

Diseño de robots móviles para trabajo colaborativo con aplicaciones a la agricultura de precisión.

Mobile robots design for collaborative work with applications for precision agriculture.

Wendy Yesenia Govea Rodríguez (1).
Programa Verano Delfín.
wgovea@ucol.mx

Oscar Alejandro Lemus Pichardo (2). Programa Verano Delfín, oscarl.ocho@gmail.com.

Alejandro Medina Santiago* (3). Universidad de ciencia y tecnología Descartes, cidit@universidaddescartes.edu.mx.

Lorena del Rosario Rojas Nucamendi (4). Instituto de Ciencia, Tecnología e Innovación del Estado de Chiapas, lorenarojas@mexicoled.com.

*corresponding author.

Artículo recibido en marzo 25, 2019; aceptado en abril 11, 2019.

Resumen.

La programación de un robot autónomo de tres grados de libertad para la agricultura de precisión, el robot aprenderá a moverse dependiendo de la zona donde se encuentre, identificar algunos elementos de las plantas como color, humedad, etc. para determinar si la planta está viva o muerta, localizar su posición a través de un GPS para proceder a retirarla de un invernadero mediante otro robot especializado para dicha tarea. Las posibilidades que ofrecen los robots móviles son muy importantes en los campos militar, sanitario, de expediciones terrestres con fines académicos o civiles, en la inspección de instalaciones nucleares, y en general en todo tipo de entornos hostiles y de seguridad. Se presenta el desarrollo de un sistema que realiza el seguimiento de una trayectoria para hacer la ubicación de plantas dentro de un terreno, las cuales ya se encuentren muertas. La aplicación se ejecuta en un dispositivo móvil que cuenta con sistema operativo Android que se comunica con el robot a través de Bluetooth, ya que éste cuenta con una tarjeta de Arduino UNO en la que están conectados 4 sensores ultrasónicos y un sensor de humedad. Para facilitar el diseño, nuestro primer paso fue desarrollar diversas actividades con redes neuronales para determinar el mejor diseño de RNA-BP.

Palabras clave: robots móviles, agricultura de precisión, trabajo colaborativo, redes neuronales.

Abstract.

Programming an autonomous robot of three degrees of freedom for precision agriculture, the robot will learn to move depending on the area where it is; to identify some elements of plants such as color, humidity, etc. to determine if the plant is alive or dead; to locate its position through a GPS to remove it from a greenhouse using another specialized robot for this task. The possibilities offered by mobile robots are very important in the military and health fields; in terrestrial expeditions; for academic or civil purposes, in the inspection of nuclear facilities, and in general in all types of hostile and security environments. A system development that tracks a path to the location of dead plants in a field is presented. The application runs on a mobile device that has an Android operating system that communicates with the robot via Bluetooth, since it has an Arduino UNO card to which 4 ultrasonic sensors and a humidity sensor are

connected. To facilitate the design, our first step was to develop various activities with neural networks to determine the best design of RNA-BP.

Keywords: mobile robots, precision agriculture, collaborative work, neural networks.

1. Introducción.

La robótica móvil es una valiosa herramienta para explorar entornos inaccesibles al ser humano por su lejanía, coste o peligrosidad y para realizar tareas desagradables o laboriosas. Es un campo relativamente nuevo, hasta hace poco experimental, pero que ya se está aplicando a problemas reales con resultados satisfactorios. Una mayor autonomía del robot, es decir, una menor supervisión humana para su correcto funcionamiento es uno de los propósitos primordiales de esta área de conocimiento (Fernández, 2009).

