismos, un sistema híbrido centralizado, la implementación de procedimientos multitareas, contar con diferentes tecnologías y estrategias en tiempo real que garanticen este ejercicio. Uno de los procesos iniciales es la adquisición de datos, para ello se requiere el acondicionamiento de sensores para la detección de los movimientos y así poder tomar las decisiones de mando correctas.

La robocopa tiene establecidas las reglas para poder participar en la liga:

1. Sistema de visión
2. Sistema de inteligencia artificial, y
3. Un equipo de robots

En la práctica, el método de posicionamiento basado únicamente en el sistema de inteligencia artificial tiene algunas falencias, tanto en el diseño del control mismo, como en el aprovechamiento de los canales de comunicación disponibles, puesto que se utiliza un sistema centralizado para el control de los cinco robots dentro del campo. Para solucionar este problema, se propone un sistema diferente, en el cual, dentro de cada robot se embarca su propio control de posición, en donde, el sistema de inteligencia artificial se encarga de los cálculos del atractor, y el canal entre la visión artificial y la ejecución de los movimientos de cada uno de los robots, con lo cual cada uno puede conocer su posición y controlarla de forma independiente.

Adicional a esto, para tener una medida más robusta y confiable se propone una fusión multisensorial con otro sistema que dependa de una medida inercial dentro del robot, que permita una validación y corrección de los datos de posición utilizando filtros de Kalman.

Tener dos tipos de medidas es muy usado en los robots para tener mayor confiabilidad en la medida de posición. De aquí se desprende también la necesidad de implementar un protocolo de comunicaciones que se adapte al control propuesto y permita garantizar la integridad y confiabilidad de los datos.

2. ARQUITECTURA GENERAL

El desarrollo de este proyecto debe estar enfocado al cumplimiento de las reglas establecidas para el juego de la liga SSL. La arquitectura establecida está formada por capas independientes que acopladas y en interacción funcionan adecuadamente. La primera capa es un sistema embebido en cada uno de los robots. La segunda capa es un sistema de visión artificial que se encarga de recopilar la información del entorno en este caso la cancha. Y se finaliza con la tercera capa que ejecuta sus funciones desde un computador controlando el grupo de robots, como se muestra en la figura 1.

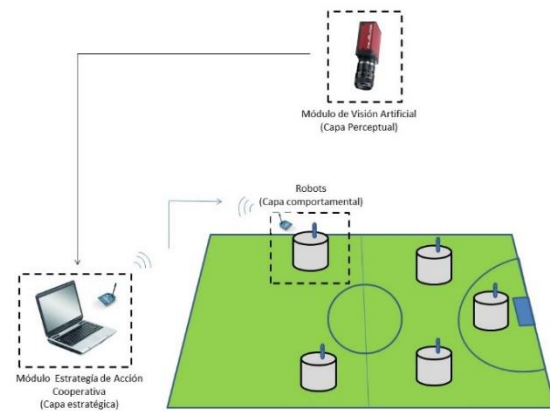


Figura 1. Módulos robocopa y su interacción

En la tabla 1 se muestra la arquitectura del proyecto con las características de cada una de las capas que se compone.

Tabla 1. Capas Arquitectura AGENTCOOP
(González et al., 2011)

Capa	Hardware de ejecución	Entrada	Salida
Percepción Externa	Visión artificial	Imagen entorno general	Entorno de los robots
Estrategia de acción cooperativa	Servidor	Entorno de los robots	Posición robot y atractor
Comportamiento	Robot	Coordenadas de posición	Movimiento

3. ARQUITECTURA LÓGICA

Normalmente la arquitectura de un sistema de control embebido de motor eléctrico DC, consiste en el un módulo de potencia, modulo codificador, generador de PWM y comunicación con el ordenador principal (Zhou, 2008).

Cada uno de los robots cuenta con dos clases de procesadores en su arquitectura. El primero, al que se llamará robot maestro, corresponde al procesador principal, el de mejores características de procesamiento, y el encargado de coordinar las principales funciones del robot. El segundo, que corresponde al módulo de control de los motores, se encarga del procesamiento de todo lo que concierne al control de la variable de velocidad de cada motor del móvil. (Salamanca, J. E., et al, 2020).

A. Módulo Maestro del Robot

Este módulo representado en la figura 2. determina el manejo lógico de cada una de las funciones del robot, como son el control de posición y de motores con comunicación bidireccional, la conversión y adecuación de las velocidades de móvil a velocidades del motor, también se desarrolla el cálculo de la odometría del móvil y la fusión multisensorial de posición mediante el filtro de Kalman no lineal.

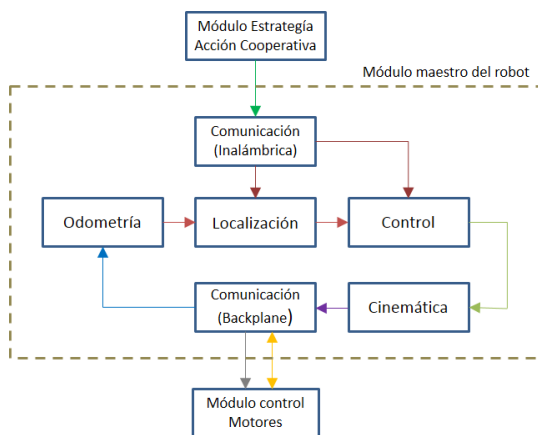


Figura 2. Módulo maestro del motor

B. Módulo de control de motores

El control de velocidad de los motores es la función que debe cumplir este módulo, teniendo en cuenta las indicaciones que envía el sistema maestro del robot, en la figura 3 se muestra la conexión del módulo con las funciones lógicas para las estrategias de mando. En este módulo se hace lectura de la velocidad del motor, por medio de la comunicación bidireccional se realiza el control de velocidad del

motor por medio de la modulación y multiplexación de ancho de pulso de las variaciones de velocidad obtenidas para hacer control de giro de cada uno de los motores. (Valderrama Joaquin Andres, n.d.)

