

Desempeño de Algoritmos Gusano en la Navegación de una Plataforma Robótica Móvil

Luis M. Álvarez, Juan E. Figueroa, Rodolfo X. Gordillo y Luis A. Chacón

Resumen— El presente trabajo presenta la implementación de algoritmos de navegación denominados Gusano 1 y Gusano 2 sobre la plataforma robótica móvil de iRobot Create, los cuales le brindaron la capacidad de autonomía para navegar dentro de entornos desconocidos. Además, se utilizaron a los módulos de radio frecuencia XBee, que funcionan bajo el estándar ZigBee, para dotar de comunicación inalámbrica entre el robot y una PC. Se realizaron pruebas de comunicación para determinar la máxima distancia de trabajo a la que pueden funcionar los módulos de RF en interiores; mientras que, para el análisis de desempeño de los algoritmos básicos de navegación montados sobre la plataforma robótica móvil se definieron tres escenarios diferentes donde se verificó la certeza de haber alcanzado la meta por el robot.

Palabras clave Navegación; Algoritmos Gusano; IEEE 802.15.4; ZigBee; certeza.

I. INTRODUCCIÓN

En la actualidad, la Robótica está orientándose a los entornos inciertos e impredecibles dejando a un lado los lugares estáticos; lo que ha conducido al reemplazo de los tradicionales manipuladores robóticos por robots móviles. La utilización de robots móviles en aplicaciones industriales y técnicas es amplia y continúa creciendo.

En función de los avances producidos en el campo computacional, tecnologías de comunicación, sensores y dispositivos de navegación, los robots móviles han desarrollado su capacidad de autonomía. Estos robots son capaces de modificar sus comportamientos autónomamente al momento que se origine cualquier cambio en su entorno, es decir, que no requieren de la intervención de una persona para tomar alguna decisión porque la reforma de sus comportamientos está en base a la percepción del entorno real. Por lo tanto, los robots móviles autónomos deberían ser dispositivos capaces actuar de forma independiente y adaptar sus comportamientos en un entorno dinámico e incierto.

L.M. Álvarez, es graduado de Ingeniero en Electrónica, Automatización y Control en la Universidad de las Fuerzas Armadas Espe, Sangolquí Ecuador, (email:juane_figueroa@hotmail.com)

J.E. Figueroa, es graduado de Ingeniero en Electrónica, Automatización y Control en la Universidad de las Fuerzas Armadas Espe, Sangolquí Ecuador, (email:lmiguel_ar@hotmail.com)

R.X. Gordillo, es profesor del Departamento de Eléctrica y Electrónica en de en la Universidad de las Fuerzas Armadas Espe, Sangolquí Ecuador, (email:rxgordillo@espe.edu.ec).

L.A. Chacón, es profesor del Departamento de Eléctrica y Electrónica en de en la Universidad de las Fuerzas Armadas Espe, Sangolquí Ecuador, (email:lachacon@espe.edu.ec).

Para interactuar de una manera óptima y correcta con el entorno, estos dispositivos autónomos pueden utilizar un conocimiento previo de éste o procesar la información entregada por sus sensores para construir una representación adecuada tanto de su entorno como de su ubicación

Es así que, uno de los principales retos de la Robótica actual radica en el hecho de generar un planeamiento de movimiento autónomo, en donde, el objetivo apunta a implementar una tarea en lenguaje de alto nivel que al ser compilada se convierta en un conjunto de movimientos primitivos de bajo nivel que deberán ser ejecutados por el robot para su consecución. Entonces, para aprovechar la movilidad presente en este tipo de robots se requiere desarrollar habilidades de navegación, que le permitan a un robot móvil llegar a su destino de manera segura sin importar condiciones estáticas o dinámicas de su entorno.

Sin embargo, algunos robots móviles, como el caso del iRobot Create, necesitan un controlador externo para dotarlos con autonomía. Generalmente, se recurre a la utilización de una PC para ello, la misma que se comunica con el robot a través de un cable. Al emplear un cable, el área de trabajo del robot móvil está limitada por la longitud de éste.

Como solución a lo anterior, el uso de módulos de comunicación inalámbrica resultan ser un buen recurso. En este caso particular se optó por seleccionar al estándar Zigbee como un camino alternante al que brinda Bluetooth.

Zigbee se establece como un estándar para aplicaciones en redes inalámbricas que requieran comunicaciones seguras, fiables, con baja tasa de envío de datos y maximización de la vida útil de las baterías de los dispositivos que conformen a este tipo de redes. El estándar Zigbee define principalmente a las capas de red y aplicación sobre las especificaciones del estándar IEEE 802.15.4 completando de esta manera al modelo OSI [2].

II. TRABAJOS RELACIONADOS

El desarrollo de algoritmos de navegación y el empleo de dispositivos que trabajan con el estándar ZigBee es poco conocido en nuestro país y también en desarrollo de aplicaciones como proyectos de grado universitario. Los proyectos relaciones nutren de una gran perspectiva del proyecto desarrollado.

En [3] y [4] se desarrollo la técnicas de localización utilizando un robot virtual mente ciego, el cual solo cuenta con sensores de contacto. En [5] implementan prototipos para limpieza y servicio en el hogar.