La robótica cooperativa surge como una idea de sentido común para mejorar el desempeño en la realización de la tarea a ejecutar. Hacer un trabajo entre varios puede tener muchas más ventajas como, por ejemplo, reducir el tiempo de ejecución de la tarea o realizar tareas más complejas, que hacerlo uno solo. Basta con darse cuenta de que hay algunas tareas que simplemente no pueden ser realizadas únicamente por un individuo debido a que sobrepasa sus límites. Ante esta problemática, durante más de dos décadas, ha surgido un creciente interés por el estudio de sistemas que integran varios robots conocidos con el nombre de Sistemas Multi-Robot (SMR) o equipo de robots y que en algunas tareas han demostrado ventajas que los hacen más eficientes respecto a sistemas uni-robot. Los SMR ofrecen interesantes alternativas en la concepción de sistemas robóticos y pueden darnos muchos servicios útiles. Sin embargo, el diseño y la implementación de un equipo de robots enfrentan problemas concernientes a los recursos de cómputo, especificación de tareas, comunicación entre robots, localización, control y planificación entre muchos otros.

En el presente proyecto hemos diseñado e implementado un sistema cuyo propósito principal es hacer un recorrido del robot por un plantío para hacer la identificación de la flora muerta y colocar una nueva planta, esto es con ayuda de los sensores ultrasónicos, más adelante se explica detalladamente la realización de los diversos programas además de la aplicación móvil para poder observar en un mapa la ubicación de la planta.

2. Métodos.

De trabajos anteriores a este proyecto ya se cuenta con el primer robot móvil el cual realiza la detección de las plantas muertas, se desarrolla la etapa de control y localización vía GPS, como resultado, se espera que los datos del GPS se pasen al segundo robot móvil, éste se dirigirá a las coordenadas especificadas por el primero para hacer la obtención de la planta muerta y colocar una nueva en la misma posición además de hacer la revisión de la humedad para regarla.

2.1 Diseño de la arquitectura neuronal.

Por lo que se hizo una tabla de verdad que con respecto a cada uno de los sensores ultrasónicos se dirigirá el robot móvil a una dirección. El siguiente código de la figura 1, lo que realiza es un ciclo para determinar las diferentes condiciones posibles dependiendo de los resultados que se quieren.

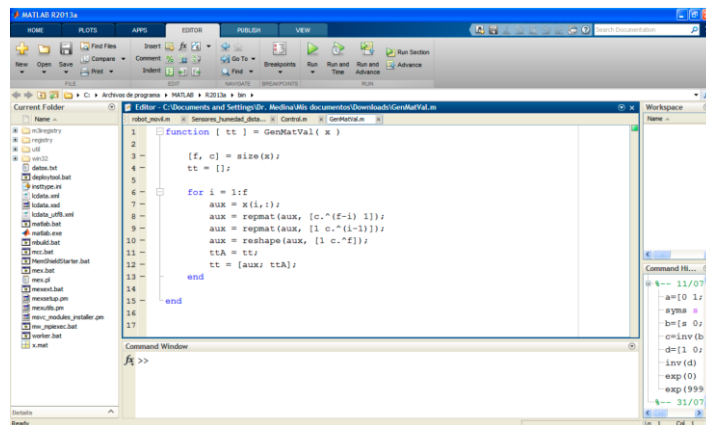


Figura 1. Código para las distintas combinaciones.

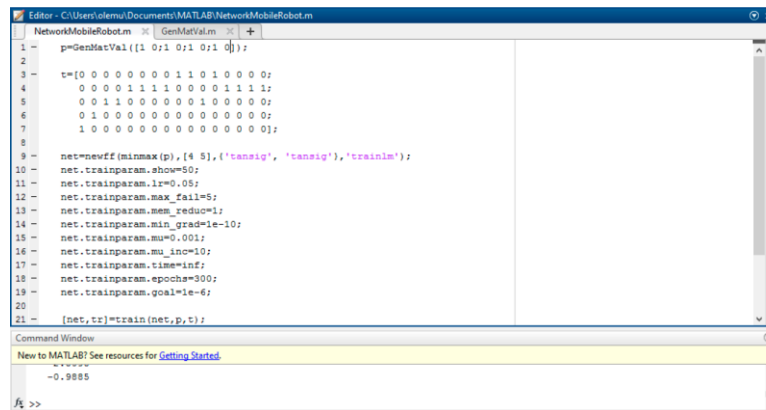
La tabla 1 muestra las acciones planteadas al robot dependiendo de los datos arrojados por los sensores de distancias montados en su chasis.