4. CINEMÁTICA DEL ROBOT

La parte operativa actúa directamente sobre la máquina. Son los elementos que hacen que la máquina se mueva y realice la operación para la cual está diseñada. Forman parte de ella los accionadores de las máquinas como son los motores en este caso. (Piedrafita Moreno, 2004)

La figura 4. muestra el esquema geométrico del robot Agentcoop actual, el cual consta de tres ruedas que se acoplan a tres motores posicionados de la forma que se muestran en el diagrama.

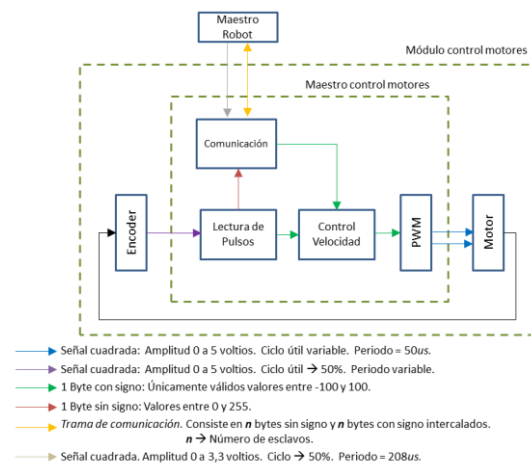


Figura 3. Módulo control de motores

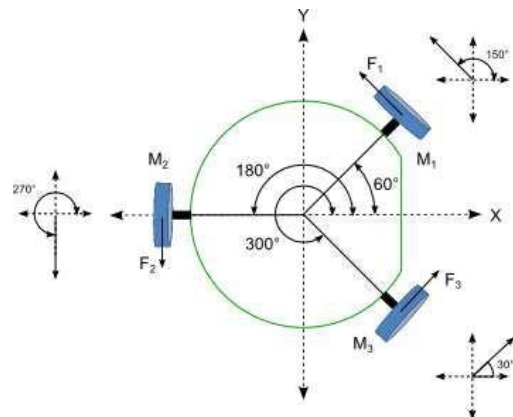


Figura 4. Esquema geométrico robot

Las fuerzas de tracción se obtienen cada vez que los motores son activados, de ahí se generan el torque rotacional y la fuerza traslacional. Cada fuerza de

tracción F_i , se obtiene a partir del torque del motor multiplicado por el radio de la rueda.

En el segundo caso se demostrará que la libertad del movimiento del robot en el que tenemos un control sobre la orientación del robot dinámicamente mientras el robot se está moviendo. Por lo tanto, el robot puede cambiar su orientación de manera controlada mientras se va moviéndose sobre su trayectoria (Jain, 2014). La cual es la forma que se eligió para el movimiento de los robots teniendo en cuenta los requerimientos.

La aceleración traslacional a y la aceleración angular $a\omega$ del centro de masa del robot (que se asume está ubicado en el centro del círculo dibujado por cada robot), se encuentran dadas por las ecuaciones:

$$a = \frac{1}{M} [F_1 + F_2 + F_3] \quad (1)$$

$$\omega = \frac{R}{I} [f_3 + f_2 + f_1] \quad (2)$$

donde:

$M \rightarrow$ Masa del robot

$R \rightarrow$ Radio del robot

$I \rightarrow$ Momento de inercia del robot

$f_i \rightarrow$ Magnitud de la fuerza F_i

Este cálculo es posible utilizando las expresiones dadas, debido a que las fuerzas son tangentes al plano circular del robot, y apuntan hacia la misma dirección rotacional, por lo cual es posible utilizar únicamente la magnitud de los vectores de fuerza. Las magnitudes f_1 , f_2 y f_3 , pueden ser positivas o negativas, dependiendo de la dirección de rotación de los motores, considerándose el sentido horario como una fuerza positiva y el sentido antihorario como una fuerza negativa. (Rojas & F, 2006) Ahora, para un cilindro homogéneo se tiene $I = \frac{1}{2}MR^2$, de modo que, modelando el robot de esta forma, y descomponiendo el vector de la aceleración, se obtiene:

$$\begin{bmatrix} a_x \\ a_y \\ Ra\omega \end{bmatrix} = \frac{1}{M} \begin{bmatrix} \cos(150^\circ) & \cos(270^\circ) & \cos(30^\circ) \\ \sin(150^\circ) & \sin(270^\circ) & \sin(30^\circ) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

(Rojas & F, 2006)

Análogamente al desarrollo mostrado, es posible obtener las ecuaciones para las componentes de velocidad traslacional y rotacional del robot, tomando como referencia de movimiento el centro del robot, que en este caso se muestra como el origen del eje de coordenadas del mismo en la figura 4.

