Revista Nexos Científicos
Julio –Diciembre 2019 pp. 28-42
Volumen 3, Número 2
Fecha de recepción: octubre 2019

ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2019

Diseño de robots de Competencia en Categorías Seguidor de Línea y Robot Laberinto

Luis Toca[1]; Diego Romo[2]; Carlos Ruiz[3]; Bayron Machay[4]

[1]Carrera de Tecnología Superior en Electromecánica, Instituto Superior Tecnológico Vida Nueva Quito, Ecuador, luis.toca@istvidanueva.edu.ec
[2]Carrera de Tecnología Superior en Electricidad, Instituto Superior Tecnológico Vicente Fierro Tulcán, Ecuador, diegoromo_408@hotmail.com
[3]Carrera de Tecnología Superior en Electromecánica, Instituto Superior Tecnológico Vida Nueva Quito, Ecuador, carlos.ruiz@istvidanueva.edu.ec
[4]Carrera de Tecnología Superior en Electromecánica, Instituto Superior Tecnológico Vida Nueva Quito, Ecuador, bayron.machay@istvidanueva.edu.ec

Resumen: En el presente artículo se desarrolla el diseño e implementación de un robot seguidor de línea velocidad y un robot laberinto con las especificaciones técnicas del Concurso Ecuatoriano de Robótica permitiendo la implementación de los mismos en parte mecánica, eléctrica y programación. Estos robots son diseñados especialmente para competencias de robótica nacionales e internacionales. En el diseño de los prototipos se toma en cuenta el ambiente en el que se van a desenvolver, y aspectos tales como: forma del chasis, peso del prototipo, distribución de elementos en el chasis, motores, sensores, algoritmos de control a utilizarse, entre otros. Los robots son completamente autónomos, el robot seguidor de línea velocidad utiliza un controlador borroso o un controlador proporcional integral derivativo (PID) para el seguimiento de la línea. El robot laberinto está provisto del algoritmo de la mano derecha, el algoritmo de la mano izquierda y el algoritmo de relleno de callejones sin salida para la resolución de laberintos.

Palabras clave: Seguidor, laberinto, controlador, borroso, competencia, algoritmo.

Design of Competition robots in Line Follower and Maze Robot Categories

Abstract: This article develops the design and implementation of a speed line tracker robot and a maze robot with the technical specifications of the Ecuadorian Robotics Competition allowing the implementation of the same in part mechanical, electrical and programming. These robots are specially designed for national and international robotics competitions. The design of the prototypes takes into account the environment in which they are to be developed, and aspects such as: shape of the chassis, weight of the prototype, distribution of elements in the chassis, motors, sensors, control algorithms to be used, among others. Robots are completely autonomous, the speed line tracker robot uses a fuzzy controller or a derivative integral proportional controller

Keywords: Follower, maze, controller, fuzzy, competition, algorithm.

I. INTRODUCCIÓN

En los últimos años, la robótica de competencia ha tenido una mayor acogida en el mundo entero, es así como en Ecuador, se tiene varios torneos de robótica que se realizan año a año, uno de los más importantes es el Concurso Ecuatoriano de Robótica (CER) en el cual compiten Universidades, Escuelas Politécnicas, e Institutos Tecnológicos del país (RobotChallenge, 2016).

Otro torneo que está tomando acogida en los últimos años es el Robot Games Zero Latitud, en este torneo participan delegaciones de varios países y se usan reglamentos similares al Concurso Ecuatoriano de Robótica con algunas modificaciones. A nivel internacional uno de los torneos más importantes es el RobotChallenge que se lo realiza en Viena, Austria desde el 2004 y en el cual participan delegaciones de muchas partes del mundo para demostrar su talento en la robótica (RobotChallenge, 2016).

1. luis.toca@istvidanueva.edu.ec
2. diegoromo_408@hotmail.com
3. carlos.ruiz@istvidanueva.edu.ec
4. bayron.machay@istvidanueva.edu.ec

ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2019

Por esta razón se ha realizado una investigación del desarrollo de este tipo de robots, enfocándose en las categorías seguidor de línea velocidad y robot laberinto.

El objetivo del robot seguidor de línea es recorrer un circuito en el menor tiempo posible. En cambio, el robot laberinto debe ser capaz de encontrar la salida de un laberinto en un tiempo máximo preestablecido (J. S. Tercero, 2009).

II. ROBÓTICA DE COMPETENCIA

Una competición de robótica es un evento donde participan prototipos de robots, los cuales pueden ser construidos o adquiridos en casas comerciales y programados por los participantes para realizar una tarea específica cumpliendo con un reglamento preestablecido por la organización, con el fin de superar al resto de participantes. Las características mecánicas como electrónicas de los prototipos están definidas por los organizadores de los torneos.

A. Seguidores de línea velocidad

La tarea del robot es seguir una pista marcada con una línea negra en un fondo blanco (el robot debe ser autónomo). La complejidad de la pista depende del reglamento de los organizadores del torneo. En la Fig. 1 se puede observar la pista de la categoría robot seguidor de línea velocidad del Concurso Ecuatoriano de Robótica 2015.

Fig. 1 Pista del robot seguidor de línea velocidad del Concurso Ecuatoriano de Robótica 2015.

B. Robot laberinto

En esta categoría, el prototipo es autónomo y debe enfrentarse a la tarea de salir de un laberinto; al robot se lo coloca en la entrada de éste y el prototipo debe ser capaz de encontrar su salida en un tiempo máximo preestablecido Concurso (Ecuatoriano de Robótica 2015).

III. DISEÑO MECÁNICO DE LOS PROTOTIPOS

El diseño de los robots de este proyecto se lo realizó bajo el reglamento del Concurso Ecuatoriano de Robótica. Teniendo en cuenta que las pistas donde participan los prototipos son planas, se escoge la locomoción por ruedas, por las ventajas que ésta presenta. Pero existen varios tipos de configuraciones en este tipo de locomoción, como: tracción diferencial, triciclo, ackerman, entre otras.

Con respecto al robot laberinto, los principales movimientos que este realiza son: en línea recta, curvas un U a la derecha o izquierda, curvas de 90º y giros de 180º sobre su propio eje cuando este entra en un camino sin salida.

El robot seguidor de línea, se desplaza en línea recta y realiza giros en curvas de diferentes radios, el radio más pequeño que se puede encontrar es de 5 cm según el reglamento, todo esto a la mayor velocidad posible.

Teniendo en cuenta el tipo de movimientos que deben realizar los prototipos se escoge la configuración tipo tracción diferencial, debido a que permite realizar giros en curvas de radio pequeño que es lo que se necesita en ambos robots y es la única que permite realizar un giro sobre su propio eje, el cual se necesita en el robot laberinto.