Tabla 1. Acciones del robot con respecto a los sensores de distancia.

SI	SA	SD	SR	Derecha	Adelante	Izquierda	Atrás	Stop
1	1	1	1	0	0	0	0	1
1	1	1	0	0	0	0	1	0
1	1	0	1	0	0	1	0	0
1	1	0	0	0	0	1	0	0
1	0	1	1	0	1	0	0	0
1	0	1	0	0	1	0	0	0
1	0	0	1	0	1	0	0	0
1	0	0	0	0	1	0	0	0
0	1	1	1	1	0	0	0	0
0	1	1	0	1	0	0	0	0
0	1	0	1	0	0	1	0	0
0	1	0	0	1	0	0	0	0
0	0	1	1	0	1	0	0	0
0	0	1	0	0	1	0	0	0
0	0	0	1	0	1	0	0	0
0	0	0	0	0	1	0	0	0

Los valores SI, SA, SD y SR corresponden a cada sensor ultrasónico montado (izquierda, derecha, adelante, atrás), los siguientes datos de la tabla corresponden a la acción que tomará el carro.

En la figura 2 se tiene el código para obtener las bias y los pesos de cada una de las neuronas, se hicieron diversas pruebas cambiando las capas y la cantidad de neuronas en cada una de ellas hasta encontrar la mejor, resultando de dos capas teniendo en la primera 4 neuronas y en la segunda 5 que nos daba la matriz final idéntica a la deseada. A la red neuronal se le aplicó un entrenamiento backpropagation con funciones de activación “tansig” para las capas ocultas y la de salida (Hagan, 1996).



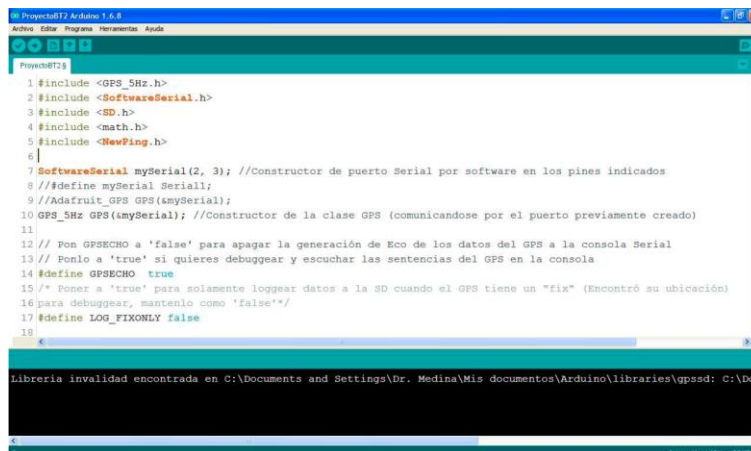
```

Editor - C:\Users\olemu\Documents\MATLAB\Network\MobileRobot.m
NetworkMobileRobot.m  GenMatVal.m  +
1 - p=GenMatVal([1 0:1 0:1 0:1 0]);
2
3 -
4 - t=[0 0 0 0 0 0 0 0 1 1 0 1 0 0 0 0];
5 - 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1;
6 - 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0;
7 - 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
8
9 - net=newff(minmax(p),[4 5],{'tansig','tansig','trainlm'});
10 - net.trainparam.show=50;
11 - net.trainparam.lr=0.05;
12 - net.trainparam.max_fail=5;
13 - net.trainparam.mem_reduc=1;
14 - net.trainparam.min_grad=1e-10;
15 - net.trainparam.mu=0.001;
16 - net.trainparam.mu_inc=10;
17 - net.trainparam.time=inf;
18 - net.trainparam.epochs=300;
19 - net.trainparam.goal=1e-6;
20
21 - [net,t]=train(net,p,t);
Command Window
New to MATLAB? See resources for Getting Started
-0.9885
f >>

```

Figura 2. Código para la obtención de pesos sinápticos y bias de la RNA-BP.

