

Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de
Telecomunicación

Evaluación y análisis de los paquetes de apoyo para
LEGO MINDSTORMS EV3 y Android en Simulink

Autor: Juan Antonio Núñez Sánchez

Tutor: Federico Cuesta Rojo

Dpto. de Ingeniería de Sistemas y Automática
Universidad de Sevilla

Sevilla, 2018



Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de Telecomunicación

Evaluación y análisis de los paquetes de apoyo para LEGO MINDSTORMS EV3 y Android en Simulink

Autor:

Juan Antonio Núñez Sánchez

Tutor:

Federico Cuesta Rojo

Profesor titular

Dpto. de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2018

Proyecto Fin de Carrera: Evaluación y análisis de los paquetes de apoyo para LEGO MINDSTORMS EV3 y
Android en Simulink

Autor: Juan Antonio Núñez Sánchez

Tutor: Federico Cuesta Rojo

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2018

El Secretario del Tribunal

A mi familia

A mis amigos

A mis maestros

Resumen

Se han analizado las posibilidades de programación que ofrecen los paquetes de apoyo para LEGO MINDSTORMS EV3 y Android en Simulink y se han llevado a cabo diferentes ejemplos prácticos, concluyendo que es una plataforma interesante para el aprendizaje relacionado con la automatización, robótica o ingeniería de sistemas. Esto se debe a su bajo coste comparado con otras plataformas robotizadas y a la variedad de aplicaciones que se pueden crear gracias a las librerías que nos proporciona Matlab integrándolo con Simulink, que goza de una gran comunidad de desarrollo donde la gente comparte sus proyectos y aplicaciones. Además, al incluirse el toolbox de Android hemos ampliado los posibles sensores que puede usar el robot para recopilar información y retroalimentarse.

Resumen	ix
Índice	xi
Índice de Tablas	xiii
Índice de Figuras	xv
1 Introducción	1
2 Paquete de apoyo LEGO MINDSTORMS EV3	3
2.1. <i>Hardware y Software</i>	3
2.1.1 Simulink y otros softwares de programación	3
2.1.2 Requisitos	3
2.1.3 Sensores y actuadores	3
2.1.4 Brick EV3	4
2.1.5 Robot	5
2.1.6 Comunicación	5
2.2 <i>Ejemplos</i>	6
2.2.1 Seguidor de línea	6
2.2.2 Controlar EV3 desde ordenador	11
3 Paquete de apoyo Android	15
3.1 <i>Hardware y software</i>	15
3.1.1 Matlab, Simulink y otros softwares de adquisición de datos	15
3.1.2 Requisitos	15
3.1.3 Sensores y actuadores	15
3.1.4 Comunicación	17
3.2 <i>Ejemplos</i>	17
3.2.1 Recepción y filtrado de señales	17
3.2.2 Control EV3 mediante Android	20
4 Calibración, integración y validación experimental	23
4.1 <i>Montaje dispositivo Android sobre EV3</i>	23
4.2 <i>Telemetría robot</i>	24
4.3 <i>Calibración sensores</i>	27
4.4 <i>Control EV3 mediante información sensores Android</i>	28
5 Conclusiones	31
Bibliografía	33

ÍNDICE DE TABLAS

Tabla 1: Comparativa bricks LEGO	5
Tabla 2: Comparativa móviles	15

ÍNDICE DE FIGURAS

Figura 1: Sensores y actuadores	4
Figura 2: Montaje robot	5
Figura 3: Seguidor de Línea	7
Figura 4: Montaje seguilínea	7
Figura 5: Sensor luz y velocidad	8
Figura 6: Control de velocidad	8
Figura 7: Giróscopo cabeceo	9
Figura 8: Detección del obstáculo	10
Figura 9: Distancia objeto frente a velocidad	10
Figura 10: Control robot EV3 desde PC	11
Figura 11: Máquina de estados Stateflow	12
Figura 12: Receptor UDP EV3	12
Figura 13: Sensores y actuadores	17
Figura 14: Filtros Android	18
Figura 15: Acelerómetro filtros	18
Figura 16: Giróscopo filtros	19
Figura 17: Receptor	19
Figura 18: Ángulos pitch y roll	20
Figura 19: Control robot LEGO EV3 mediante Android	20
Figura 20: Móvil integrado con el robot	23
Figura 21: Envío Android	24
Figura 22: Receptor ordenador	24
Figura 23: Telemetría giróscopo seguidor línea	25
Figura 24: Telemetría acelerómetro seguidor línea	25
Figura 25: Giro ambas direcciones giróscopo	26
Figura 26: Giro ambas direcciones acelerómetro	26
Figura 27: Giros 90°	27
Figura 28: Calibración giro 90°	27
Figura 29: Control Android	28
Figura 30: Control giro mediante Android	28

