Uso de Matlab y Simulink para el Control de Robots y la observación de Sensores de Tacto y Ultrasónico

Martha Aguirre [1], Jonathan Cagua [2], Oswaldo Criollo [3], Carlos Valdivieso [4]
Facultad de Ingeniería en Electricidad y Computación [1] [2] [3]
Escuela Superior Politécnica del Litoral (ESPOL)[1] [2] [3]
Campus Gustavo Galindo, Km 30.5 Vía Perimetral, Apartado 09-01-5863. Guayaquil, Ecuador [1] [2] [3]
iaguirre@fiec.espol.edu.ec[1], jcagua@ fiec.espol.edu.ec [2], wcriollo@ fiec.espol.edu.ec [3],
cvaldiv@fiec.espol.edu.ec [4]

Resumen

El desarrollo de este proyecto tiene como objetivo hacer uso del Kit Lego Mindstorms NXT para realizar una aplicación del tipo robótico y profundizar en el funcionamiento de ciertos sensores del Kit, el software a utilizar será Matlab. El diseño es una representación de un robot explorador, el cual se moverá libremente en un área determinada por el usuario, el robot evitará obstáculos y tomará independientemente una decisión del camino a seguir, parte de este proceso se lo realizará por el uso del sensor ultrasónico, el cual detecta objetos cercanos, adicionalmente se colocó el sensor de tacto en la parte inferior para colisiones fuera del alcance del ultrasónico.

El programa usado para el movimiento del explorador fue desarrollado en Simulink, una herramienta de Matlab. Que permite la programación en bloques para el movimiento del robot. Posteriormente el programa es cargado en el cerebro del NXT.

Este diseño no solo toma en cuenta el movimiento del explorador sino que además realiza una adquisición de datos de una variable determinada para luego mostrarse al usuario. Esta variable podría ser de diferente tipo. En este caso se empleó el sensor de sonido para capturar los niveles de ruido en decibelios presentes en el ambiente. Los datos serán enviados por medio del dispositivo bluetooth que contiene el Lego a una computadora, en donde serán debidamente procesados por un código desarrollado en Matlab y presentados posteriormente en una interfaz gráfica. En donde se mostrará la adquisición mediante una ventana graficadora, además se tendrá opciones para guardar los datos en una hoja de Excel o en un archivo .bmp, permitiendo además al usuario elaborar una base de datos con lo sensado. Como resultado final se logrará tener un robot explorador con decisiones de movimiento realizando al mismo tiempo una toma de datos del lugar.

Palabras claves: *Explorador, Lego NXT, Simulink, Matlab.*

Abstract

This project uses the Lego Mindstorms NXT Kit to develop an application that emphasizes the use of certain sensors of the Kit using the Matlab software. The design is the representation of an explorer robot, which will be able to move freely on a determined area defined by the user. The robot will avoid obstacles and take independent decisions. Part of his process will be donned using an ultrasonic sensor, which detects nearby objects; in addition a touch sensor has been located in the lower part of the robot to avoid collisions that cannot be detected by the ultrasonic sensor.

The program to move the robot was developed using Simulink, a Matlab tool that allows a block programming approach for the robot movements. Afterwards the program is loaded into the NXT brain.

This robot was designed not only to control movements but also to serve as a data acquisition unit of a variable that will be shown to the user later. This variable represents sound data. In this case a sound sensor was used to capture noise levels, in decibels, from the environment. The capture data is sent from the Lego Mindstorm robot through a Bluetooth interface to the computer where it will be processed using a program developed in Matlab. The data will be presented later on a graphic interface. On this interface the acquired data will be shown on a graphic window with the option to save the data on an Excel sheet or a .bmp file. This will allow the user to create a database with the sensing information. The final result it will be an explorer robot that can take movement decisions and make data acquisitions at the same time.