Teniendo en cuenta que la distribución de los motores en el interior del robot forma un triángulo equilátero, se tiene que la velocidad angular total de móvil proviene de la suma del aporte de las velocidades de cada uno de los motores. De esta forma, es posible obtener una relación directa entre las velocidades de móvil y las velocidades de motor, a partir de la ecuación 3

$$\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \begin{bmatrix} -0,8660 & 0 & 0,8660 \\ 0,5 & -1 & 0,5 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (3)$$

Donde:

$v_x \rightarrow$ Componente de la velocidad de móvil en el eje x

$v_y \rightarrow$ Componente de la velocidad de móvil en el eje y

$\omega \rightarrow$ Velocidad angular del móvil

$v_i \rightarrow$ Velocidad del motor i

De donde se obtiene:

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} -0,5773 & 0,3333 & 0,3333 \\ -0,6667 & 0 & 0,3333 \\ 0,5773 & 0,3333 & 0,3333 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \quad (4)$$

5. COMUNICACIÓN (BACKPLANE)

La configuración maestro esclavo es usada en varias aplicaciones, un trabajo en el control de motores, establece inversores que cumplen la función de maestro enviando señales de control a otros inversores conectados al bus que actúan como esclavos (Sanabria Totaitive et al., 2016).

Existen muchas aplicaciones que requieren pequeñas cantidades de datos que se adquirirán desde múltiples ubicaciones distribuidas geográficamente en toda la planta los requisitos de tiempo de respuesta son más flexibles en estos casos (Gour et al., 2017).

Teniendo en cuenta que los procesadores esclavos tienen integrado el control de velocidad los requerimientos en tiempo del sistema se vuelven más flexibles.

En este caso el procesador maestro se asigna para proporcionar un enlace sincrónico en las dos direcciones (enviar, recibir) con los bloques control de cada motor. Su misión es transmitir la información de velocidad de cada rueda de todos los robots, y estar recibiendo la información del conteo de los pulsos obtenidos de los encoders de cada motor, Esto es lo que se llama una estructura Maestro - Esclavo conectado seriamente por medio de un bus de datos, en el cual cada microcontrolador

ubicado en el motor se categoriza como un esclavo, y la unidad de procesador principal del móvil actúa como el maestro de la comunicación, solo él tiene la potestad de iniciar una transmisión de datos, generado los pulsos de reloj que indican la sincronización de la información.

En la figura 5 se visualiza la conexión entre el procesador maestro el esclavo para el envío de la información

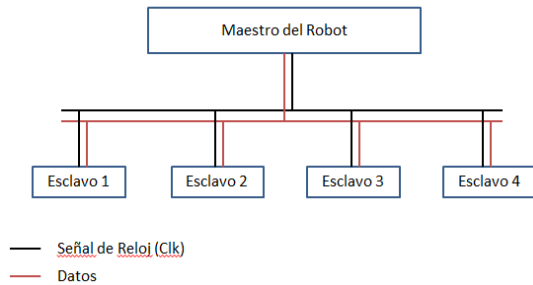


Figura 5. Comunicación maestro del robot - esclavos

Una comunicación de bajo costo que garantiza los retardos mínimos requeridos para el intercambio de datos eficientemente entre vehículos móviles autónomos es la comunicación basada en redes ad-hoc. (Santos et al., 2009)

Se diseñó el programa para el envío de datos (protocolo de encaminamiento ad hoc) para establecer un enlace entre los sistemas de control de velocidad de cada procesador de los motores con el procesador maestro, el objetivo es maximizar la circulación de la información. Se tuvo que diseñar un protocolo de comunicación que se ajustara a las especificaciones de hardware y software de los microcontroladores, ya que hay que tener en cuenta que la capacidad del microcontrolador central y el de cada motor es diferente, además los microcontroladores de los motores no cuentan con algún protocolo comercial, y de acuerdo a las especificaciones tampoco se le puede programar. Este protocolo de comunicación tiene una trama representada en la figura 6, con una secuencia de 8 bits como datos enviados, adicionalmente 3 bits para encontrar algún error, y así tener una totalidad de 11 bits.

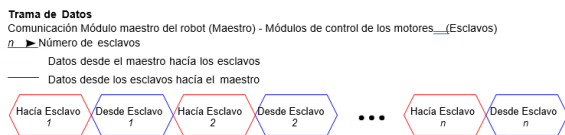


Figura 6. Comunicación del módulo maestro del robot

Si la información enviada viene del microcontrolador maestro del móvil, se interpretan como bytes con su correspondiente signo (números tipo entero con valores de 127 hasta -128), correspondiendo a la referencia de velocidad si se está en un intervalo de 100 a -100, o si no se encuentra en este rango corresponde a una información especial en el controlador de los motores. Por otra parte, si la información viene de uno de los microcontroladores esclavos, se interpreta como un byte que no posee signo (números tipo enteros que van en un intervalo de 255 a 0), referente a un dato que proporciona la medida del conteo de cada pulso en el encoder.

Los tres bits añadidos, son para el rastreo algún posible error, estos datos se localizan al terminar el ultimo bit de información de los datos enviados, sin importar que vengan de los procesadores maestros o esclavos. Esta información equivale a un valor tipo entero que toma números entre 7 y 0, el cual debería indicar el residuo del módulo 8 de las veces que se repite el 1 en la cadena de información enviada. El procesador que recibe la información, hace una validación aplicando el mismo proceso a la información recibida para luego compararla con el número de verificación enviado, si estos dos valores no coinciden, la información se considera errónea y no es tomada en cuenta. En la figura 7 se muestra cada bit de un ejemplo gráfico en la detección de un error.