1 INTRODUCCIÓN

El proyecto tiene como misión principal evaluar y analizar las posibilidades que presentan los paquetes de apoyo para LEGO MINDSTORMS EV3 y Android en Simulink.

La plataforma LEGO MINDSTORMS ya va por la tercera generación, se basa en la construcción de robots mediante piezas LEGO además de sensores y actuadores controlados por un bloque programable de forma visual. Esta plataforma ha sido pionera en robótica educativa y ofrece diferentes posibilidades y niveles para iniciarse en el área de automatización, robótica o control de sistemas. Esto, junto con los sensores extra que nos va a proporcionar un dispositivo Android y las posibilidades de programación que nos ofrece Simulink, nos va a permitir ejecutar diferentes ejemplos prácticos y algoritmos de control.

Vamos a dividir el proyecto en tres partes: primero analizaremos y realizaremos algunos ejemplos del toolbox para LEGO MINDSTORMS EV3, a continuación, haremos lo mismo con el toolbox de Android y, por último, integraremos ambos.

2 PAQUETE DE APOYO LEGO MINDSTORMS

EV3

Este paquete nos permite programar y ejecutar algoritmos en robots LEGO MINDSTORMS EV3. Incluye una librería de bloques de Simulink para configurar y acceder a los sensores, actuadores e interfaces de comunicación del brick.

2.1. Hardware y Software

2.1.1 Simulink y otros softwares de programación

Para el desarrollo de programas con el LEGO EV3 la propia compañía nos ofrece un software gratuito de programación visual mediante bloques basado en LabView. Debido a su sencillez este software resulta limitado desde una visión más enfocada al aprendizaje de disciplinas relacionadas con la ingeniería como la parte de robótica y mecatrónica. Esto ha desembocado en el desarrollo de una librería para LabView específica para LEGO MINDSTORMS y a la aparición de una variedad de alternativas de desarrollo software, como RobotC basado en el lenguaje de programación C o LeJOS basado en Java. Incluso sistemas operativos extraoficiales que se le pueden instalar en una memoria microSD como ev3dev basado en Debian Linux. Nosotros hemos elegido trabajar con Matlab y Simulink ya que es un software con el que se ha trabajado en otras asignaturas de la carrera, nos ofrece muchas opciones de desarrollo al poderse integrar con otros toolbox propios y tiene una amplia plataforma de desarrollo donde la gente comparte sus proyectos y aplicaciones.

2.1.2 Requisitos

Se necesita Matlab versión 2014a o superior, en nuestro caso trabajamos con la versión 2017b, e instalar el Simulink Support Package for LEGO MINDSTORMS EV3 Hardware. También necesitamos el LEGO MINDSTORMS EV3 core set donde vienen las piezas y sensores necesarios para montar el robot y un adaptador wifi USB compatible, lego recomienda el NETGEAR N150 Wi-Fi USB Adapter (WNA1100), pero nosotros hemos probado con el Edimax N150 Wi-Fi nano USB Adapter (EW-7811Un). Por último, la versión del firmware del LEGO debe ser mínimo la V1.03E.

2.1.3 Sensores y actuadores

El set consta de dos motores grandes, con sensores de rotación incorporados con resolución de 1 grado y funciona a un máximo de 160-170 rpm. También se incluye un motor mediano con sensor de rotación con la misma resolución, al ser más pequeño y liviano que los grandes puede responder más rápidamente que este 240-250 rpm, pero es menos potente. Desde Simulink podemos darle un entero entre [-100, 100] dependiendo del sentido de giro que deseemos y podemos acceder al encoder que nos devuelve los grados acumulados que lleva girados el motor.