Keywords: *Explorador, Lego NXT, Simulink, Matlab.*

1. Introducción

Este proyecto de graduación tiene como objetivo destacar el uso de los sensores de tacto y ultrasónico del LEGO MINDSTORMS NXT bajo la programación de Simulink de Matlab con su respectiva aplicación para la adquisición de datos. [1]

2. Fundamento teórico

Este proyecto está basado en el uso del lego Mindstorms para el diseño de un robot explorador, haciendo uso de sensores como lo son el de tacto y el ultrasónico. Además para poder simular el sensor que va a adquirir datos hemos hecho uso del sensor de sonido. Pues lo que se analizará es una variable del tipo analógica que puede provenir de cualquier dispositivo que puede ser de presión o de temperatura, estos datos serán procesados por el lego y a la vez procesadas por el entorno Matlab junto con Simulink.

3. Especificaciones de hardware

Uno de los componentes básicos del diseño lo conforma el lego NXT, y su parte central lo es el "Ladrillo Programable" de LEGO, conocido como Brick, que no es más que una microcomputadora programable que puede enviar señales a los puertos de salida y recibir información de sensores conectados a sus puertos de entrada.[2]

Los sensores que utiliza el kit el Lego Mindstorms NXT son:

- Servos motores.
- Sensor Óptico.
- Sensor ultrasónico.
- Sensor de tacto
- Sensor de sonido.

3.1 Servomotores

Los servomotores que suministran el NXT son tres motores de corriente continua y funcionamiento PWM. A pesar de su reducido tamaño, esconden en su interior un complejo sistema de reducción por tren de engranajes y un sensor de rotación de tipo tacométrico. [3]

Figura 1. Servo Motor con imagen de su parte interna.

3.2 Sensor Ultrasónico

El sensor Ultrasónico al contrario del resto de sensores, no devuelve los valores en ninguna escala ni porcentaje, sino en centímetros o en pulgadas.

Detecta las distancias y el movimiento de un objeto que se interponga en el camino del robot, mediante el principio de la detección ultrasónica. Es capaz de detectar objetos que se encuentren desde 0 a 255 cm, con una precisión relativa de +/- 3 cm.

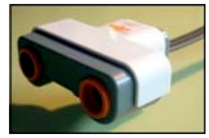

Figura 2. Sensor Ultrasónico.

3.3 Sensor de Tacto

El sensor de tacto permite detectar si el bloque que lo posee ha colisionado o no con algún objeto que se encuentre en su trayectoria inmediata. Al tocar una superficie, una pequeña cabeza externa se contrae, provocando una variación de energía de 0 a 5 V.

Figura 3. Sensor de Tacto.

3.4 Sensor de Sonido

El sensor acústico puede configurarse para devolver los valores de medida en decibelios (dB).

La sensibilidad máxima se encuentra en los 90 dB (aproximadamente el mismo nivel sonoro que una calle ruidosa con mucho tráfico).

Figura 4. Sensor de Sonido.

3.5 Fundamentos de la Tecnología Aplicada

Para este proyecto vamos a establecer una comunicación con el puerto COM de la computadora establecida por el software del Bluetooth y lega Mindstorms NXT.

Bluetooth

Bluetooth, es una especificación industrial para Redes Inalámbricas de Área Personal que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz.

Figura 5. Comunicación entre el brick y computador.

4. Diseño e implementación

La parte del hardware está compuesta por un computador, el cual dispone de un bluetooth dongle por el cual se realizará la adquisición de datos que ha sido debidamente procesada y enviada por medio del robot explorador del kit lego Mindstorm.

Figura 6. Comunicación Bluetooth entre el lego y computadora.

Simulink es el que permite realizar los programas respectivos para la movilidad del explorador, para la adquisición de datos y además la interfaz que se realizará para el usuario. [4]

ECRobot es un toolbox para Matlab, el que nos permite hacer uso de los bloques creados exclusivamente para el manejo de los Lego Mindstorms. [5]

4.1 Descripción del Programa para la movilidad del Robot

Esta es la programación desarrollada en Simulink, aquí se detalla todos los boques del lego que proporciona Simulink, así como los bloques que representan el sensor ultrasónico, sensor de tacto, los motores y el dispositivo bluetooth. [6]

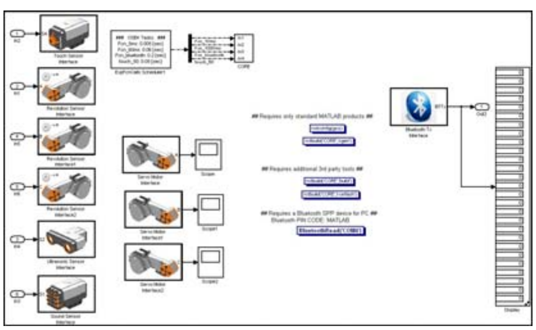

Figura 7. Programación gráfica del Sistema en Simulink.