A. Robot seguidor de línea

El diseño mecánico se inicia por el chasis, el cual da soporte mecánico al robot y es una parte muy importante ya que éste influye significativamente en el desempeño del prototipo por lo que éste debe ser resistente para que de soporte y liviano para que la inercia del prototipo sea baja y pueda acelerar, frenar y girar con facilidad.

Otro aspecto importante en su diseño es la distribución de los elementos a los cuales brinda soporte, ya que una mala distribución hará que el robot pierda tracción y derrape al tomar una curva

Revista Nexos Científicos
Julio –Diciembre 2019 pp. 28-42
Volumen 3, Número 2
Fecha de recepción: octubre 2019
ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2019

con rapidez, por esto se debe tomar en cuenta que el centro de gravedad de éste quede cerca al centro del eje que une las ruedas.

Las dimensiones a considerar en el diseño del robot son la distancia entre las ruedas y la distancia entre eje de las ruedas y los sensores.

Se puede observar en (1) del modelo cinemático de un robot de tracción diferencial que la distancia entre ruedas no debe ser muy grande ya que el robot tarda más al girar y se necesita que las variaciones de las velocidades de las ruedas sean mayores.

$$w = \frac{V_R - V_L}{L} \quad (1)$$

ω = Velocidad angular del robot.
V_L = Velocidad de la rueda derecha.
V_L= Velocidad de la rueda izquierda.
L = Distancia entre las ruedas.

Otro punto importante en el diseño es la distancia entre los sensores de línea y el eje de las ruedas, generalmente se une la placa de control con los sensores mediante un brazo, la distancia que debe tener el brazo depende de las características de la pista, si es largo le permitirá al robot anticiparse a las curvas pero si las curvas son muy cerradas el robot no va a poder tomar la curva correctamente; por lo que si las curvas son cerradas el brazo deberá ser más corto.

También los motores son una parte fundamental en el robot, ya que de estos depende la velocidad con la que puede desplazarse, aquí se debe considerar el peso del robot y el radio de la ruedas para su correcta selección. Para calcular el torque del motor se necesita la masa total del robot, el radio de la rueda y la aceleración que se desee, para lo cual se aplica **¡Error! No se encuentra el origen de la referencia.**

$$T = M * (a + g * \sin(\theta)) * r \quad (2)$$

Donde:
T: Torque del motor.
M: Masa total del robot.
a: Aceleración.
θ: Ángulo del plano.
g: Gravedad.
r: Radio de las ruedas.

En este prototipo se utilizó micromotores HP de rotor extendido con caja reductora de 10:1, 3000 RPM y 4 oz-in de torque. En la Fig. 2 se observa el motor y su acople a la rueda.

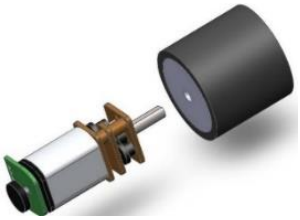


Fig. 2 Micromotor HP de rotor extendido de 10:1

B. Robot laberinto

El diseño mecánico de este robot se inicia por el chasis el cual da soporte mecánico al robot, se necesita que sea liviano, resistente y pequeño para que no se tenga problemas con los giros dentro del laberinto. Para su diseño se tiene en cuenta la distribución de los elementos a los cuales brinda soporte y que el centro de gravedad del robot quede cerca al eje que une las ruedas.

Hay varias formas que puede tomar el chasis y cumplir con lo que se dijo, para este robot se tomó la decisión de hacerlo redondo, ya que puede girar más libremente y la probabilidad de quedarse atascado con una pared en caso de choque es menor. En la Fig. 3 se puede apreciar lo mencionado; se compara un chasis redondo con uno de forma cuadrada.

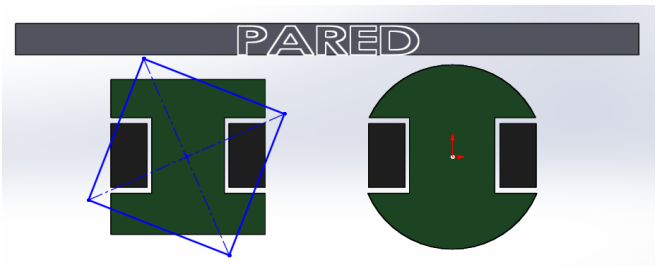


Fig. 3 Influencia del chasis sobre el comportamiento del robot laberinto

Para que el robot sea equilibrado y cumpla con lo ya dicho, el eje que une las ruedas pasa por el centro del chasis, de otra forma, el robot al girar en

Revista Nexos Científicos
Julio –Diciembre 2019 pp. 28-42
Volumen 3, Número 2
Fecha de recepción: octubre 2019
ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2019

su propio eje describirá una circunferencia de mayor tamaño aumentando las probabilidades de choque. En la Fig. 4 se puede observar un ejemplo de lo que ocurriera si las ruedas se localizaran en la parte trasera del robot.

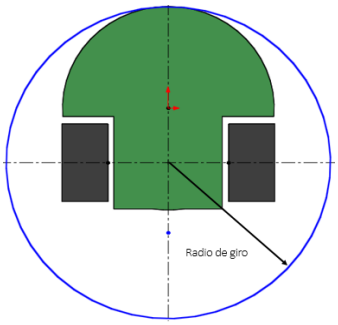
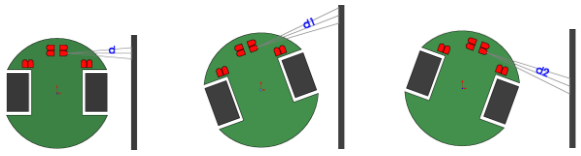


Fig. 4 Robot laberinto con eje de las ruedas en la parte trasera

Otro punto importante de diseño es la disposición de los sensores que detectan las paredes, se utilizó cuatro sensores: dos para la detección de la pared frontal y dos para la detección de las paredes laterales.

Talvez la posición más intuitiva para los sensores laterales es colocarlos perpendicularmente a las paredes laterales como se presenta en la Fig. 5 (a); pero no es la mejor opción, porque se dificulta la implementación del control de posición que sirve para evitar chocar con las paredes.