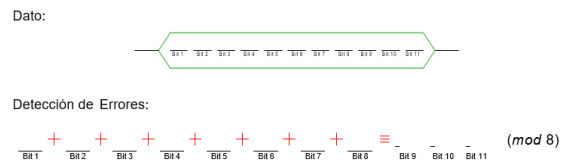


Figura 7. Detección de errores

Siempre que el procesador maestro transmite una trama de datos hacia los procesadores esclavos, se envía la referencia de velocidad al primer dispositivo de motor. Al terminar el envío de los 11 bits hacia el esclavo, este responde enseguida con los datos que tiene acumulados en los encoder. Después de terminar esta comunicación este proceso se repite con el segundo dispositivo de motor y así sucesivamente hasta que complete la totalidad de dispositivos esclavos de motor que tenga el robot, para este caso son 3 pero se puede modificar. Si se hacen cuentas de los bits de la información transmitidos se obtiene que por cada dato recibido por el esclavo también se envía uno. Si llamamos n al número de motores esclavos, el número total de bits enviados sería de 2n por cada bit de información hacia los esclavos. Adicionalmente se debe sumar

en la comunicación hacia los microcontroladores esclavos unos bits de configuración que envía el procesador maestro: 1 bit que indica el inicio de la comunicación. 1 bit que informa el número del motor esclavo con el que se va a comunicar, y 1 bit que indica la terminación de la comunicación. Esto genera un total de $1+1+1+1+1=5$ bits por cada microcontrolador esclavo con el que se comunique. Lo cual genera $25n$ bits para n microcontroladores esclavos.

De los 25 bits totales enviados por un esclavo, 14 pertenecen a la información enviada al microcontrolador esclavo, por lo que el porcentaje de datos enviados por cada comunicación esclavo maestro es de:

$$\frac{14}{25} \times 100\% = 56\% \quad (5)$$

de los datos transmitidos.

Teniendo en cuenta que el robot tiene 3 ruedas, se utilizan 3 motores esclavos. Además, los pulsos de reloj tienen una duración de $320 \mu s$. Adicionalmente se suma la información que se debe enviar al sistema de pateo del balón, y para finalizar una trama más de 25 bits, obteniendo un total de 100 bits por cada comunicación con los esclavos.

Por lo que se tiene un tiempo total de:

$$75 \times 320 \mu s = 24 ms \quad (6)$$

Cada vez que se hace una comunicación maestro-esclavo. Si únicamente se tiene en cuenta la comunicación con los motores esclavos se tiene una velocidad de:

$$\frac{75 \text{ bits}}{24 \text{ ms}} = 3,125 \text{ Kbits/seg} \quad (7)$$

6. ODOMETRÍA

Generalmente se usan dos formas para la implementación de la odometría visual, la más usada se basa en seguir características distintivas del objeto como esquinas, bordes, etc. (González et al., 2015) En este caso se usan marcadores para cada robot, indicando el quipo y número de jugador.

El módulo maestro del robot calcula la velocidad por medio del control y ésta se establece como el mando que es el conjunto de datos que comunica cada $t_{\Delta} = 800 \text{ ms}$ el módulo maestro del robot con los módulos de control de los motores, esta comunicación es el canal para que se den las modificaciones de la velocidad de los robots

móviles y las lecturas de los encoders para poder establecer el control de la variable de posición.

Ahora bien, para determinar la distancia que recorre el robot móvil en una fracción de tiempo t_{Δ} calculamos $x = v \times t_{\Delta}$ considerando constante la velocidad de cada motor y a así obtenemos:

$$\begin{bmatrix} x \\ y \\ R\theta \end{bmatrix} = \begin{bmatrix} -0,8660 & 0 & 0,8660 \\ 0,5 & -1 & 0,5 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (8)$$

Donde:

$x \rightarrow$ Distancia recorrida por el móvil en el eje x

$y \rightarrow$ Distancia recorrida por el móvil en el eje y

$R \rightarrow$ Radio del robot

$\theta \rightarrow$ Ángulo total de giro del robot

$x_i \rightarrow$ Distancia recorrida por la rueda del motor i

La coordenada de orientación $R\theta$ indica la medida de la trayectoria de giro total realizada por el jugador o la medida de la trayectoria del recorrido por el jugador en el eje ω . Esta medida de distancia está determinada por el ángulo de giro total del robot móvil multiplicado por la medida del radio de este. La distancia recorrida por cada una de las ruedas del jugador es de $2r\pi$ por un giro completo, representando r el radio de la rueda que tiene una medida de 2.5 cm.

El número de giros dado por cada rueda del jugador en el tiempo t_{Δ} está representado por el valor del conteo que se obtiene del bloque de lectura de los pulsos de cada uno de los módulos, cada giro total de la rueda tiene 100 pulsos, así el número de giros esta dado por el número de pulsos dividido entre 1000.