El bloque ExpfcnCalls Scheduler1 sirve para declarar cada cuanto tiempo efectúa la funciones dentro del bloque CORE.

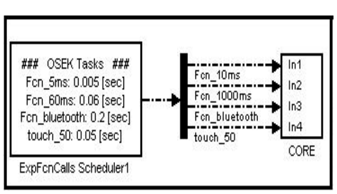

Figura 8. Bloque ExpfcnCalls Scheduler1.

Estos tres bloques son los que permite el movimiento del explorador.

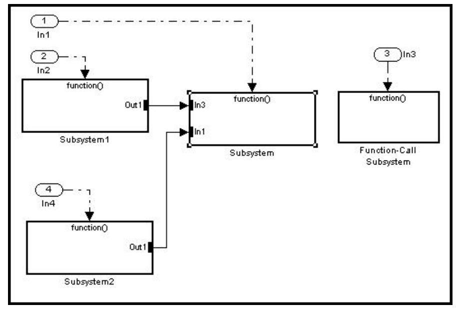

Figura 9. Bloques internos del Bloque CORE.

Otro punto importante fue la necesidad de agregar un bloque llamado Memory debido a que sin este no se podía realizar una retroalimentación en la aplicación puesto que surgía un error en la compilación de la aplicación.

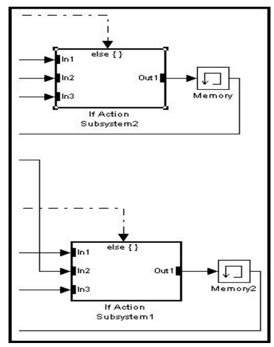

Figura 10. Bloques internos del Bloque if Action Subsystem1.

4.2 Programa para el envío de datos por parte del Robot Explorador

Para el envío de datos desde Simulink se usó el bloque de bluetooth, de esta manera la adquisición de datos también es incluida en el programa realizado por Simulink. En donde consta la interface del bluetooth así como la interface del sensor de sonido.

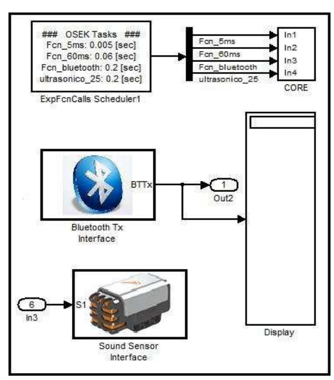

Figura 11. Bloques para la adquisición de datos.

Este el procedimiento empleado para obtener los datos del sensor y la conversión realizada para enviarlos vía bluetooth. Cada indicar que la comunicación se realiza de la siguiente manera: se envía 4 paquetes de 8 bits y a eso se le agrega un bit de inicio y uno de parada.

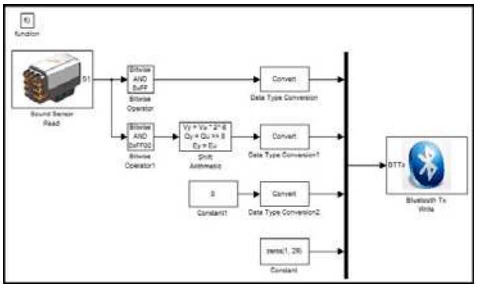

Figura 12. Bloques internos para la adquisición de datos.