Se desarrolló el estudio de una red tipo Mesh para el control y cierre de cajas en un casino [6].

Finalmente, [7] se desarrolla el estudio de ZigBee en una aplicación de iluminación.

III. FUNDAMENTO TEÓRICO

El desarrollo de este trabajo tuvo dos directrices que conforman su fundamento teórico. A continuación se describen a cada una de ellas.

A. Navegación

La navegación en un robot es el proceso de planificar una trayectoria libre de obstáculos posible y guiarlo a través de ella para desplazarse desde una posición a otra en un área de trabajo.

Surgen entonces tres problemas por solucionar en relación a la navegación. El primero está relacionado con la auto-localización del robot, en donde, el robot recurre al conocimiento previamente adquirido y a sus observaciones locales, por medio de encoders acoplados a sus llantas, y externas, empleando sistemas de posicionamiento, para determinar su posición con relación a su entorno. La segunda problemática consiste en la identificación de la posición del objetivo con relación al entorno. Finalmente, el tercer inconveniente corresponde a la planificación de la trayectoria que permitirá establecer un camino posible desde la posición inicial a la posición del objetivo o meta esquivando obstáculos.

Un robot puede navegar en dos tipos de ambientes. Los entornos estructurados que permanecen estáticos y no sufren cambios con el transcurso del tiempo, y los entornos no estructurados que son dinámicos y pueden presentar cambios inesperados en los elementos que lo conforman. Con lo anterior, dentro de ambientes desconocidos o no estructurados, el robot deberá desarrollar capacidad para reaccionar ante la presencia de imprevistos en su trayectoria.

1) Algoritmos de navegación

En relación a la tarea de navegación se pueden considerar a dos clases de algoritmos que se enlistan a continuación:

- **Algoritmos fuera línea.** Para emplear un algoritmo fuera de línea se requiere una amplia información acerca del entorno, en la cual, se detalle con exactitud la localización de los obstáculos. El planeamiento de la trayectoria, en este caso, se realiza una sola vez y antes de que el robot móvil comience a desplazarse. De ahí su nombre de algoritmos fuera de línea. El problema en este tipo de algoritmos recae en contar con un proceso computacional eficiente, y no en garantizar que una trayectoria posible exista. Además, a los algoritmos fuera de línea se les pueden añadir criterios de optimización que den como resultado trayectorias en las cuales se tengan: un menor tiempo en atravesar un entorno, trayectorias que representen el camino más corto hacia la meta o simplemente trayectorias que sean las más seguras para el robot móvil.
- **Algoritmos fuera línea.** Al contrario de los anteriores algoritmos, la información del entorno que tienen los

algoritmos en línea es incompleta, por lo cual, la planeación de una trayectoria en estos métodos es un proceso continuo que para generar las acciones de control se recurre a la información del entorno entregada por los sensores del robot móvil. Al utilizar los sensores se retiran los requerimientos de conocer la localización y forma de los obstáculos, e inclusive de que éstos sean estacionarios.

La principal diferencia entre estas dos categorías de algoritmos radica en la cantidad de información que necesitan para su implementación. Tanto los algoritmos fuera de línea como los algoritmos en línea deben alcanzar la meta o determinar que el objetivo es inalcanzable tras el transcurso de un tiempo finito.

2) Propiedades del robot para la navegación

Las propiedades del robot relacionadas con tareas de navegación son las siguientes:

- **Configuración.** La configuración de un robot, denotada por q , es una completa especificación de la posición actual de dicho robot.
- **Espacio de configuración.** Corresponde al espacio C formado por todas las configuraciones que pueden ser alcanzadas por el robot R dentro del entorno de trabajo, y su dimensión es equivalente a los grados de libertad del robot. La dimensión del espacio de configuración es no Euclideano. Además, existe un subconjunto de C que determinan el espacio ocupado por el robot R , que esta denotado por $R(q)$ cuando el robot R se encuentra en q . Se define entonces a $R(q)$ por (1):

$$R(q) = \{q_i \in C \mid \|q, q_i\| \leq \rho\} \quad (1)$$
 Donde, ρ es el radio del robot R
- **Grados de libertad.** Los grados de libertad especifican el número de movimientos independientes que puede hacer el robot y determinan al mínimo número de variables que se necesita para representar a una configuración del robot.

B. Estándar Zigbee

Zigbee utiliza como base el estándar IEEE 802.15.4. Las características más relevantes que se encuentra en IEEE 802.15.4 son:

- Puede operar en tres bandas de frecuencias: 868 Mhz en Europa, 915 Mhz en los Estados Unidos y en la banda libre ISM¹ de 2.4 Ghz a nivel mundial.
- Tasa de transmisión de datos de 250 Kbps a 2.4 GHz, de 20 Kbps a 868 MHz y de 40 Kbps a 915 MHz.
- El acceso al medio lo hace por medio del algoritmo CSMA-CA (*Carrier Sense Multiple Access with Collision Avoidance*).
- Está orientado hacia un bajo consumo de energía.

¹ ISM (Industrial, Scientific and Medical): son bandas reservadas internacionalmente para uso no comercial de radiofrecuencia electromagnética en áreas industrial, científica y médica. En la actualidad estas bandas han sido popularizadas por su uso en comunicaciones WLAN (e.g. Wi-Fi) o WPAN (e.g. Bluetooth).