Así x me representará la distancia de cada una de las ruedas en un tiempo t_{Δ} , Pulsos es el número enviado en el mismo tiempo, obtenemos que:

$$x = 2\pi r \frac{\text{Pulsos}}{1000} = 5\pi \frac{\text{Pulsos}}{1000}$$

Considerando el radio del robot (8cm) y la escala de 16 para cada conteo, a partir de la ecuación 3, es posible obtener finalmente las ecuaciones:

$$\begin{bmatrix} x \\ y \\ R\theta \end{bmatrix} = \frac{5\pi}{1000} \times 16 \times \begin{bmatrix} -0,8660 & 0 & 0,8660 \\ 0,5 & -1 & 0,5 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} \quad (9)$$

$x \rightarrow 0,21765 \times (L_3 - L_1)$

$y \rightarrow 0,12566 \times (L_1 - 2L_2 + L_3)$

$\theta \rightarrow 0,03141 \times (L_1 + L_2 + L_3)$

Donde:

$L_i \rightarrow$ conteo de pulsos enviado desde el módulo de control i , representa el total de pulsos obtenidos del

encoder, dividido en 16

7. CONTROL DE VELOCIDAD DE LOS MOTORES

Si se puede obtener un modelo matemático de la planta, es posible aplicar diversas técnicas de diseño con el fin de determinar los parámetros del controlador que cumpla las especificaciones del transitorio y del estado estacionario en lazo cerrado. (Ogata, 2010)

En este caso la función de transferencia está dada por el voltaje como entrada y la velocidad del motor como la salida, aproximándolo a un sistema de primer orden. Para aplicar las diferentes técnicas de control de velocidad en motores se hace necesario de disponer un modelo matemático que facilite el análisis del motor y la manipulación de las variables propias para el control (Martin et al., 2016).

Se puede establecer una acción de control Proporcional Integral para la velocidad teniendo en cuenta las características de procesador, finalmente en lazo cerrado se tiene un sistema de segundo orden con parámetros ajustables como la sobreelongación y el tiempo de asentamiento. (figura 8)

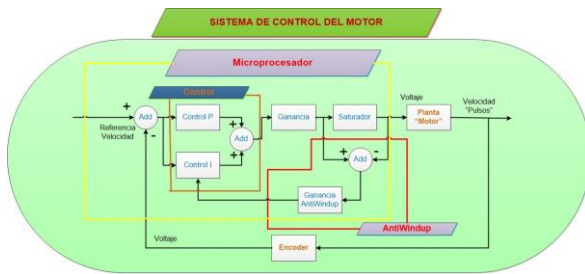


Figura 8. Diagrama de bloques control con Antiwindup

En el modelado de los motores el voltaje aplicado es en proporción a la velocidad de la rotación. Cuando la batería se gasta, la tensión baja, disminuyendo la velocidad del mayor valor de entrada (100%). Teniendo en cuenta que el controlador que se diseñó tiene la integral del error, en el momento que la batería se descargue y no logre llegar al 100% suponiendo que el control desea que el motor este al máximo, la integral del error comienza a aumentar sin llegar a un tope, produciendo que el conteo del procesador se reinicie (llegue a la mayor magnitud aceptada por la memoria y reinicie el conteo nuevamente) (Sierra, O. J. S., et al, 2016).

A. Modelado de los motores

Para modelar la función de transferencia del sistema de primer orden, se realizaron varios experimentos

con la variable del voltaje máximo de la batería (6 volt.), aplicando a los motores entradas escalón multiplicadas en diferentes factores, 0,2, 0,4, 0,6, 0,8 y 1 que representa la energía máxima de la batería, ver figura 9, así se promedia el valor de la ganancia y de la mínima constante de tiempo y se obtiene la función de transferencia

$$G(s) = \frac{148.85 \times 10^{-3}}{s + 139.1130 \times 10^{-3}} \quad (10)$$

Haciendo el montaje experimentalmente encontraron unas salidas muy pequeñas en las cuales los motores no se mueven, aunque se le esté poniendo este voltaje las cifras encontradas rodean el 7% de el voltaje máximo. Eso se debe a que las llantas deben superar la fuerza de fricción.

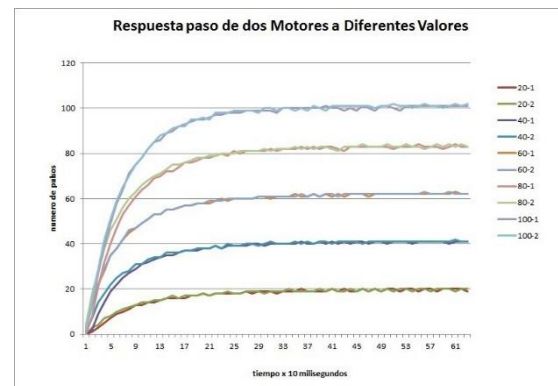


Figura 9. Respuesta de un motor a diferentes entradas

B. Control de velocidad de motores

Para establecer el control de la velocidad del robot homonímico se utiliza un controlador PID, teniendo en cuenta que la implementación del algoritmo PID en un robot omnidireccional de tres ruedas redujo el tiempo de asentamiento y el período de ciclo del encabezado respuesta a un punto fijo (Shaw et al., n.d.).

Se establece una técnica de control proporcional integral para la velocidad del motor, al desarrollar el sistema retroalimentado se tiene:

$$G_{\text{retro}}(s) = \frac{s \left(\frac{K_p K}{\tau} \right) + \frac{K_i K}{\tau}}{s^2 + \left(\frac{K_p K + 1}{\tau} \right) s + \frac{K_i K}{\tau}} \quad (11)$$

De donde obtenemos:

$$K_i = \frac{\omega_n^2 \tau}{K} \quad (12)$$

$$K_p = \frac{2\zeta\omega_n\tau - 1}{K} \quad (13)$$

Con el valor de la mínima constante de tiempo y el tiempo de levantamiento en el cual más se demoró en responder el motor a una entrada impulso unitario, se tomaron como parámetros un tiempo de asentamiento de 0.2 segundos y que la respuesta responda a un sistema críticamente amortiguado con un factor de amortiguamiento $\zeta = 1$. Con estos valores se tiene un $K_i = 26,9$ y un $K_p = 1.75$.