4.3 Diseño de la aplicación realizada en Matlab

Para la aplicación hecha en Matlab utilizamos la herramienta GUI que nos permitirá desarrollar una interfaz a nivel de usuario. Las ventajas que trae esta herramienta es que se puede enlazar cada botón con ciertas instrucciones tanto para la adquisición o graficación de los datos. [7

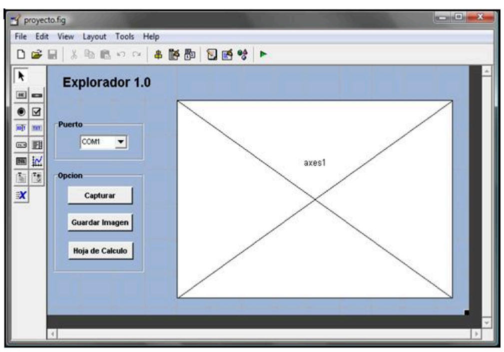

Figura 13. Ventana de GUI para la adquisición.

El botón capturar que se encuentra en la interface es la que se servirá para capturar los datos enviados por el Lego.

```
disp('Ready for DAQ via Bluetooth. Run your NXT!');
%El for es usado para adquirir un numero de muestras
for i = 1:NUM_OF_CYCLE
    packet = fread(s, PACKET_SIZE, 'uint8');
    %Comparamos la longitud del tamaño de trama q adquirimos
    if length(packet) == PACKET_SIZE
        SoundSensorDAQ(i) =1023-( packet(4)*256+packet(3));
        %UltrasonicSensorDAQ(i) = packet(5);
```

En estas líneas se encuentra tanto la parte de graficación en el axes como en la hoja de Excel. Se realizó la transpuesta a la variable SoundSensor para hacerlo una columna y enviarla a Excel. [8]

```
a=SoundSensorDAQ';%Hacemos transpuesta a la matriz para graficar en excel
fclose(s);%cerramos el puerto
disp('DAQ is finished.');
assignin('base', 'SoundSensorDAQ', SoundSensorDAQ);
%Con plot mostramos la grafica en GUI
plot(PERIOD:PERIOD:NUM_OF_CYCLE*PERIOD, SoundSensorDAQ(1:NUM_OF_CYCLE));
title('Sound Sensor A/D ');
%Esta funcion es para graficar en EXCEL y establecemos las celda donde
%seran enviado los datos
xlswrite('datos',a(1:NUM_OF_CYCLE),'A1:A50');
```

La función btnGuardar se realizó para grabar una imagen .bmp de los datos que se muestran en el axe y en un futuro todas estas imágenes puedan ser almacenado en una base de datos.

```
function btnGuardar_Callback(hObject, eventdata, handles)
formatos = {'*.bmp','BMP (*.bmp)'};
[nomb,ruta] = uiputfile(formatos,'GUARDAR Datos');
```

5. Pruebas del proyecto

El diseño básicamente es del robot explorador, en el diseño consta los sensores de tacto y ultrasónico que son de nuestro interés y se encuentran colocados en su posición correspondiente para aprovechar todas sus características.

Figura 14. Diseño del robot explorador.

En la interfaz que se realizó en GUI se muestran los datos obtenidos, esta gráfica presenta los datos que el sensor de sonido sensa por un tiempo de 20 segundos, todo esto después que el usuario presiona el botón de Capturar.

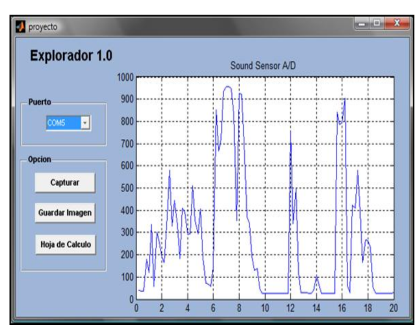

Figura 15. Interfaz para la adquisición de datos.

Como una opción para el usuario optamos por colocar un botón de Guardar Imagen, lo que permite guardar los datos obtenidos en un directorio que el mismo establece.