Al obtener los resultados se lleva estos datos a la programación del sistema embebido Arduino, en la que se monitoreaban los sensores y se condicionaban para que activaran los motores e hicieran lo que ya se había indicado mediante programación.



```

Project071 Arduino 1.6.8
Archivo Editor Programa Herramientas Ayuda
Project071
1 #include <GPS_SHz.h>
2 #include <SoftwareSerial.h>
3 #include <SD.h>
4 #include <math.h>
5 #include <NewPing.h>
6
7 SoftwareSerial mySerial(2, 3); //Constructor de puerto Serial por software en los pines indicados
8 //define mySerial Serial1;
9 //Adafruit_GPS GPS(mySerial);
10 GPS_SHz GPS(mySerial); //Constructor de la clase GPS (comunicandose por el puerto previamente creado)
11
12 // Pon GPSECHO a 'false' para apagar la generación de Eco de los datos del GPS a la consola Serial
13 // Ponlo a 'true' si quieres debuggear y escuchar las sentencias del GPS en la consola
14 #define GPSECHO true
15 /* Poner a 'true' para solamente loggear datos a la SD cuando el GPS tiene un "fix" (Encontró su ubicación)
16 para debuggear, manténlo como 'false'*/
17 #define LOG_FIXONLY false
18
Librería invalida encontrada en C:\Documents and Settings\Dr. Medina\Mis documentos\Arduino\libraries\gpsd: C:\Doc

```

Figura 3. Código del sistema embebido para la obtención de datos de los sensores.

Con el código de la figura 3 se programa para que el sensor ultrasónico activara un pin de salida el cual va directo a los motores para indicar que algún objeto se encuentra enfrente, esto es para dirigir al robot móvil, orientarlo durante su recorrido por el plantío, mientras tanto el sensor de humedad solo activa la opción de guardar las coordenadas en la memoria micro SD si se detecta humedad, esto es para que el robot vaya directamente a los puntos señalados y regar las plantas que lo necesitan.

Haciendo distintas pruebas, además de hacer mucha investigación se comprendió el código NMEA como el que se muestra a continuación:

```
$ GPGGA, 123519,4807.038, N, 01.131,000, E, 1,08,0.9,545.4, M, 46,9, M,, * 47
```

Dónde:

Datos Fix GGA Sistema de Posicionamiento Global

123519 Fix tomada en 12:35:19 GMT

4807.038, latitud N 48 ° 07,038 'N

01131.000, E Longitud 11 ° 31.000 'E

1 Fijar la calidad: 0 = no válido

1 = punto de GPS (SPS)
 2 = fijo DGPS
 fix 3 = PPS
 4 = Tiempo real cinemática
 5 = RTK flotante
 6 = estimado (estima) (2.3 función)
 7 = modo de entrada manual
 8 = Modo de simulación
 08 Número de satélites rastreados
 0,9 dilución horizontal de la posición
 545,4, M Altitud, Metros, sobre el nivel del mar
 46,9, H Altura del geoide (nivel medio del mar) por encima de WGS84
 (Campo vacío) el tiempo en segundos desde la última actualización de DGPS
 (Campo vacío) Número de identificación de la estación DGPS
 * 47 los datos de suma de comprobación, siempre comienza con *

Entonces con respecto a esto en la aplicación que se encuentra en internet GPS Visualizer se podía ver la ruta que se seguía, además mostraba los puntos en los que se encontraba humedad dentro del terreno de exploración.

2.2 Diseño electrónico.

En lo que respecta al diseño electrónico se hicieron las pruebas debidas sobre un protoboard, montando la circuitería en la que solo se hizo uso de cuatro sensores ultrasónicos uno para la parte delantera, trasera, izquierda y derecha del robot móvil; todo se realizó con un Arduino UNO puesto que el Shield del GPS solo podía hacerse uso con ese modelo de Arduino (figura 4); al observar la poca cantidad de pines con las que se podía contar se tuvo que hacer un arreglo con un multiplexor y un demultiplexor para los 4 sensores ultrasónicos y el sensor de humedad trabaja con un pin analógico, además de las salidas a utilizar para enviar las señales a los motores.