Para que el maestro acepte los datos que deben ser números enteros se debe adaptar la información multiplicando por 2^7 esto permite que la acción integradora realice su trabajo cuando se procesen los datos y se logre una adecuada respuesta, posteriormente el valor se divide en 128 que corresponde a 7 corrimientos binarios

$$K_{p2}(K.T_s) = 224.E(K.T_s) \quad (14)$$

$$K_{i2}(K.T_s) = 34,43.E(K.T_s) + K_{i2}((K-1)T_s) \quad (15)$$

Determinación de las constantes teniendo en cuenta las características del procesador en variables como la velocidad y error en tiempos de milisegundos

$$K_{p3}(K.T_s) = 256.E(K.T_s) \quad (16)$$

$$K_{i3}(K.T_s) = 32.E(K.T_s) + K_{i2}((k-1)T_s) \quad (17)$$

En la figura 10 esta representado el sistema de control de velocidad en cada motor, con las ganancias del controlador establecidas un periodo de muestreo de 10 ms, y el bloque adicional de anti-windup para limitar la parte integral.

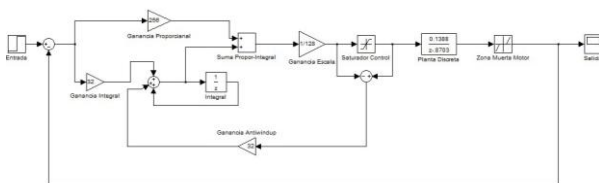


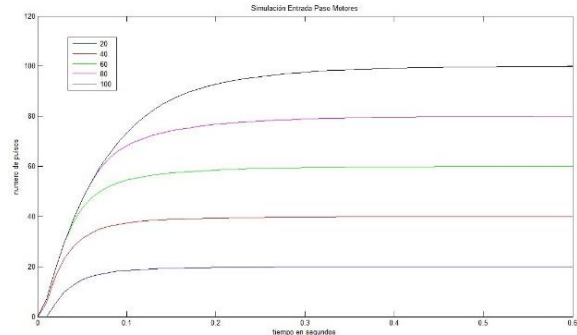
Figura 10. Esquema del sistema de control de velocidad en cada motor

Las respuestas del sistema simulado del motor con el control en tiempo discreto ante varias entradas del escalón en rango de porcentajes, se representa en la figura 11.

Figura 11. Grafica de distintos valores de entradas paso ingresadas a un motor con controlador

A. Anti-windup Control de velocidad de Motores

El anti-windup es la estrategia que se determinó para



que se saturara la acción integral del controlador y poder tener un límite.

$$\text{Pulso_Anterior} = \text{Pulsos_Sat} + \frac{K_i}{K_{\text{Anti Windup}}} \text{Error} \quad (18)$$

Teniendo en cuenta que la constante integradora tiene un valor de 32, se elige este mismo para el anti-windup, figura 10

Como la constante integral es 32 para optimizar cálculos hacemos la constante del anti-windup igual a 32 ver figura 10 con esto se tiene

$$\text{Pulso_Anterior} = \text{Error} + 100 \quad (19)$$

El mayor error permitido es de ± 200 ya que la salida se mide de 100 a -100, un error de 200 quiere decir que la batería está entregando el voltaje al 100%, si este valor coincide con el último valor medido significa que la batería no se ha descargado y el anti-windup no entra en funcionamiento para esta ocasión.

La situación más extrema es mientras la batería está lo más descargada posible y la salida de voltaje es de 100 o -100 el error más extremo es de 200. Teniendo en cuenta los cálculos anteriores la mayor cifra en la entrada del saturador y antes de hacer la división por 128 es ± 200 lo que es fácilmente procesado por el microcontrolador.

8. SISTEMA CONTROLADOR DE POSICIÓN

El sistema controlador de posición hace que el móvil se sitúe en la posición deseada cambiando la información de velocidad en todas las ruedas del jugador (las ruedas se mueven independientemente mediante motores eléctricos). Se desea que el sistema este embebido en cada uno de los robots móviles y que el algoritmo se encuentre en el microcontrolador maestro de cada uno de los robots.

Con el objeto de llevar a cabo el controlador de posición de los móviles holonómicos se estudiaron algunos artículos sobresalientes en el tema como: (Baede, 2006) y (Rojas & F, 2006). En todos los artículos se usan control tipo PID obteniendo buenos desempeños. Teniendo en cuenta lo anterior se optó por utilizar un control tipo PID en los móviles en cada coordenada ortogonal ejes X y Y el ángulo de orientación θ . Con el fin de encontrar las constantes del control PID se hace una simulación de todo el bloque de los móviles en el software Matlab y generando una rejilla de valores se hace la minimización de la acumulación del error (parte integral) poniendo de ingreso una función tipo paso.

Antes de implementar el software que controle la posición se debe tomar en consideración que el móvil, a pesar de que se modela como una zona puntal definida por coordenadas X, Y y θ , en realidad el robot tiene una distribución que se puede ajustar de forma rectangular. En ciertos momentos el punto de posición deseada tiene la posibilidad de convertirse en la ubicación de la pelota, sin embargo, el móvil no debería situarse en la misma posición de la pelota. Basándose en el análisis anterior, el tamaño del móvil, y que los datos son digitales se encuentra el requisito de tener una región donde el robot se quede inmóvil para todos los ejes de translación (que el robot no presente movimiento en el eje X y Y).