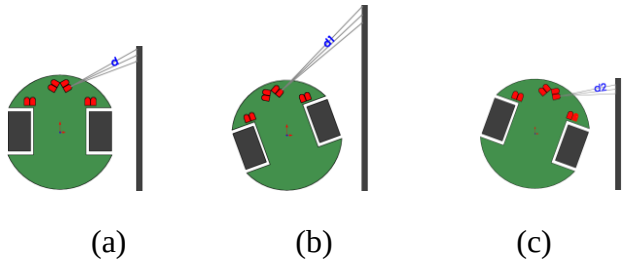
Al usar los sensores de esta manera no se puede discernir si el robot se está alejando o acercando a la pared, como se muestra en la Fig. 5 (b) y en la Fig. 5 (c) ya que las lecturas del sensor dan un valor más alto del que debería ser en ambos casos.



(a) (b) (c)

Fig. 5 Disposición de sensores ortogonalmente. (a) Robot paralelo a la pared. (b) Robot con un ángulo de inclinación de 25º a la izquierda. (c) Robot con un ángulo de inclinación de 25º a la derecha

Otra opción es colocar los sensores con un cierto ángulo como se observa en la Fig. 6 (a), con esta configuración se conoce si el robot se está alejando o acercando a la pared, ya que el robot al estar desviado como en la Fig. 6 (b) proporciona un valor mayor del que debería ser y al estar desviado como en la Fig. 6 (c) entrega un valor menor.



(a) (b) (c)

Fig. 6 *Disposición de sensores con un ángulo de 30º. (a) Robot paralelo a la pared. (b) Robot con un ángulo de inclinación de 25º a la izquierda. (c) Robot con un ángulo de inclinación de 25º a la derecha.*

Para la selección del ángulo (α) de los sensores laterales se coloca al robot en una de las celdas de laberinto de tal manera que el eje que une las ruedas pase por el centro de ésta (Fig. 7) y se debe apuntar los sensores al final de la celda. Con los sensores colocados de ésta manera se puede detectar con anticipación si existe un camino libre y la distancia suficiente para poder realizar el giro.

Los sensores delanteros son orientados perpendicularmente a la pared frontal, se usa dos sensores ya que así puede estimar mejor la distancia entre el robot y la pared evitando que el robot se acerque demasiado en caso de llegar con cierta desviación.

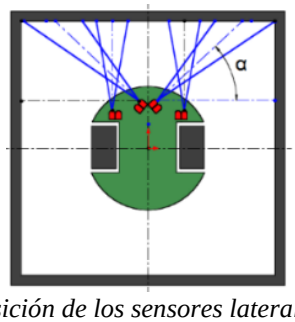


Fig. 7 Disposición de los sensores laterales y frontales

Para la selección de los motores se tiene en cuenta la velocidad que se quiere alcanzar, y el torque y se utiliza la ecuación (2).

Revista Nexos Científicos
Julio –Diciembre 2019 pp. 28-42
Volumen 3, Número 2
Fecha de recepción: octubre 2019
ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2019

IV. DISEÑO ELECTRÓNICO

A. Robot seguidor de línea

El robot, como se lo puede ver en la Fig. 8, está compuesto de un sistema de control, de un sistema sensorial en el cual están los encoders, sensores de línea y medición de batería, un sistema de periféricos de entrada, salida y un controlador de potencia con el cual se accionan los motores.

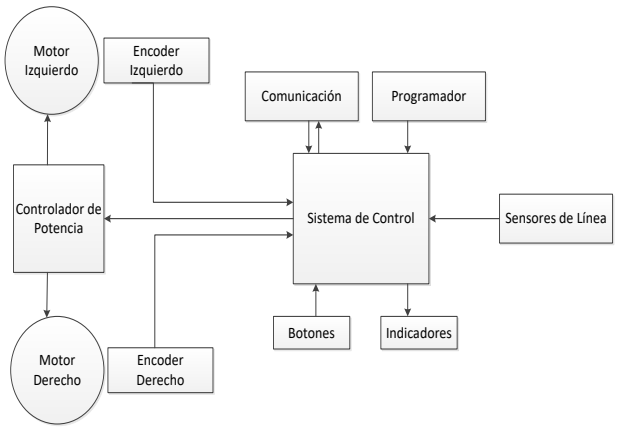


Fig. 8 Arquitectura del robot seguidor velocidad

El cerebro del seguidor de línea es un microcontrolador ATmega 324PA, tiene 32 Kbyte de memoria de programa, 1 Kbyte de memoria EEPROM, y 2 Kbyte de memoria SRAM. Algunos de sus periféricos son: dos temporizadores de 8 bits con pre escalamiento separado, un temporizador de 16 bits con pre escalamiento, 8 canales ADC de 10 bits, dos puertos seriales USART, entre otros.

Para la detección de la línea en la pista se usa una matriz de sensores QTR- 8A, la cual viene con 8 pares led-fototransistor, la separación entre sensores es de 9.52 mm. Proporciona 8 salidas analógicas con las cuales se detecta la posición de la línea. En la Fig. 9 se muestra el arreglo de sensores (Pololu, 2015).

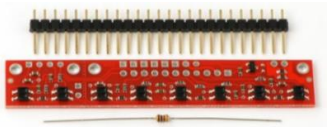


Fig. 9 Sensores de reflectancia analógicos QTR-8 A

También se utiliza un encoder de tipo incremental para medir la distancia que recorre el robot. Este encoder utiliza un disco magnético de seis polos y dos sensores de efecto de Hall. A la salida se tiene dos señales digitales en cuadratura con lo cual se obtiene hasta 12 pulsos por revolución del eje cuando se lee ambos bordes de los dos canales. En la Fig. 9 se puede apreciar este tipo de encoder (Pololu, 2016).

Fig. 10 Encoder magnético de cuadratura

Para darle movimiento al prototipo se utiliza un motorreductor metálico de corriente continua con escobillas (Fig. 10), en el cual viene incorporada una caja reductora con relación 10:1, en la parte posterior del motor tiene una extensión del eje para la colocación de algún tipo de encoder. La tensión nominal de alimentación es de 6 V aunque también puede trabajar con voltajes mayores teniendo en cuenta de que con tensiones superiores a 9 V se reduce significativamente la vida útil de motor (Pololu, 2016).

Fig. 11 Micromotor 10:1 HP de robot extendido

Características:

- Voltaje nominal: 6 V
- Velocidad: 3000 RPM
- Corriente en vacío: 120 mA
- Máxima corriente: 1600 mA
- Torque: 0.28 kg-cm
- Relación de la caja reductora 10:1
- Peso: 9.5 gr

Para accionar los motores se utiliza el controlador de potencia TB6612FNG que trae dos puentes H para controlarlos, sus características son:

- Voltaje de motor: 4.5 a 13.5 V.
- Voltaje lógico: 2.7 a 5.5 V.
- Salida de corriente máxima por canal: 3.2 A.