- Emplea direcciones físicas de 64 bits IEEE y direcciones lógicas o de red de 16 bits.
- Proporciona un indicador de la calidad del enlace, LQI (*Link Quality Indicator*).
- Cuenta con un sistema que detecta la energía, ED (*Energy Detector*).
- La potencia de transmisión es de al menos 1 mW con un alcance mínimo de 10 m.

1) Tipos de dispositivos

En ZigBee se definen básicamente a 3 tipos de elementos:

- **Coordinador.** Es el nodo encargado de medir la energía en cada uno de los canales escogiendo el más silencioso y coloca un identificador de red de área personal (*Personal Area Network ID, PAN ID*) para establecer la red ZigBee. Una vez establecido estos parámetros, el coordinador puede formar una red, permitiendo unirse a él, routers y dispositivos finales. Ya formada la red, el coordinador hace las funciones de Router en el enrutamiento de paquetes y origen.
- **Router.** Es el encargado del enrutamiento de tramas hacia otros dispositivos de la red. Además, puede actuar como coordinador de un pequeño grupo de estos y capaz de almacenar los mensajes para ser enviados posteriormente.
- **Dispositivo final.** Los dispositivos finales tienen una funcionalidad reducida, debido a que no realizan tareas de enrutamiento. Por esta razón, la interacción con otros dispositivos de red se lo realiza a través del coordinador o de un router. Normalmente estos nodos tienen una configuración de bajo consumo de energía (Modo Sleep) para lograr una mayor duración de la batería al momento que interactúan dentro de la red.

2) Comparación con otras tecnologías

Resulta importante realizar una comparación entre Zigbee con los estándares Wi-fi y Bluetooth, que funcionan en la misma banda de frecuencia de 2.4 GHz y han alcanzado una gran acogida a nivel mundial.

La principal diferencia entre estas tecnologías radica en la cantidad de corriente que necesitan para operar, en donde, se evidencia un bajo consumo de energía por parte del estándar Zigbee. Esta característica permite que las baterías de los dispositivos en redes Zigbee duren un mayor periodo de tiempo.

Sin embargo, la tasa de transmisión de datos en Zigbee es la menor comparada con la que maneja Bluetooth y Wi-fi, lo que limita a esta tecnología a ser utilizada únicamente en redes LR-WPAN (*Low Rate – Wireless Personal Area Network*). A pesar del gran consumo de energía por parte del estándar Wi-fi hay que resaltar la capacidad de transmitir gran cantidad de datos.

IV. MATERIALES Y MÉTODOS

A. iRobot Create

El *iRobot Create* es una plataforma de desarrollo enfocada al ámbito educacional o aficionados por la Robótica. Este robot móvil, creado por la empresa iRobot, se programa en base a un conjunto de comandos, mediante los cuales, se pueden desarrollar nuevos comportamientos o inclusive emular a un instrumento musical. Además de ello, se puede añadir electrónica a su estructura, como sensores, indicadores, un brazo robótico, etc., utilizando sus conectores físicos. En la Fig. 1 se puede visualizar a esta plataforma y a algunas de sus partes.

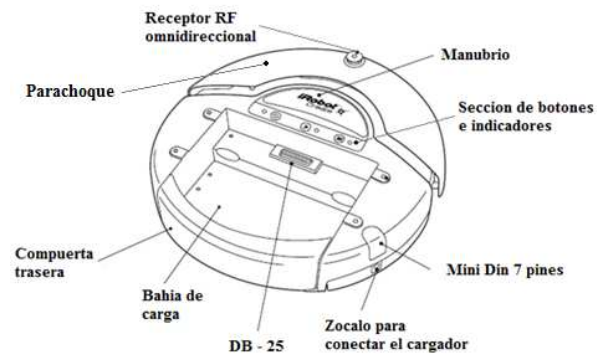


Fig. 1. Vista superior del *iRobot Create*.

El robot cuenta con un cable serial de programación, mediante el cual, se establece su control desde una PC.

B. Módulos RF XBee

Los módulos utilizados en este proyecto son:

- XBee (Series 2).
- XBee – PRO (Series 2).

Estos dispositivos son módulos de radio frecuencia diseñados para operar dentro estándar IEEE 802.15.4 en la banda 2.4GHz ISM. Los módulos proporcionan una comunicación inalámbrica de datos fiable y satisfacen las necesidades de bajo coste y reducido consumo de las aplicaciones inalámbricas. Además, son fáciles de usar y poner en marcha de manera inmediata, ya que poseen el stack correspondiente pre-cargado. En la Fig. 2. se visualizan a los módulos empleados.

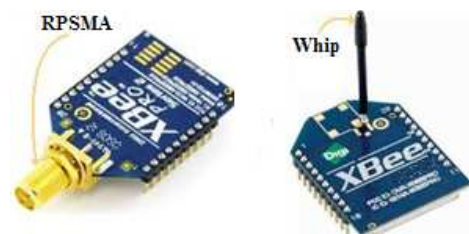


Fig. 2. Módulos XBee

Los módulos pueden ser configurados a través de comandos AT; y, operar de manera transparente ó *API* (*Application Programming Interface*). Cuando se los destina para operar de forma transparente actúan como una extensión

inalámbrica del puerto serial. Por otro lado, cuando su manera de operar es API, toda la información, tanto entrante como saliente está contenida en tramas que definen eventos u operaciones en el módulo.