Siempre y cuando las dimensiones sean de un tamaño considerable se satura el controlador generando que los niveles aumenten aleatoriamente sin límites lo que modifica la salida del controlador mientras se encuentra en la sección saturada. A fin de solucionar este inconveniente se utiliza un bloque de activación que únicamente utiliza el controlador cuando la posición se encuentra en un lugar permitido ver figura 12 (saturación o posición inactiva). Si el robot se encuentra en un lugar que permite aplicar la acción de control PID y la medida de la posición al punto deseado es superada a la permitida se pone en saturación el voltaje de los motores (mayor velocidad en todas las coordenadas X, Y y θ), en cambio, si la medida del robot al punto deseado es inferior a la permitida se pone nula la velocidad en todas las coordenadas X, Y y θ . Posteriormente a calcular la velocidad de cada coordenada X, Y y θ se transforma el voltaje y se separa la magnitud de cada motor para luego sumarla según el motor correspondiente (realizar una superposición de voltaje para cada motor del móvil).

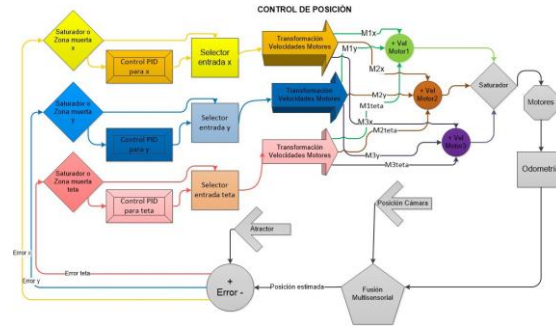


Figura 12. Diagrama de control

A fin de asegurar que cada motor tenga el intervalo de voltaje y el móvil se traslade al punto deseado, el voltaje de cada motor pasa por una saturación que estandariza en proporción a cada valor de el voltaje, cuando la salida más elevada supere a la carga de la batería que tenga en ese momento (sea mayor del 100% de el voltaje de las baterías y se aplique a los motores).

Si la salida del mayor valor de los motores supera el tope máximo permitido que establece la carga de la batería, la salida se intercambia al máximo permitido que se calcula de acuerdo a la carga de las baterías. El resto de las salidas bajan en proporción a lo que bajo la salida más alta (se incrementa en una porción de valor equivalente a el voltaje más alto permitida por las baterías dividiéndolo la salida de voltaje más alta). Después la salida de voltaje que pasa por el saturador es aplicada a cada motor para que por medio de la medición en la imagen trate de localizar la ubicación del móvil. Finalmente, con la posición deseada a la que debe llegar el robot y la posición actual se estima un error para luego repetir todos los pasos anteriores. Este procedimiento se puede encontrar en la figura 12. Con el controlador se desea optimizar la integral de error de la posición, lo que se convierte en una mejor velocidad del móvil hacia el punto deseado.

9. CONCLUSIONES

De forma global lo que se obtuvo experimentalmente en contraste a lo que se esperaba utilizando simulaciones y modelos matemáticos fue muy semejante, además, se resalta un mejoramiento en comparación a proyectos similares que se realizaron anteriormente. También, el controlador de posicionamiento se desempeñó de acuerdo a los criterios calculados y respondió con las proyecciones de rendimiento ya que el sistema embebido está incluido en los robots del equipo.

En el dispositivo de microcontroladores y motores secundarios el movimiento de cada uno de los motores se modelo teniendo en cuenta que se

comporta como un sistema de primer orden, teniendo un desempeño con buena precisión en este uso, aunque, el microcontrolador esclavo comparado con otros no es muy robusto por el tamaño reducido, se pudo maximizar y estabilizar la velocidad logrando las exigencias temporales y lograr un mejor funcionamiento cuando participe en la competencia de la robocopa.

Al encontrar el modelo odométrico para hacer el control de velocidad, se tuvo que cambiar o hacer una corrección de los valores encontrados teóricamente por medio de cálculos matemáticos, porque se tienen unos valores que derivan en unos cálculos que carecen de exactitud cuando se toman datos del sistema real, por ejemplo la masa que tiene el robot, el nivel de carga que tienen las baterías, la pérdida de fricción en las llantas o patinaje de la rueda del robot y la inexactitud de algunas medidas en los parámetros físicos del robot, Aplicando simulaciones con estos nuevos parámetros se encontró un mejor acercamiento en el modelado de los robots.

Por otra parte, se corroboro que si se utiliza solo un sistema que brinde la localización del robot no es suficiente para el buen funcionamiento en el controlador de la posición de los robots. Esto se debe a que la medida en la rueda físicamente genera un error que es acumulativo que hace que en un lapso de tiempo relativamente grande no se tenga correctamente la posición del robot.

La odometría por medio de las imágenes tiene un poco de retardo para llegar al procesador maestro del robot y queda sometida a variaciones lumínicas y del medio ambiente según donde se encuentre el móvil, lo que puede producir algún error y/o toma de datos incorrectos de la posición del móvil, esto causa un resultado que no es deseado. Sin embargo, si se aplica el filtro de Kalman garantiza minimizar de forma significativa la repercusión de los dos focos de incertidumbre.

Los resultados del protocolo de comunicación que se implementaron junto con los equipos usados para controlar cada motor se acoplaron favorablemente para esta aplicación específica. Para proyectos más generales se recomienda utilizar microcontroladores que contengan programas pertinentes que soporten cualquier algoritmo de comunicación estandarizado, porque vuelve el sistema más robusto y permite aligerar la programación de los microcontroladores, adicionalmente ayuda a minimizar los tiempos de recepción y transmisión de la información.