Revista Nexos Científicos
Julio –Diciembre 2019 pp. 28-42
Volumen 3, Número 2
Fecha de recepción: octubre 2019
ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2019

- Salida de corriente continua: 1.2 A.
- Máxima frecuencia de PWM: 100 kHz.

B. Robot laberinto

Para el control del prototipo se utiliza el microcontrolador ATmega 324PA como en el seguidor de línea. Para la detección de las paredes se utiliza cuatro sensores infrarrojos acondicionados para medir hasta una distancia de 30 cm. Los sensores están construidos por el led infrarrojo IR333 y un fototransistor PD333. En la Fig. 11 se muestra el circuito de acondicionamiento del sensor infrarrojo de distancia.

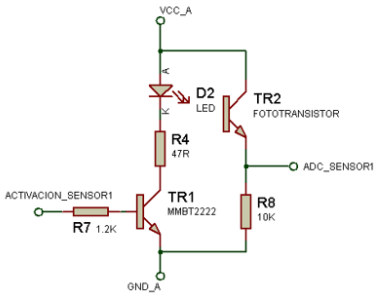


Fig. 12 Circuito del sensor de distancia del robot laberinto

Para activar el led infrarrojo se utiliza el transistor MMBT2222 que soporta hasta 150 mA a través de colector-emisor.

Con el fin de realizar giros de mayor precisión se utiliza un encoder óptico incremental (Fig. 12). A la salida del encoder se tiene dos señales digitales en cuadratura pudiéndose obtener hasta 48 pulsos por revolución de la rueda.

Fig. 13 Circuito del sensor de distancia del robot laberinto

V. SOFTWARE

Para la codificación de los programas que controlan los robots se utilizó el lenguaje de programación C. Los programas se los dividió en varios archivos los cuales contienen secciones de código para controlar diferentes partes de los robots y los algoritmos que les permiten a éstos cumplir con sus objetivos de manera autónoma.

A. Robot seguidor de línea

Los algoritmos de control a implementarse en el seguidor de línea son el proporcional-integral-derivativo (PID) y borroso con el objetivo de que el robot siga la línea a alta velocidad y que el error con respecto a ésta sea cercano a cero, pero en realidad el único parámetro a tomar en cuenta en el rendimiento del controlador es el tiempo que le toma al robot en recorrer la pista.

Para que el seguidor cumpla con su objetivo primero se realiza el cálculo del error de posición con respecto a la línea, este dato va al controlador (PID o Borroso) y de éste se obtienen las señales PWM que van al driver de potencia que acciona los motores. Para realizar el cálculo del error de posición primero se realiza un acondicionamiento de los sensores de línea.

1) Acondicionamiento de sensores: Se debe calibrar los sensores antes de cada competencia, el objetivo de la calibración es hacer que todos los sensores se comporten de manera similar y se adapten a las condiciones de luz del lugar.

Para la calibración se coloca al robot sobre la pista y se lo hace girar de modo que todos los sensores pasen sobre las superficies blanca y negra; el algoritmo de calibración guarda los valores máximo (superficie negra) y mínimo (superficie blanca) de cada sensor en memoria.

Después con los valores máximo y mínimo de cada sensor se realiza un escalamiento (3) entre 0 y 1000, como se aprecia en la Fig. 13, donde 0 corresponde al valor mínimo (el sensor esta sobre una superficie blanca) y 1000 corresponde al valor máximo (el sensor esta sobre una superficie negra).

Revista Nexos Científicos
Julio –Diciembre 2019 pp. 28-42
Volumen 3, Número 2
Fecha de recepción: octubre 2019
ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2019

$$V_N = \frac{1000 * (V_B - V\min)}{V\max - V\min} \quad (3)$$

Donde:

V_N *: Valor acondicionado*
V_B *: Valor entregado por el sensor*
$V\max$ *: Valor máximo*
$V\min$ *: Valor mínimo*

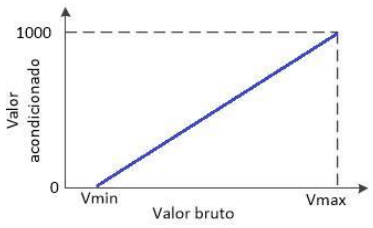


Fig. 14 Escalamiento de sensores

2) Cálculo del error de posición: Para calcular el error de posición respecto a la línea existen varios algoritmos, como: interpolación cuadrática y promedio ponderado. En este trabajo se utilizó el promedio ponderado ya que es el algoritmo que tiene menor error en la estimación (Alonso, 2015).

Con este algoritmo se toman los valores ya escalados de cada sensor y se usa (4) para realizar el cálculo del error.

$$Error\ e(n) = \frac{\sum_{0}^{7} L_i * (2i)}{L_i} - 14 \quad (4)$$

Donde:

L_i *: I-ésimo sensor de línea.*
$e(n)$ *: Error*

Para valores positivos el robot esta desviado a la izquierda de la línea y para valores negativos esta desviado a la derecha.

3) Controlador Proporcional – Integral – Derivativo (PID): El objetivo del controlador PID es ubicar al robot en el centro de la línea variando las velocidades de las ruedas por medio de una acción de control u(n). Se parte de la ecuación de un controlador proporcional integral derivativo en forma paralela (5).

$$u(t) = k_p * e(t) + k_I * \int_0^t e(t) * d(t) + k_D * \frac{d\,e(t)}{dt} \quad (5)$$

Para poderlo aplicar se lo discretizó. El término que representa el tiempo de muestreo se lo simplifica debido a que es una constante que se multiplica por las constantes k_I y k_D.

$$u(n) = K_P * e(n) + K_I * \sum_0^n e(n) + K_D * (e(n) - e(n-1)) \quad (6)$$

Donde: $K_I = k_I * ts$ y $K_D = k_D / ts$

La implementación directa del controlador propuesto tiene un problema con la parte integral ya que funciona bien cuando la pista se compone de una curva simple pero cuando ésta tiene varias curvas y rectas el robot se desestabiliza.

Para resolver los problemas que provoca la implementación directa de la acción integral se realiza una modificación, se toma únicamente los últimos 10 valores para esta acción en lugar de todos, ecuación (##):

$$u(n) = K_P * e(n) + K_I * \sum_{n-9}^n e(n) + K_D * (e(n) - e(n-1)) \quad (7)$$

Esto permite un esfuerzo de control integral para atacar el error en estado estable que se da en las curvas a la vez que también da la adaptabilidad a los cambios de curvatura de la pista.

Las velocidades de los motores se establecen en función de dos parámetros, la salida del controlador y la velocidad del robot (velocidad crucero), las ecuaciones **¡Error! No se encuentra el origen de la referencia.**, **¡Error! No se encuentra el origen de la referencia.**, **¡Error! No se encuentra el origen de la referencia.** y **¡Error! No se encuentra el origen de la referencia.** representan las velocidades de cada motor.