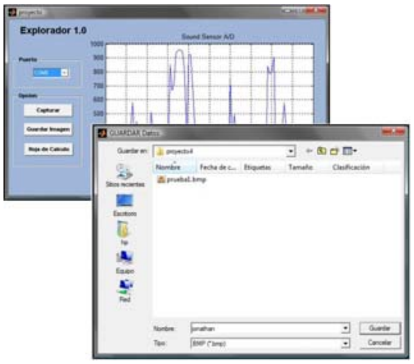

Figura 16. Ventana que permite Guardar Imagen.

Esta imagen se guarda en formato .bmp, esto se pudo obtener gracias a los comandos propios en la programación de GUI. [9]

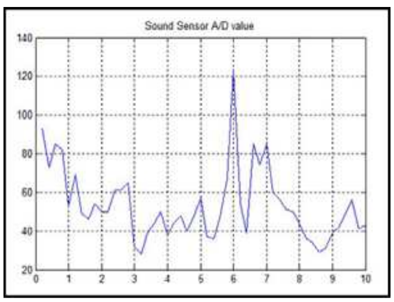

Figura 17. Imagen .bmp que se guarda.

Otra opción que le añadimos en la programación en GUI es la de guardar los datos en una hoja de Excel por medio del botón Hoja de Excel, y poder obtener una gráfica de los datos obtenidos.

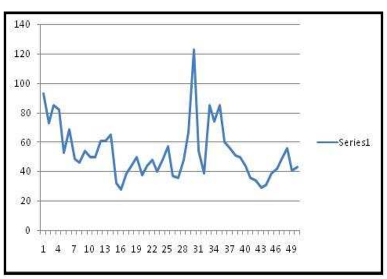

Figura 18. Gráfica que se obtiene de Excel.

6. Conclusiones

Con el desarrollo del proyecto llegamos a la conclusión que la elección de Simulink para la programación del Lego NXT fue correcta, ya que este no presentó un nivel de dificultad muy grande y hace que el robot explorador no tenga una dependencia con una base o computador para cumplir su propósito de movilidad, sin que afecte el factor de distancia.

La adquisición que se realizó gracias a las herramientas de Matlab y Simulink permitió hacer la lectura eficiente del sensor del robot explorador, para luego con la herramienta de GUI presentar dichos datos en una interfaz que permita al usuario manejar las opciones que esta interfaz le permite.

Se logró utilizar todas las herramientas necesarias para desarrollar un sistema en Simulink de tal manera que controle el robot explorador, siendo ECRobot una de las herramientas indispensables para alcanzar este fin, pues permitió agregar un toolbox de Lego NTX en Simulink para realizar la programación necesaria para lograr el objetivo planteado.

Por medio de las prácticas realizadas se conoció que en la parte del muestreo la frecuencia no sea muy alta por lo que nos dió errores a la hora de receptar datos por el motivo de que el convertidor analógico digital propio del lego está dentro de un microcontrolador y este tiene un retardo para dicha conversión, por lo que se obtuvo lecturas erróneas que llegó a valores de 0, con las muestras que se desarrolló se conoció que el tiempo de muestreo adecuado es de 0.2 segundos, este tiempo fué el que no nos trajo errores a la hora de trabajar y obtener las muestras.

7. Referencias

[1]. Lego con Matlab, Página HTML, www. blogs.mathworks.com

[2]. Lego Mindstorms, Página HTML, http://www.wikipedia.org/wiki/Lego_Mindstorms

[3]. MECHATRONICS, Archivo PDF, www.microchip.com

[4]. Simulink y Excel, Archivo PDF, www.tecnun.es/asignaturas/.../aprendainf/matlab70/mat71.pdf

[5]. Legos-ECRobot, Página HTML, http://lejos-osek.sourceforge.net/api.htm

[6]. Firmware de lego, Página HTML, www. mindstorms.lego.com

[7]. Excel con Matlab, Página HTML, http://www.robotshop.ca/hitec-hs311-servo.html

[8]. Introducción a MATLAB / SIMULINK, Archivo PDF, www.fceia.unr.edu.ar/dsf/files/TP_Introd.pdf

[9]. Aprenda Matlab, Cátedra D.S.F, Archivo PDF, www.tecnun.es/asignaturas /matlab60/matlab60.pdf