El dispositivo de identificación de fallos en la transmisión de información mostró que es idóneo y fiable para encontrar y erradicar el 99% de los

envíos erróneos. Más aun, la información errónea, en la recepción y transmisión de los datos no alcanzaron a superar el 3% del total de información recibida y transmitida en la comunicación. No obstante, como es un algoritmo establecido totalmente por programación, además que el sistema reparte el proceso de datos con el resto de trabajos que deben hacer los móviles, la velocidad del sistema que se obtuvo fue menos que la estándar en una conexión (9600 bps) UART.

REFERENCIAS

- Baede, T. A. (2006). *Motion control of an omnidirectional mobile robot*.
- Cantillo Maldonado, A., Gualdron Guerrero, O., & Ortiz Sandoval, J. (2018). Procesamiento De Señales Emg En Un Sistema Embebido Para El Control Neuronal De Un Brazo Robótico. *Revista Colombiana De Tecnologías De Avanzada (RCTA)*, 2(32).
<https://doi.org/10.24054/16927257.v32.n32.2018.3037>
- Gonzalez, E., Rosa, F. de, Miranda, A. S., Rodríguez, C. F., & Manrique, M. (2011). *Bochica 2011 - Team Description Paper*. 1–9.
- González, R., Rodríguez, F., & Guzmán, J. L. (2015). Robots móviles con orugas. Historia, modelado, localización y control. *RIAI - Revista Iberoamericana de Automatica e Informatica Industrial*, 12(1), 3–12.
<https://doi.org/10.1016/j.riai.2014.11.001>
- Gour, A., Sujith, K. R., Sakthivel, M., Behera, R. P., Jayanthi, T., Madhusoodanan, K., & Murty, S. A. V. S. (2017, February 13). Novel approach of backplane communication for compact & field mountable I&C systems of FBR. *1st IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems, ICPEICES 2016*.
<https://doi.org/10.1109/ICPEICES.2016.7853340>
- Jain, P. (2014). Odometry and motion planning for omni drive robots. *Proceedings of the International Conference on Innovative Applications of Computational Intelligence on Power, Energy and Controls with Their Impact on Humanity, CIPECH 2014*, 164–168.
<https://doi.org/10.1109/CIPECH.2014.7019080>
- J Astrom, K., & Wittenmark, B. (1997). *Computer-control system* (3rd ed.). Prentice Hall.

- Martin, I., Nieves, G., Jaime, A., Gonzalez, J. L., Diaz, J. A., & Velazco, O. (2016). Estimation Of The Induction Engine Speed Using Neural Networks Estimación De La Velocidad Del Motor De Inducción Utilizando Redes Neuronales. *Ta Colombiana De Tecnologías De Avanzada (RCTA)*.
- Ogata, K. (2010). *Ingeniería de control moderna* (5th ed.). Prentice Hall.
- Piedrafita Moreno, R. (2004). *Ingeniería de la automatización industrial* (2nd ed.). Alfaomega.
- Rojas, R., & F, A. G. (2006). *Holonomic Control of a robot with an omni- directional drive*.
- Sanabria Totaitive, C. A., Mauricio Hernández Gómez, O., & Abdel Hay, M. K. (2016). Control Strategy For A Single Phase Inverter Based On Droop Method And Ac Bus Connection. *Ta Colombiana De Tecnologías De Avanzada (RCTA)*.
- Salamanca, J. E., Diaz Rodriguez, J. L., & Pardo Garcia, A. (2020). Speed estimation of the three-phase induction motor with Volts/Hertz control using artificial intelligence techniques. Paper presented at the Journal of Physics: Conference Series, , 1704(1) doi:10.1088/1742-6596/1704/1/012017
- Santos, R. A., González Potes, A., Villaseñor Gonzalez, L. A., Crespo, A., Sanchez, J., & Gallardo, J. R. (2009). *Simulación de Algoritmos para regular el Flujo Vehicular y la Comunicación entre Vehículos Móviles Autónomos utilizando Redes Ad Hoc*. <http://www.revista-riai.org>
- Shaw, R. N., Walde, P., Galgotias University, Institute of Electrical and Electronics Engineers, IEEE Industry Applications Society, & International Conference on Computing, P. and C. T. (GUCON) G. U. G. N. U. India. S. 27-28, 2019. (n.d.). *Path Planning and Controlling of Omni-Directional Robot Using Cartesian Odometry and PID Algorithm*.
- Sierra, O. J. S., Garcia, A. P., & Sanchez, E. N. (2016). Intelligent control for the exciter of a synchronous machine with closed loop voltage. Paper presented at the World Automation Congress Proceedings, , 2016-October doi:10.1109/WAC.2016.7582983
- The Robocup Federation [Online]. (n.d.). Robocup. <Http://Www.Robocup.Org/>. <http://www.robocup.org/>
- Valderrama Joaquin Andres, P. A. O. (n.d.). *Diseño de Leyes de Control Basadas en la Teoría de la Estabilidad de Atractores, Aplicadas en Robots Agentcoop Jugadores de Fútbol*. Universidad Javeriana.
- Zhou, H. (2008). DC servo motor PID control in mobile robots with embedded DSP. *Proceedings - International Conference on Intelligent Computation Technology and Automation, ICICTA 2008, 1*, 332–336. <https://doi.org/10.1109/ICICTA.2008.426>