Revista Nexos Científicos
Julio –Diciembre 2019 pp. 28-42
Volumen 3, Número 2
Fecha de recepción: octubre 2019
ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2019

Si el error es negativo:

$$motor_{ixquierdo} = velocidad_{robot} + u(n) \quad (8)$$

$$motor_{derecho} = velocidad_{robot} \quad (9)$$

Si el error es positivo:

$$motor_{ixquierdo} = velocidad_{robot} \quad (10)$$

$$motor_{derecho} = velocidad_{robot} - u(n) \quad (11)$$

De lo mencionado anteriormente se mira que cuando el robot se desvía a la derecha (error negativo), el motor derecho se mantiene con la velocidad de crucero mientras que el motor izquierdo reduce la velocidad para que el robot se mueva a la izquierda; y cuando está desviado a la izquierda (error positivo) el motor izquierdo se mantiene con la velocidad de crucero y el motor derecho reduce su velocidad.

El control PID que se implementó funciona bien para líneas rectas y curvas moderadas pero cuando las curvas son de radios pequeños (radio menor a 30 cm) el desempeño del prototipo no es muy bueno. Esto debido a que cuando el robot pierde la línea hace que el error se sature. Bajo estas circunstancias el esfuerzo de control producido por el PID no puede hacer girar al robot lo suficientemente rápido, para resolver este problema se utiliza un esquema de control de lazo abierto.

Esta idea consiste en setear ciertas velocidades a los motores dependiendo del sentido de la curva y de su radio hasta que el robot detecta nuevamente la línea y pueda seguir con el lazo de control.

4) Controlador borroso: El sistema de control consta de dos entradas y una salida, donde las entradas son: el error (e) que es la desviación actual del robot con respecto a la línea menos la desviación que se desea (desviación cero) y Δe que representa el cambio del error. La variable de salida u representa la diferencia de velocidades de las ruedas para que el robot siga la línea y tenga un error de cero (sin desviación); en la Fig. 14 se indica el esquema de control.

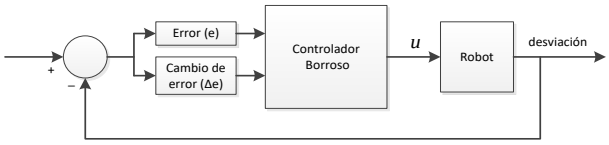


Fig. 15 Esquema de control borroso para el robot seguidor de línea

En el diseño del controlador borroso se parte de la borrosificación, consiste en calcular el valor de pertenencia de las variables de entrada $(e, \Delta e)$ a las etiquetas lingüísticas $(E_i \ , \ \Delta E_i)$ utilizando las funciones de pertenencia. Para esto primero es necesario establecer el universo de discurso para cada una de las variables.

Para la definición del número de etiquetas lingüísticas que se debe usar no existe una regla formal, todo depende del diseñador (CER, 2014). Para la variable error se utiliza tres funciones de membresía cuyas etiquetas lingüísticas son: negativo (N), cero (C) y positivo (P).

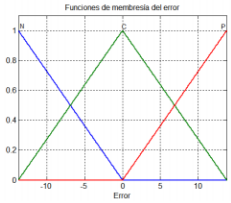


Fig. 16 Funciones de membresía del error

Para la variable cambio de error (Δe) también se utilizó tres etiquetas lingüísticas (Fig. 16), negativa (N), cero (C) y positiva (P).

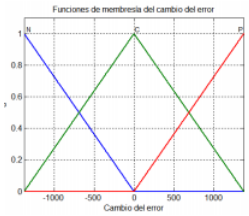


Fig. 17 Función de membresía del cambio del error

Los nombres que se les dé a las etiquetas no son determinantes en el desempeño del controlador borroso, éstas solo ayudan a identificar a cada función de membresía. Se usó funciones de membresía tipo triangulares ya que son las de menor coste computacional teniendo en cuenta que el controlador se va a implementar en un microcontrolador,

Revista Nexos Científicos
Julio –Diciembre 2019 pp. 28-42
Volumen 3, Número 2
Fecha de recepción: octubre 2019
ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2019

En esta parte también se definen las funciones de membresía de la acción de control: negativa muy grande (NMG), negativa grande (NG), negativa mediana (NM), negativa pequeña (NP), cero (C), positiva pequeña (PP), positiva mediana (PM), positiva grande (PG) y positiva muy grande (PMG). En la Fig. 3.14 se aprecia las funciones de membresía para la acción de control.

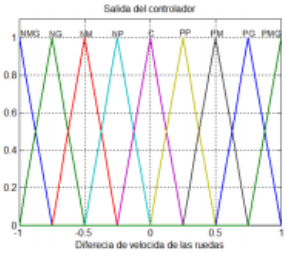


Fig. 18 Función de membresía de la salida del controlador

Con el diseño que se propone, cuando el error es positivo (desviación a la izquierda) la rueda derecha girará a menor velocidad que la izquierda, cuando el error es negativo (desviación a la derecha) la rueda izquierda girará a menor velocidad que la derecha, y cuando el error sea cero las ruedas girarán a la misma velocidad.

El siguiente punto es establecer las reglas de control, es necesario cubrir todos los posibles valores que pueden darse en las entradas del controlador y así asegurar que en su salida exista un valor concreto y exacto que promueva una acción de control óptima (N. B. Kiat, 2015).

El número de entradas y el número de funciones de membresía por cada entrada influye en el número de reglas que puede tener el controlador borroso, por lo que si se tiene en cuenta que el diseño posee dos entradas y tres funciones de membresía por cada una de ellas, van a existir 9 reglas que pueden ser activadas.