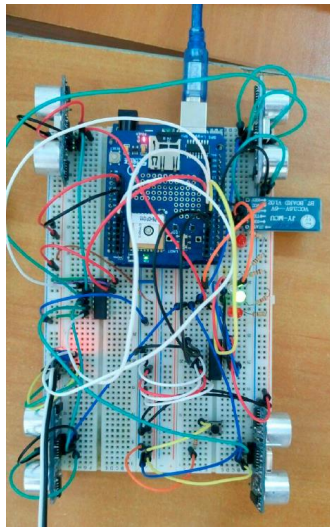


Figura 4. Vista superior de la circuitería implementada.

También se añadió un módulo Bluetooth para enviar estos datos guardados a una aplicación móvil diseñada, en la que la persona pueda estar verificando en su celular la ubicación del robot. Las pruebas con el dispositivo se hicieron con una batería recargable para hacer diversos recorridos y estar verificando que los datos eran guardados además de que se marcarán los puntos donde se humedecía el sensor FC-28 (figura 5).

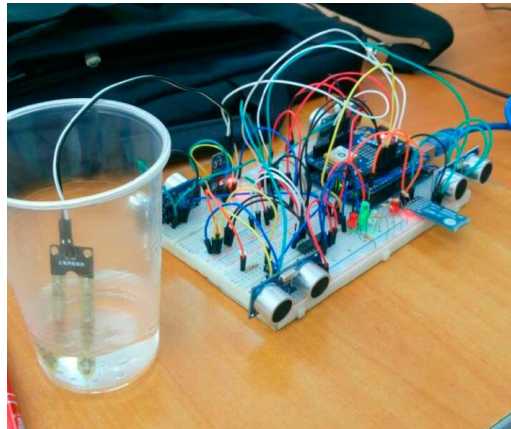


Figura 5. Pruebas de sensores montados en el robot móvil.

2.3 Aplicación móvil.

Para el desarrollo de la aplicación móvil se hizo uso del entorno de Android Studio que se aprecia en la figura 6 en el cual se requiere conceptos de programación en Java, está basado en el software IntelliJ IDEA de JetBrains. Se fue haciendo el diseño para poder observar en la pantalla del celular hacia qué dirección se dirige el robot móvil, primeramente, se fue haciendo la conexión por Bluetooth con el que se encuentra en la circuitería que irá montada sobre el dispositivo, se hizo una prueba primero con el encendido y apagado de un led con una aplicación sencilla.

Después lo que se hizo fue obtener los datos guardados en la memoria colocada en el Shield del GPS en el celular y ya con eso que se mostrara en el celular la ubicación dando la latitud y longitud, con esto observar en que punto sí hubo humedad.

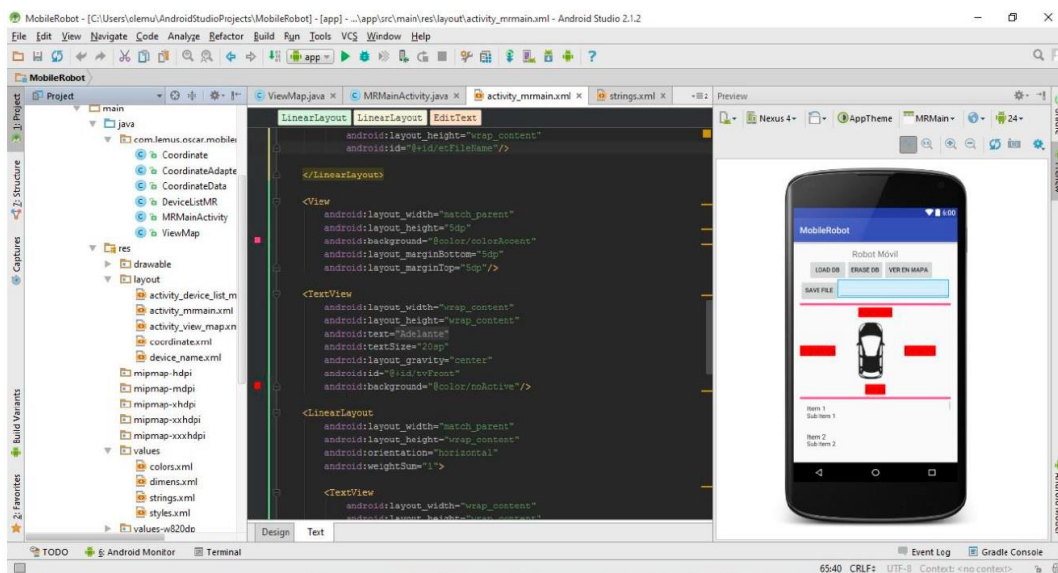


Figura 6. Android studio.