C. Comunicación inalámbrica entre los módulos Xbee

Para montar la comunicación inalámbrica se configuró a los módulos de tal manera que operen de manera transparente. En la Tabla 1. se sintetiza la configuración de cada uno de los módulos

TABLA 1
Parámetros para la configuración de los módulos Xbee

Parámetro	XBee	Xbee - PRO
Velocidad de transmisión puerto serial	57600 Kpbs	57600 Kpbs
Bits de datos	8	8
Bits de paridad	0	0
Bits de parada	1	1
Direccionamiento	Físico	Físico
Firmware	Zigbee Router AT	Zigbee Coordinator AT

D. Algoritmos de navegación

Para seleccionar los algoritmos de navegación dentro de este proyecto se tomaron las siguientes consideraciones:

- El escenario que se planteó es de carácter desconocido, en donde, la información inicial del entorno con la que se le proveyó al robot fue únicamente su posición inicial y la ubicación de la meta. Los obstáculos en este caso se ubicaron de forma dinámica dentro del entorno.
- Los sensores que fueron considerados para que el *iRobot Create* se desempeñe en tareas de navegación fueron los de rotación y de impacto. El primer grupo de sensores se encarga de efectuar las mediciones en desplazamiento y en rotación del robot, mientras que los sensores del segundo grupo son utilizados para localizar a los obstáculos a lo largo del trayecto desde el origen hacia la meta. Los sensores de impacto del *iRobot Create* detectan a un obstáculo solo después de haber impactado contra éste. De lo anterior se desprende que el robot sustenta la implementación de algoritmos básicos para la navegación porque entre más complejo es el algoritmo mayor es la demanda en los requerimientos de los sensores.

Por lo anterior, se escogió a los algoritmos Gusano 1 y 2 para ser implementados en el *iRobot Create* porque requieren de un robot móvil con sensores de impacto para su funcionamiento y son empleados en ambientes desconocidos y además, le brindarán al robot la capacidad de alcanzar un objetivo mediante el planeamiento de trayectoria basado en la información que entreguen sus sensores.

Los algoritmos Gusano consideran al robot móvil como un punto con una perfecta localización y para detectar obstáculos utilizan sus sensores de impacto. Su funcionamiento está formado por dos comportamientos principalmente:

- Movimiento sobre una línea recta para alcanzar la meta; y,
- Seguimiento del contorno del obstáculo.

El éxito de estos algoritmos radica en usar los dos comportamientos anteriores para generar un camino libre de colisiones que permita alcanzar la meta. Existen tres clases de algoritmos Gusano que se basa en sensores de impacto, sin embargo, solo se toman en cuenta a dos de ellos porque son los que mayor difusión han tenido. A continuación se detallan a los algoritmos Gusano implementados.

1) Gusano 1

El robot parte del conocimiento la partida q_{inicio} , que constituye la primera configuración de salida; y, su llegada q_{meta} , que es la ubicación del objetivo que se desea alcanzar.

Las configuraciones de salida son etiquetadas como q_{i-1}^S , donde $i = 1, 2, 3 \dots$. Para el caso de la primera configuración de salida se tiene que $i = 1$, entonces $q_0^S = q_{inicio}$. Se establece la línea m , que une a q_{i-1}^S con la localización de la llegada q_{meta} , sobre la cual, el robot deberá desplazarse utilizando el comportamiento de moverse hacia un objetivo hasta alcanzar la meta o encontrar a un obstáculo. Si alguno de los sensores de contacto informa sobre un obstáculo, al primer punto de choque con el obstáculo se lo nombra como punto de impacto q_i^I . Entonces, el robot emplea el comportamiento de rodear obstáculos hasta alcanzar la meta o retornar a q_i^I . Después de ello, se determina el punto sobre el perímetro del obstáculo más cercano a la meta, conocido como punto de salida q_i^S . El robot debe dirigirse hasta q_i^S , incrementar en 1 el valor de i e invocar nuevamente al comportamiento de moverse hacia un objetivo. Para los casos en los cuales el robot llega a la meta se declara meta alcanzada. Sin embargo, si la línea que une a un punto de salida q_i^S con el punto de llegada q_{meta} se encuentra el obstáculo con el que se ha chocado anteriormente, el objetivo se declara inalcanzable. En la fig. 3 se muestra un ejemplo del desempeño del Algoritmo Gusano 1.