Ahora, para la selección de las reglas, el grado de cumplimiento de la premisa puede tomarse como el producto de las condiciones (12), tomándose éste valor para la conclusión final, es decir que se toma como operador lógico para la intersección la función producto. Cada regla tiene como peso w, donde, w es el producto de los respectivos grados de pertenencia de e y Δe a las etiquetas correspondientes a esa regla.

$$w_i = u_{Ei}(e) * u_{\Delta ei}(\Delta e) \qquad (12)$$

Hasta aquí se dispone de un conjunto de reglas y de un peso de cada una para la conclusión final, para calcular la conclusión de cada una de las reglas se multiplica la función primitiva por el peso (13).

$$u_{Ui}(u) = w_i * u_{Ui}(u) \quad \forall u, \forall i \qquad (13)$$

El resultado final de la aplicación de todas las reglas es una serie de conjuntos borrosos con sus respectivas funciones de pertenencia. Con el fin de obtener un único conjunto a partir de los anteriores se utiliza la función máximo.

$$u_U(u) = max(u_{Ui}(u)) \qquad (14)$$

Como salida se tiene un conjunto borroso pero éste no se lo puede aplicar a los actuadores directamente por los que es necesario una etapa de desborrosificación para poder obtener un valor numérico a partir del conjunto borroso de salida, existen varios métodos para esto, en este trabajo se empleó el método del centroide, ecuación (3.18); en la que la salida es el centro de gravedad del área total resultante; es el método que más información toma en cuenta para el cálculo y contribuye a que la salida se mueva suavemente.

$$cg = \frac{\sum y u_D(y)}{\sum u_D(y)} \qquad (15)$$

B. Robot laberinto

Los algoritmos que se implementan en el robot laberinto son: algoritmo de la mano derecha, algoritmo de la mano izquierda y algoritmo de relleno de callejones sin salida.

Lo primero que hace el programa es medir a que distancia se encuentra el robot de las paredes del laberinto y detectar los posibles caminos que puede tomar para lo cual primero se realiza un acondicionamiento de los sensores de pared.

1) Acondicionamiento del sensor de pared: Para medir la distancia del robot con respecto a las paredes primero se disminuye los efectos de la luz ambiental sobre el sensor, para lo cual se lee el

Revista Nexos Científicos
Julio –Diciembre 2019 pp. 28-42
Volumen 3, Número 2
Fecha de recepción: octubre 2019
ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2019

sensor con el emisor apagado (proporciona una medida de la luz ambiental), luego se lee el sensor con el emisor encendido (proporciona una medida de la luz ambiental más la luz infrarroja del led emisor), el valor del sensor con el emisor apagado se resta del valor del sensor con el emisor encendido dando como resultado el valor de la luz infrarroja del led emisor reflejada en la pared quitando la luz ambiental, esto ayuda a evitar que los sensores sean afectados por los cambios de la luz ambiental.

El valor obtenido solo es una medida de la reflexión de la luz infrarroja emitida por el led por lo que se debe transformar a unidades de distancia (mm), para lo cual se utiliza una regresión. Para la realización de la regresión se toman datos del voltaje que da el sensor y de la distancia de éste con respecto a la pared (los datos se tomaron cada 10 mm), a los sensores se los acondicionó para medir una distancia de hasta 30 cm con lo cual se puede prever una pared con más de una celda de anticipación (una celda mide 25 x 25 cm).

En la figura 18 se puede apreciar el comportamiento del sensor, en el eje de las abscisas se tiene el valor del conversor análogo digital y en el eje de las ordenadas la distancia con respecto a la pared.

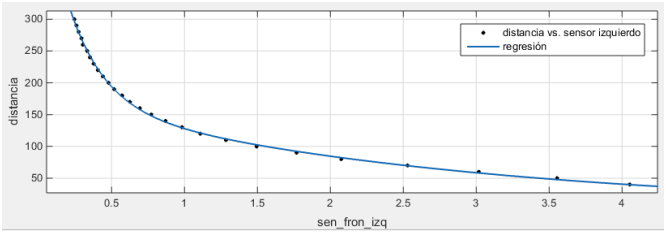


Fig. 19 Distancia en mm vs lectura del sensor frontal derecho

Al obtener la curva, se puede ver claramente la formación de una función exponencial. Con ayuda de Matlab se realizó una regresión para obtener la ecuación que mejor se ajusta a los datos obtenidos, la ecuación resultante es

$$f(x) = \ a * \ e^{b*x} + c * \ e^{d*x} \quad (16)$$

La ecuación tiene cuatro parámetros: a, b, c y d que se los calcula con Matlab y que permiten ajustar la curva para que concuerde con los datos obtenidos. Entonces, la lectura que da el sensor es ingresada en la ecuación **¡Error! No se encuentra el origen de la referencia.** dando como resultado la distancia del sensor con respecto a la pared.

- Detección de paredes

Se realiza una medición de la distancia de cada sensor a la pared y en base a una distancia referencia de cada una se determina si existe o no pared

- Obtener el error de posición

El error de posición (en milímetros) que tiene el robot con respecto a una pared se calcula midiendo la distancia que tiene el sensor con respecto a ésta y se lo resta del set point (valor de la mitad de una celda), con esto se puede conocer la desviación que tiene el robot.

2) Algoritmo de la mano derecha: Este algoritmo realiza el seguimiento de la pared derecha mientras la tenga, si no existe la pared el robot girará a la derecha. Se debe realizar dicho proceso de forma recursiva hasta solucionar el laberinto.

3) Algoritmo de la mano izquierda: El algoritmo de la mano izquierda funciona de manera similar al de la mano derecha pero usando la pared izquierda. Estos algoritmos son los más utilizados para resolver laberintos que tengan la entrada y salida en las paredes externas y no posean islas en su interior.

4) Algoritmo de relleno de callejones sin salida: El algoritmo tiene como objetivo rellenar todos los callejones que lleven al robot a un camino que provoque un giro de 180º, se realiza un primer recorrido con un algoritmo de detección de caminos sin salida, en el cual se va guardando en un vector las intersecciones que tengan dos o más posibles caminos y los caminos sin salida que existan en el recorrido.

Después de obtener este vector se usa un algoritmo de optimización de caminos, el cual devuelve un vector en el que están eliminados todos los caminos sin salida; al obtener el vector optimizado, la segunda vez que el robot recorra el

Revista Nexos Científicos
Julio –Diciembre 2019 pp. 28-42
Volumen 3, Número 2
Fecha de recepción: octubre 2019
ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2019

laberinto se usa la rutina de resolución del laberinto evitando caminos sin salida, cada que llegue a una intersección el robot revisa en el vector de memoria que giro debe realizar y así evita entrar innecesariamente a caminos ayudando a disminuir el tiempo de resolución del laberinto.

VI. PRUEBAS Y RESULTADOS

A. Robot seguidor de línea

Se tomó como referencia la pista del Concurso Ecuatoriano de Robótica 2015 (Fig. 4.5). La pista es completamente plana, no tiene intersecciones ni discontinuidades, está hecha con cinta aislante negra en un tablero triplex de color blanco.

Fig. 20 Robot seguidor de línea

El desempeño del robot se lo evalúa por medió del tiempo que le toma recorrer la pista (el objetivo es recorrer la pista en el menor tiempo posible), el robot no debe salirse de la pista ni acortar camino (el reglamento no lo permite).

Lo primero que se probó fueron las ruedas, se probó con algunos materiales como: caucho, silicona, goma y espuma.