En la figura 7 se observa la aplicación descargada en un celular móvil con sistema operativo Android, la cual contiene diversos botones para hacer el guardado de archivo, descargar la posición y observar la longitud y latitud, ya con esto poder observarse en un mapa de OpenStreetMap.

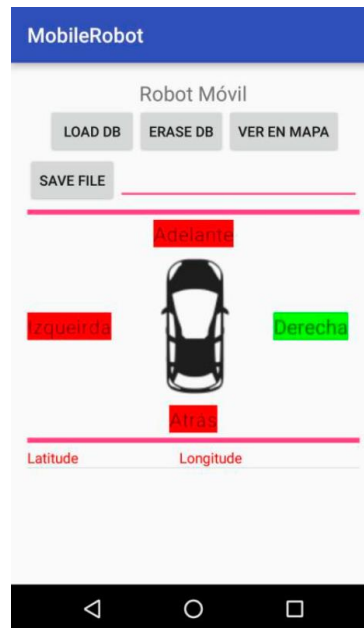


Figura 7. Aplicación móvil en celular con SO Android.

3. Resultados.

Los primeros resultados con respecto a la programación fue la obtención de la matriz dada por las neuronas ya entrenadas, además de que se nos indicaran cuales eran los pesos y bias más apropiados para la obtención de la matriz ya mencionada (figura 8).

```
w1 =
-1.9297  -3.0054  0.1144  6.6970
-2.9303  -0.0637  8.2350  0.2260
-1.5331  5.0651  -6.7401  5.0552
2.3902   1.7816  4.2843  -6.0167

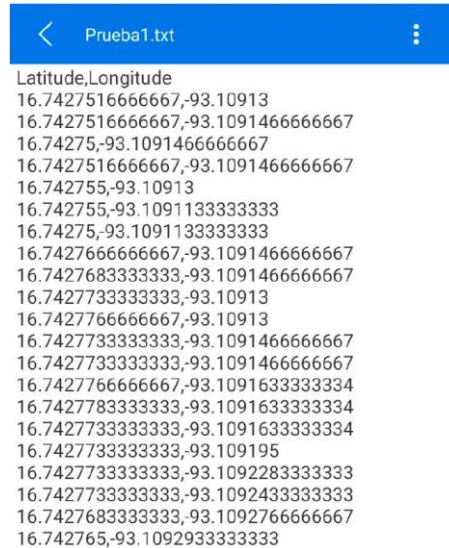
b1 =
6.3076
-3.4431
-0.4439
3.2458

w2 =
-2.2982  -0.7529  0.7796  2.5676
0.4774   -8.9804  1.9640  0.8466
2.6394   -2.2430  0.5034  -2.7472
3.4145   -4.4876  0.9579  -2.4312
-4.2235   5.0130  4.5686  0.3417
2.8706   -7.6595  4.2369  -2.7627
```

Figura 8. Pesos sinápticos y bias de la arquitectura neuronal desarrollada.

Los resultados en lo electrónico fueron exitosos pues todo funcionó correctamente conforme a los esperado, solo deseando haber avanzado un poco más respecto a montarlo en una baquelita, pero ya será más adelante.