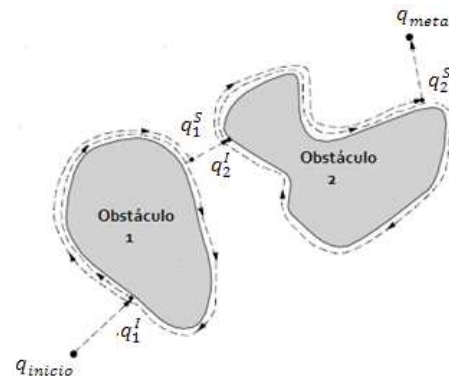


Fig. 3. Ejemplo del Algoritmo Gusano 1

2) Gusano 2

Durante el movimiento a la meta, el robot se mueve hacia el punto de llegada q_{meta} sobre una trayectoria rectilínea desde el punto de partida q_{inicio} , pero esta línea m que une a estos dos puntos permanece fija durante toda la ejecución del algoritmo Gusano 2. El comportamiento de seguimiento de borde es invocado si el robot encuentra un obstáculo, pero dicho comportamiento difiere del Gusano 1, porque el robot circunnavega el obstáculo hasta que encuentre un nuevo punto x que cumpla con la condición de ser el más cercano a la línea m . En este momento el robot nuevamente se llama al comportamiento de moverse hacia el objetivo, repitiendo este proceso si encuentra a otro obstáculo durante su trayecto. La Figura 4.3 indica un ejemplo del algoritmo Gusano 2.

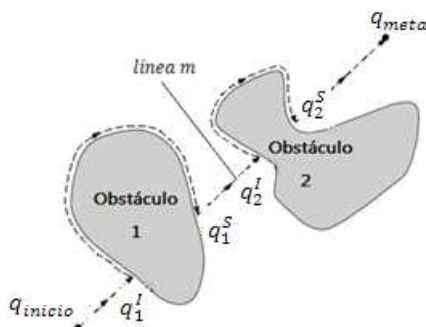


Fig. 4. Ejemplo del Algoritmo Gusano 2

V. IMPLEMENTACIÓN Y PRUEBAS

A. Comunicación Inalámbrica

Para la implementación de la comunicación inalámbrica entre los módulos Xbee se tomaron las siguientes consideraciones:

- **Requerimiento del proyecto.** Se necesita una comunicación inalámbrica fiable que brinde un PER (Packet Error Rate) del 0%. Con esto se garantiza que no existan paquetes perdidos en la recepción.
- **Carga de datos enviada.** Se establece que el periodo de envío de datos sea de 50ms y la máxima carga de datos a ser enviada corresponda a 30 bytes. Estos valores se establecen por la consulta del estado de los sensores al robot.

La Fig. 5. indica la arquitectura que se obtuvo después de integrar estas dos tecnologías.

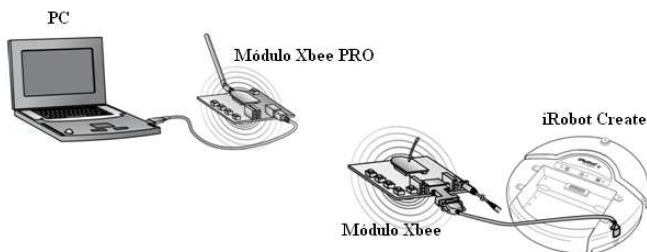


Fig. 1. Comunicación inalámbrica entre la PC y el iRobot Create.

B. Algoritmos Gusano

Para la implementación de los algoritmos Gusano, se requiere determinar las propiedades relacionadas a la tarea de navegación del iRobot Create. Éstas son descritas a continuación:

- **Grados de libertad.** La plataforma robótica móvil tiene tres grados de libertad definidos por sus capacidades: de traslación en línea recta o en arco a lo largo de un plano, y de rotación sobre su propio eje.
- **Configuración.** La información que contiene la configuración q se describe en (2) y se distribuye de la siguiente manera:
 - Dos variables x e y , que informan sobre la posición del robot en un plano.
 - Una variable θ , que detalla la orientación del robot.

$$q = (x, y, \theta) \quad (2)$$

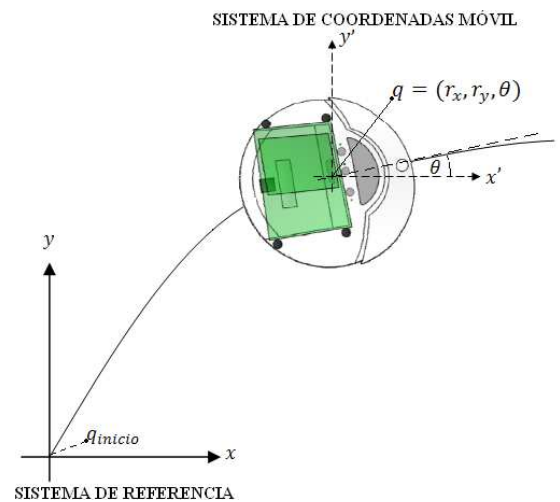


Fig. 6. Sistemas de coordenadas empleados para calcular las configuraciones del robot.

Para calcular las configuraciones alcanzadas por el robot se considera un sistema de referencia cuyo origen coincide con la ubicación de partida conocida como q_{inicio} , y un sistema de coordenadas móvil. Al robot se le considera como un punto, sobre el cual descansa el origen de dicho sistema de coordenadas móvil. La configuración calculada corresponde a la relación existente entre ambos sistemas de coordenadas, empleando vectores de posición y de desplazamiento para encontrar la posición del robot; y, Geometría Euclidiana para hallar la orientación. La Fig. 6. indica los sistemas de coordenadas utilizados.