Las pruebas arrojaron que las ruedas de goma, silicona y caucho (Fig. 4.6) no son las adecuadas ya que éstas no dan suficiente adherencia al prototipo, no importa los ajustes que se le haga al controlador, no es posible incrementar la velocidad del robot ya que éste derrapa fácilmente y pierde pista, otra desventaja que tienen este tipo de ruedas es que les afecta mucho la suciedad (polvo) que puede estar en la pista por lo que hay que limpiarla continuamente.

(a) (b)

Fig. 21 Ruedas. (a) Ruedas de silicona. (b) Ruedas de goma

Las mejores ruedas para el prototipo fueron las de espuma (Fig. 4.7), éstas dieron la suficiente adherencia para poder incrementar la velocidad del prototipo además que a éstas no les afecta demasiado la suciedad que pueda haber en la pista.

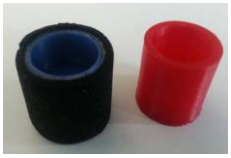


Fig. 22 Ruedas de espuma

La Tabla I se presenta los resultados obtenidos para algunas distancias del eje de las ruedas a los sensores. El objetivo para el robot seguidor de línea es superar el tiempo del ganador del Concurso Ecuatoriano de Robótica 2015 el cual recorrió la pista en 11.4.

Tabla I
Tiempos obtenidos con diferentes distancias de los sensores al eje de las ruedas.

Distancia de los sensores al eje de las ruedas	Control PID	Controlador Borroso
9 cm	11.2 s	18.2 s
13.5 cm	10.9 s	17.9 s
15 cm	10.6 s	16,4 s
16.5 cm	10.8 s	17.3 s

De la Tabla I se puede apreciar que el controlador con el que mejores resultados se obtuvo fue el PID ya que los tiempos en recorrer la pista son menores.

B. Robot laberinto

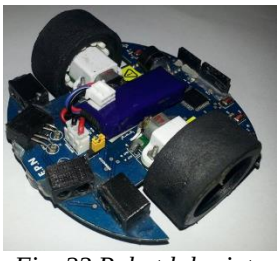
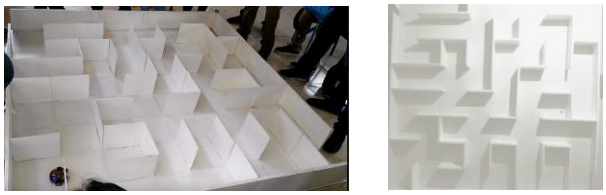


Fig. 23 Robot laberinto

Las pruebas se realizaron en base a la pista del Concurso Ecuatoriano de Robótica 2015 (Fig. 4.11). El objetivo para el robot es solucionar un

ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2019

laberinto en el menor tiempo posible sin topar las paredes, ya que recibiría una penalización de acuerdo al reglamento. La dimensión del laberinto es de 2 x 2 m y una distancia entre paredes de 25 cm. El ganador del Concurso Ecuatoriano de Robótica 2015 resolvió el laberinto en un tiempo de 30 segundos.



(a) (b)

Fig. 24 Pista de robot laberinto. (a) Pista del CER 2015. (b) Pista recreada para pruebas

Para las pruebas primero se usó el algoritmo de la mano derecha. Con este algoritmo el robot consiguió superar el laberinto en un tiempo aproximado de 25 s, es posible bajar este tiempo incrementando la velocidad de navegación pero al hacerlo la probabilidad de choque aumenta lo cual no es deseado ya que de acuerdo al reglamento se recibe una penalización.

Después se usó el algoritmo de la mano izquierda, con lo cual el robot consiguió solucionar el laberinto en un tiempo de 21 s. Aunque el algoritmo anteriormente probado funciona de forma similar se tiene resultados diferentes en tiempo debido a que la distancia que recorre el robot para resolver el laberinto es menor por uno de ellos.

Finalmente con el algoritmo de relleno de callejones sin salida se tiene que recorrer el laberinto dos veces, debido a que la primera vez el robot memoriza las intersecciones y caminos sin salida que existan en su recorrido, la segunda vez que lo recorre evita entrar a esos caminos sin salida para poder resolverlo en un menor tiempo. Con este método se obtuvo un tiempo de 35 s en el primer recorrido, después, al realizar el segundo recorrido (sin recorrer callejones sin salida) se obtuvo un tiempo de 26 s. En la Tabla II se muestra la comparación en tiempo que se demora en solucionar el laberinto con los algoritmos implementados.

Se debe tener en cuenta que el tiempo que se demora en solucionar el laberinto por los diferentes algoritmos dependerá del diseño del laberinto.

Tabla II
Comparación en tiempos de los diferentes algoritmos de resolución.

Algoritmos de resolución	Tiempo
Algoritmo de la mano derecha	25 s
Algoritmo de la mano izquierda	21 s
Algoritmo de relleno de caminos sin salida	26 s

VII. CONCLUSIONES

El controlador que mejores resultados dio en el robot seguidor de línea fue el PID ya que la puesta a punto del robot es más rápida frente al controlador borroso lo cual fue fundamental en los torneos de robótica ya que el tiempo que se tiene para calibraciones es limitado. Al PID se lo calibra con las constantes proporcional, integral y derivativa en contraparte del controlador borroso en donde no hay constantes para calibrar, en éste se debe cambiar las reglas de inferencia o las funciones de membresía lo cual conlleva más tiempo.

La implementación del control en lazo abierto cuando el arreglo de sensores pierde completamente la línea permitió al robot mejorar su rendimiento ya que el robot pudo tomar las curvas de radio pequeño (radio menor a 30 cm) de mejor manera y así disminuir el tiempo en recorrer la pista.

Para el cálculo del error de posición se utilizó un promedio ponderado ya que este es el algoritmo que menor error de estimación presenta lo cual es fundamental ya que el robot seguidor podrá seguir la línea de mejor manera.

Se mejoró la lectura del error de posición del robot seguidor con respecto a la línea mediante la calibración de los sensores haciendo que éstos se comporten de manera similar y se adapten a las condiciones de luz del lugar.

ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2019

Una mala elección de las ruedas en el robot seguidor de línea hará bajar el rendimiento del prototipo sin importar los ajustes que se le hagan al controlador, el tiempo en el cual el prototipo supera la pista va a ser alto ya que la velocidad del robot debe ser baja para que no derrape y pierda pista.

Se mejoró la lectura de distancia de los sensores infrarrojos del robot laberinto por medio de un algoritmo que permite eliminar en un gran porcentaje la luz ambiental, ya que ésta tiene componentes infrarrojos que afectan al correcto funcionamiento de los sensores.