El código de Arduino también fue un resultado muy bueno, ya que la tarjeta de Arduino UNO fue suficiente para abastecer a todos los sensores y se pudo colocar toda esa programación dentro del dispositivo en el que ya mencionado con la aplicación se pudo hacer el guardado de la posición obteniendo un archivo “txt” que con esos datos se obtenía la latitud y longitud como se puede observar en la figura 9.



```

Latitude,Longitude
16.7427516666667,-93.10913
16.7427516666667,-93.1091466666667
16.74275,-93.1091466666667
16.7427516666667,-93.1091466666667
16.742755,-93.10913
16.742755,-93.1091133333333
16.74275,-93.1091133333333
16.7427666666667,-93.1091466666667
16.7427683333333,-93.1091466666667
16.7427733333333,-93.10913
16.7427766666667,-93.10913
16.7427733333333,-93.1091466666667
16.7427733333333,-93.1091466666667
16.7427766666667,-93.1091633333334
16.7427783333333,-93.1091633333334
16.7427733333333,-93.1091633333334
16.7427733333333,-93.109195
16.7427733333333,-93.1092283333333
16.7427733333333,-93.1092433333333
16.7427683333333,-93.1092766666667
16.742765,-93.1092933333333

```

Figura 9. Datos en formato TXT donde se guarda la posición gps del robot.

En la siguiente figura 10 se puede observar cómo se marca la posición en OpenStreetMap, los cuales indican que el sensor FC-28 detectó humedad y es para que el robot móvil se dirija a esa posición.

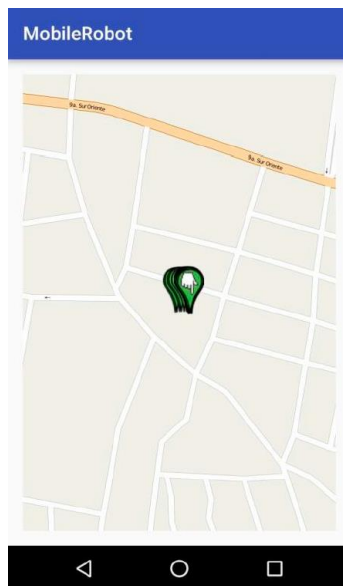


Figura 10. Posición de la actividad del sensor de humedad dentro del terreno explorado.

Conclusiones.

Con los resultados obtenidos se comprueba el funcionamiento del sistema desarrollado y por otro lado se pueden realizar muchos más proyectos como por ejemplo en el rescate de personas cuando suceda algún terremoto ya que se cuenta con el GPS, o al igual para la ayuda policial en el que se pueda colaborar en encontrar personas desaparecidas o secuestradas. Además de mostrar las capacidades de comunicación del sistema con aplicaciones móviles para obtener las ubicaciones del robot en sus recorridos en un terreno de prueba no mayor a 200 metros cuadrados.

Créditos.

Universidad de Ciencia y Tecnología Descartes, Centro de Investigación, Desarrollo e Innovación Tecnológica. Programa Delfín.

Agradecimientos.

Se agradece al alumno de programa delfín 2017 que participo activamente en este proyecto de investigación, así mismo las facilidades de la Universidad de Ciencia y Tecnología Descartes para el espacio y material para el desarrollo del proyecto.

Referencias Bibliográficas.

Fernández, M. Á., Fernández, D., & Valmaseda, C. (2009). Planificación de trayectorias para un robot móvil. Recuperado el 2016, de <http://eprints.sim.ucm.es/11301/1/MemoriaProyectoSSII.pdf>

Hagan, Martin T.; Demuth, Howard B.; Beale, Mark, (1996). Neural Network Desing, 1° Edición, USA: PWS Publishing Company

Información de los autores.



Wendy Yesenia Govea Rodríguez, Estudiante de ingeniería mecatrónica en el tecnológico de Michoacán.



Oscar Alejandro Lemus Pichardo, Estudiante de ingeniería mecatrónica en el IPN.



Alejandro Medina Santiago, Doctor en Ingeniería Eléctrica, Cinvestav-IPN, Universidad de ciencia y tecnología Descartes. Investigador.



Lorena del Rosario Rojas Nucamendi, Ingeniero en sistemas computacionales, Colegio de ingenieros en sistemas computacionales de Chiapas. Instituto de Ciencia, Tecnología e Innovación del Estado de Chiapas.