- **Espacio de configuración.** El espacio de configuración corresponde a todas las configuraciones que puede alcanzar el robot dentro de un plano que tiene 2.0 m de ancho por 1.8 m de largo. En la Fig. 7 se muestra el espacio de trabajo definido.

Como el radio del robot es de 16.75 cm, el subconjunto C se define en (3).

$$R(q) = \{q_i \in C \parallel \|q, q_i\| \leq 16.75\} \quad (3)$$

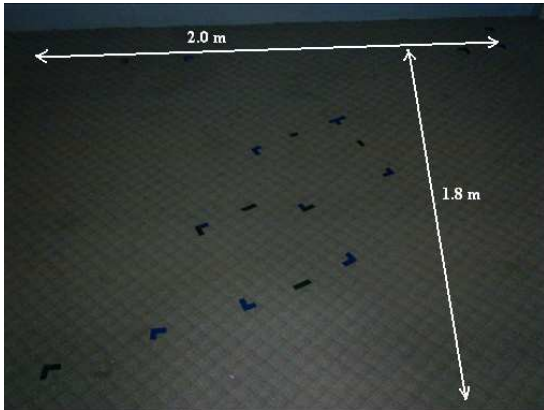


Fig. 7. Espacio de trabajo para la implementación de los algoritmos Gusano.

La implementación de los algoritmos se fundamenta básicamente en el desarrollo de dos rutinas que se describen a continuación:

- **Desplazamiento en dirección a la meta.** Esta rutina establece un vector que está definido desde la configuración de salida q_{i-1}^S hacia la configuración de llegada q_{meta} . Para que al robot se le permita moverse a lo largo de la línea de acción del vector, se calcula su dirección con respecto al sistema de coordenadas asociado a la configuración de salida y su módulo que es la distancia que separa a q_{i-1}^S de q_{meta} . Determinada la dirección, el robot rota sobre su propio eje, para después trasladarse de forma rectilínea el valor de la magnitud del vector. El desplazamiento hacia la meta termina cuando el robot ha alcanzado a q_{meta} , si un obstáculo ha sido encontrado durante el trayecto, ó si la meta es declarada inalcanzable.
- **Seguimiento del contorno del obstáculo.** La terminación de esta rutina depende del tipo de algoritmo Gusano. Es la encargada de indicarle al robot cómo bordear a un obstáculo después haberlo impactado para después planificar la trayectoria que le permita alcanzar la meta nuevamente. El bordeado de un objeto basa su funcionamiento en la información entregada por los sensores de impacto y se lo hace siempre en sentido horario. Para ejecutar esta rutina es necesario determinar la configuración del primer punto de impacto q_i^I , después de lo cual, el robot realiza lo siguiente:
 - a. **Retroceder en la misma dirección.** El robot recorre en sentido contrario una determinada distancia pero manteniendo la dirección con la cual impacto al obstáculo. Esto le permite al robot alejarse del obstáculo.
 - b. **Rotación sobre su propio eje.** Se le ordena al robot girar 45° en sentido anti horario con el propósito de no alcanzar un punto de impacto de manera consecutiva.
 - c. **Traslación en arco de circunferencia.** El robot procede a desplazarse sobre un arco de circunferencia que tiene un radio constante. Esta

acción está encaminada a alcanzar nuevamente al obstáculo definiendo un punto de choque diferente al anterior.

Para la programación de los algoritmos se optó por utilizar como herramienta de programación a MatLab que se caracteriza por manejar un lenguaje de alto nivel garantizando cálculos exactos en un corto periodo de tiempo y permitiendo la creación de una interfaz grafica.

VI. ESCENARIOS Y PRUEBAS

A. Comunicación inalámbrica

El escenario de pruebas que se consideró fue un entorno interno. Para esta prueba se tomó las siguientes consideraciones:

- **Área de trabajo de pruebas.** Para determinar el rendimiento de los módulos se define una separación máxima entre módulos de 30m dada por sus especificaciones en ambientes internos. En la Fig. 8. Se esquematizó anterior.

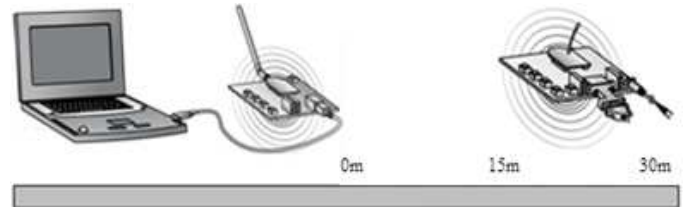


Fig. 8. Prueba de comunicación.

- **Carga de datos enviada.** Se estableció que el periodo de envío de datos sea de 50ms y la máxima carga de datos a ser enviada corresponda a 30 bytes. Estos valores se definieron por la consulta del estado de los sensores al robot.

B. Algoritmos de Gusano

Para efectuar las pruebas se han definido tres escenarios que están en función del tipo de obstáculos mostrados en la Fig.7., Fig. 8., y Fig.9. En cada escenario de pruebas se asume que el robot parte de posición (0 ; 0) m con respecto al sistema de referencia y se especifica las coordenadas de la meta que se desea alcanzar.