La ubicación de los sensores laterales de pared es de suma importancia ya que influye en la rapidez con la que el robot navega por el laberinto, en éste trabajo se ubicó a los sensores con un ángulo de 34°, de tal forma que le permita saber al controlador si el robot se está acercando o alejando de la pared.

Con respecto a la navegación del robot laberinto es necesario realizarla por el centro de la celda, minimizando la probabilidad de choque con las paredes, para esto se implementó un controlador que con el cual se obtuvo una navegación correcta sin ningún choque.

Los tres algoritmos implementados se basan en el seguimiento de paredes, con los cuales se lograron resultados favorables para la solución de laberintos con las especificaciones del concurso ecuatoriano de robótica.

El tiempo de resolución de un laberinto siempre dependerá del diseño del laberinto de cada concurso, ya que cada algoritmo tiene una forma de resolver.

El robot laberinto con el algoritmo de relleno de callejones sin salida se ve limitado a trabajar a bajas velocidades, debido a que posee una forma de navegación más compleja que el de mano derecha o izquierda, ya que este reconoce en todo su trayecto las paredes que tiene a su alrededor. Es por esto que este algoritmo podría demorarse un mayor tiempo en resolver un laberinto.

Para el buen funcionamiento de los robots y que éstos alcancen un nivel competitivo la parte mecánica como la de control debe hacer sinergia ya que si una falla los resultados no van a ser buenos.

VII. DISCUCIÓN DE LOS RESULTADOS

En la implementación de estos robots se pudo identificar los parámetros de diseño mecánico, eléctrico y de programación que se debe tener en cuenta para el diseño de cada uno de los robots permitiendo identificar un posible diseño que permita la mejor movilidad en cada una de las categorías de competencia. Con esto se pretende dar un referente para realizar nuevas investigaciones dentro de estas categorías y ais mejorar los diseños.

REFERENCIAS

Robot Challenge, Winners of RobotChallenge 2016, Robot Challenge, 13 Marzo 2016. [En línea]. Available: https://www.robotchallenge.org/robotchallenge/resultate-2016/. [Último acceso: 24 Marzo 2016].

J. S. Tercero, Cibernética aplicada Robots educativos, México: Alfaomega, 2009.

CER, Reglamento para la categoría "Seguidor de línea", Concurso Ecuatoriano de Robótica 2014, 9 Diciembre 2014. [En línea]. Available: http://www.cer2015.com/index.php/bases-del-concurso. [Último acceso: 18 Diciembre 2015].

J. Alonso, Proyecto Reborn, Hiyalife, 15 Julio 2014. [En línea]. Available: http://hiyalife.com/meemo/95e747ede9. [Último acceso: 20 12 2015].

N. B. Kiat, Proyecto futura, Micromouse USA, 29 noviembre 2011. [En línea]. Available:

ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2019

http://micromouseusa.com/?page_id=1342. [Último acceso: 20 diciembre 2015].

B. García García, Diseño y construcción de un robot móvil controlado con técnicas de lógica difusa implementadas en un FPGA, México D.F: Instituto Politécnico Nacional, 2012.

S. Ortigoza, Una panorámica de los robots móviles, Publicaciones urbe, 12 Febrero 2007. [En línea]. Available: http://publicaciones.urbe.edu/index.php/telematique/article/viewArticle/833/2 037. [Último acceso: 12 Enero 2015].

Pololu, High-Power (HP) Micro Metal Gearmotors, Pololu, 1 Enero 2001. [En línea]. Available: https://www.pololu.com/product/999. [Último acceso: 12 Enero 2016].

Pololu, Micro Metal Gearmotors with Extended Motor Shafts, Pololu, 1 Enero 2001. [En línea]. Available: https://www.pololu.com/product/2211. [Último acceso: 12 Enero 2016].

C. Electrónica, Data sheet A139 Gearmotors, Carrod Electrónica, 1 Marzo 2010. [En línea]. Available: http://www.carrod.mx/products/motores-reductor48-1-65-rpm. [Último acceso: 12 Enero 2016].

Pololu, QTR-8A Reflectance Sensor Array, Pololu, 1 Enero 2001. [En línea]. Available: https://www.pololu.com/product/960. [Último acceso: 21 Diciembre 2015].

Polulo, QTR-8RC Reflectance Sensor Array, Pololu, 1 Enero 2001. [En línea]. Available: https://www.pololu.com/product/961. [Último acceso: 12 Enero 2016].

Sparkfun, SparkFun Line Follower Array, Sparkfun, 31 Julio 2007. [En línea]. Available: https://www.sparkfun.com/products/13582. [Último acceso: 21 Diciembre 2015].

Pololu, Sharp GP2Y0A51SK0F Analog Distance Sensor, Pololu, 1 Enero 2001. [En línea]. Available: https://www.pololu.com/product/2450. [Último acceso: 12 Enero 2016].

Pololu, Parallax PING Ultrasonic Sensor, Pololu, 1 Enero 2001. [En línea]. Available: https://www.pololu.com/product/1605. [Último acceso: 12 Enero 2016].

Pololu, Magnetic Encoder Pair Kit for Micro Metal Gearmotors, Pololu, 1 Enero 2001. [En línea]. Available: https://www.pololu.com/product/2598. [Último acceso: 12 Enero 2016].

Pololu, Optical Encoder Pair Kit for Micro Metal Gearmotors, Pololu, 1 Enero 2001. [En línea]. Available: https://www.pololu.com/product/2590. [Último acceso: 12 Enero 2016].

Pololu, Encoder for Pololu Wheel 42x19mm, Pololu, 1 Enero 2001. [En línea]. Available: https://www.pololu.com/product/1217. [Último acceso: 12 Diciembre 2015].

Pololu, 3pi Robot, Pololu, 1 Enero 2001. [En línea]. Available: https://www.pololu.com/product/975. [Último acceso: 2 Enero 2016].

The Rutgers IEEE Robotics club, Line Following Kit, IEEE, 1 Enero 2011. [En línea]. Available: http://ieee.rutgers.edu/line-following-kit. [Último acceso: 31 Diciembre 2015].

ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2019

Robomart, Robomart White Line Follower Robot, Robomart, 4 Febrero 2014. [En línea]. Available: https://www.robomart.com/robomart-white-line-followerrobot. [Último acceso: 31 Diciembre 2015].

PICAXE, Data sheet PICAXE, PICAXE, 3 Marzo 2011. [En línea]. Available: http://www.picaxe.com/Hardware/Robot-Kits/PICAXE-PICone-Micromouse/. [Último acceso: 31 Diciembre 2015].