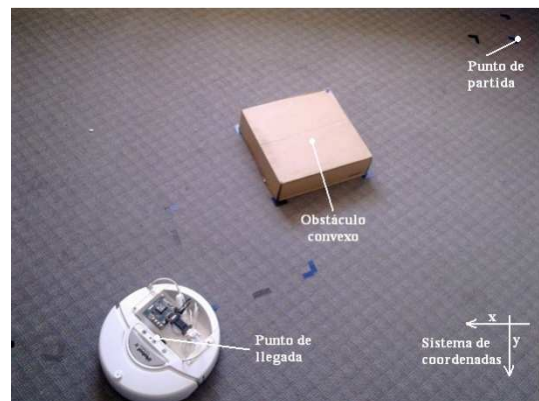


Fig 7. Escenario 1: contiene un a obstáculo convexo.

VII. RESULTADOS

A. Comunicación inalámbrica

A partir de una separación en distancia de 20m existieron pérdidas en la recepción de paquetes independiente de la carga de datos que fue transmitida. Entonces, se considera que el alcance máximo de la red, en la cual se garantiza que 100% de recepción de paquetes recibidos con una intensidad promedio de señal recibida de -64dBm, corresponde a un valor de 15m con una carga máxima de datos de 30 bytes.

El promedio de la intensidad de señal recibida se encuentra dentro del rango de -39dBm a -96dBm, parámetros que especifica el fabricante de las tarjetas.

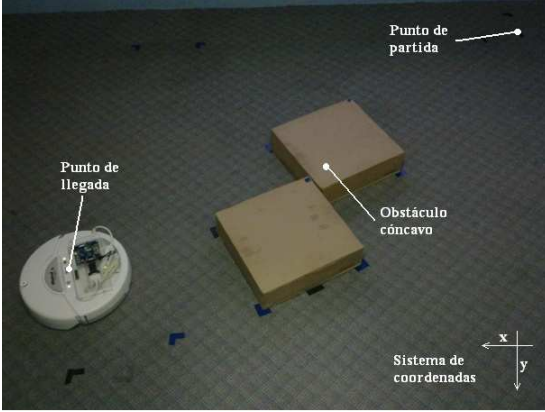


Fig 8. Escenario 1: contiene un obstáculo cóncavo.

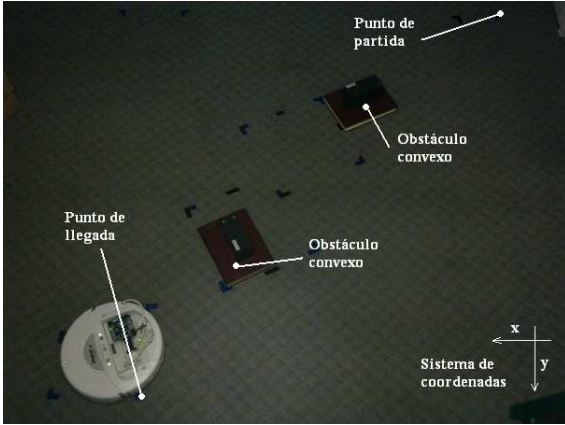


Fig 9. Escenario 3: contiene a dos obstáculos convexos.

Además de detalló un criterio que sirvió para la recolección de los resultados obtenidos después del desempeño de los algoritmos Gusano. Para ello, se utilizó la definición del espacio ocupado por el robot para determinar un valor porcentual que indique la certeza con la cual el robot alcanza la meta en un determinado número de pruebas. Este proceso consistió en conocer cuan cercano ha quedado el robot de la meta utilizando (3). El momento que se considera para efectuar la medición es cuando la meta se declara alcanzada en el programa implementado sobre el robot. La medición a efectuarse es la distancia d existente entre el centro del robot q y q_{meta} . Existen tres posibilidades cuando esto ocurra:

a. **El robot ha alcanzado la meta.** Este caso corresponde al ideal y no es necesario realizar ninguna medición porque su valor porcentual se considera como 100%. Esta dado por (4):

$$d \leq \rho \quad (4)$$

b. **El robot se encuentra cercano a la meta.** Si pasaría esto el valor porcentual sería menor al 100% pero mayor al 0%. Este caso dependerá de que se cumpla (5):

$$\rho < d \leq 2\rho \quad (5)$$

c. **El robot se encuentra alejado de la meta.** A esta posibilidad se le considera la peor de todas. Se determina que el robot esta distante de la meta y su valor porcentual corresponde al de 0%. En este caso se da por (6):

$$d > 2\rho \quad (6)$$

En la Fig. 10. se ilustran los tres casos que pueden pasar.

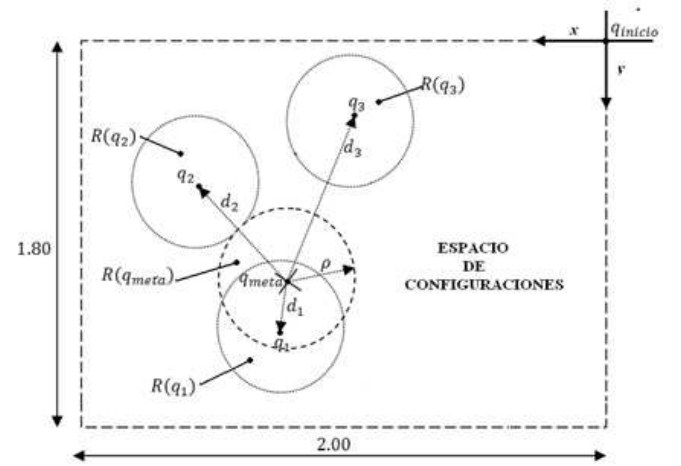


Fig. 10. Consideraciones para determinar el valor porcentual de alcanzar la meta, en donde, para $d_1 \leq \rho$ garantiza que el robot ha alcanzado la meta, para $\rho < d_2 \leq 2\rho$ entrega un valor porcentual indicando la cercanía del robot a la meta; y, para $d_3 > 2\rho$ contradice totalmente a la declaración de meta alcanzada porque el robot se encuentra alejado del punto de llegada.

B. Algoritmos Gusano

En la Figura. 5.8 se visualizan tres diagramas de barra que sintetizan la certeza de alcanzar la meta en cada algoritmo para los diferentes escenarios propuestos. En base a esta figura, se puede evidenciar que al utilizar el algoritmo Gusano 2 se obtuvo una mayor certeza de llegar a la meta para obstáculos convexos comparado con el Gusano 1.

El Gusano 1 sin importar el tipo de obstáculo mantiene una certeza parecida cuando sólo se presenta un obstáculo en su trayecto. Su caso crítico resulta cuando se enfrenta a más de un obstáculo, pues la seguridad de alcanzar la meta en este tipo de escenarios es muy baja como resultado del error de posición.

Otro aspecto a consideración es la distancia recorrida por el robot y el tiempo que demanda cada algoritmo para su ejecución en cada uno de los escenarios propuestos. En todos los escenarios de pruebas, el Gusano 2 le permite al robot recorrer una menor distancia hacia la meta cuando existen obstáculos en su trayecto comparado con lo que hace el robot bajo el Gusano 1. Además de ello, el tiempo que emplea el Gusano 2 para su culminación es inferior al que requiere el Gusano 1.

VIII. CONCLUSIONES

Al finalizar este trabajo, se demostró que los algoritmos implementados sobre la plataforma del *iRobot Create* le permitieron realizar la tarea de navegación en entornos no estructurados y contando únicamente con la información de su posición de origen o salida y la ubicación de la meta o punto de llegada.

Para los escenarios de pruebas propuestos, se comprobó que la certeza de llegar a la meta propuesta en el Gusano 1 depende del número de obstáculos que atraviesan su trayectoria hacia el punto de llegada. Bajo este algoritmo, el robot para más de un obstáculo presenta problemas en alcanzar la meta debido al error de posición que induce el bordeado completo de los obstáculos. Mientras que, la certeza conseguida con el Gusano 2 depende de cuan cercano es el punto determinado para salir del obstáculo con respecto la línea fija que une al punto de partida y llegada. Dentro del algoritmo Gusano 2, el número de obstáculos existentes entre el punto de inicio y de meta no afecta a la certeza de alcanzar la meta de manera considerable.

Bajo el análisis de resultado se demostró que los dispositivos XBee y XBee – PRO Series 2 actúan como una extensión wireless del puerto serie, es decir todos los datos enviados al puerto serie son transmitidos vía RF. Cuando los datos son recibidos por RF, estos son enviados al puerto serie.

IX. FUTUROS TRABAJOS

Para trabajos futuros se recomienda implementar un método que disminuya el error de posición.

También, dotar al robot de un mapa a más de la capacidad de desempeñarse en ambientes desconocidos, para que, sea capaz de ejecutar tareas de vigilancia de una determinada área, ó desempeñarse como guía de museo o bibliotecas.

Por último, añadir otro tipo de sensores a su electrónica que permita implementar algoritmos de navegación más sofisticados.

X. REFERENCIAS

- [1] S. M. LaValle, *Planning Algorithms*. Reino Unido, Universidad de Cambridge, 2006, también disponible en <http://msl.cs.uiuc.edu/planning/>.
- [2] The Institute of Electrical and Electronics Engineers, Inc., *IEEE Standard 802.15.4 - IEEE Standard for Information technology Telecommunications and information exchange between systems - Local and Metropolitan area networks Specific requirements*, October, 2003.
- [3] Lawrence H. Erickson, Joseph Knuth, Jason M. O’Kane, Steven M. LaValle, *Probabilistic localization with a blind robo*. In *Proceedings IEEE International Conference on Robotics and Automation*, 2008.
- [4] Ilari Vallivaara, Janne Haverinen, Anssi Kemppainen, Juha Rönning, *Simultaneous Localization and Mapping Using Ambient Magnetic Field*. 2010.
- [5] Max Kuipers *Localization with the iRobot Create, 2009*.
- [6] Roberto P. Baca, Sanny R. Checha, *Diseño y construcción de una red inalámbrica para cierre de caja de mesas de juego del Best Western Hotel Casino Plaza, utilizando tecnología ZigBee, Febrero 2010*.
- [7] O. Hyncica, P. Kacz, P. Fiedler, Z. Bradac, P. Kucera and R. Vrba, *The ZigBee Experience*.