Dentro de los sensores nos encontramos con un sensor de color que puede detectar el color o la intensidad de luz que recibe reflejada o ambiental desde 0 que sería totalmente oscuro a 100. En el modo color reconoce siete

colores. El muestreo se puede configurar hasta 1kHz. También nos ofrece un sensor táctil que detecta si estamos presionando el pulsador o no. Otro sensor que incorpora es el girosensor que detecta el movimiento de rotación en un eje simple que nos indica mediante flechas, mide la tasa de rotación en grados por segundo hasta un máximo de 440 grados por segundo. El robot incorpora una función para registrar el ángulo de rotación total con una precisión de ± 3 en 90 grados, aunque a este valor interno no podemos acceder desde Simulink. Por último, tenemos un sensor ultrasónico que puede medir la distancia a un objeto a través de ondas de sonido de alta frecuencia. Detecta distancias entre 0 y 255 centímetros con exactitud de ± 1 cm. Existe un sensor infrarrojo y una baliza remota que no están incluidos en el set.

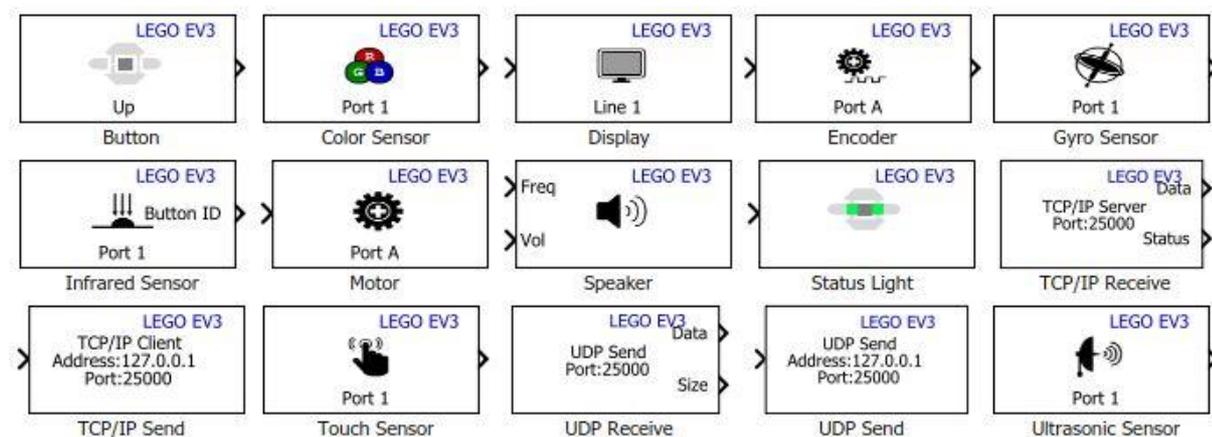


Figura 1: Sensores y actuadores

En La figura superior podemos observar los diferentes bloques que nos ofrece Simulink para trabajar con ellos. Aparte de los sensores y motores nombrados anteriormente podemos establecer comunicaciones por una red Wifi con otro dispositivo, también se puede acceder a los sensores del bloque EV3 para saber si pulsamos algún botón de este, mostrar números por pantalla, y emitir una luz o un sonido.

2.1.4 Brick EV3

Es el bloque inteligente programable del robot, donde se encuentra el firmware. Para el funcionamiento correcto con nuestra versión de Matlab y la compatibilidad con el dongle wifi debimos instalar la versión V1.04H del firmware oficial, el firmware se puede cambiar de forma sencilla con el software que ofrece LEGO. Dispone de cuatro puertos de entrada para conectar los sensores y otros cuatro de salida para los motores mediante los cables conectores. También de un puerto mini USB para conectarlo al PC, uno para la tarjeta SD por si queremos aumentar la memoria disponible y un puerto USB para agregar el conector USB a la red inalámbrica. Tiene una pantalla y 6 botones para navegar por la interfaz del dispositivo y así poder conectarse a una red inalámbrica o acceder a la información del dispositivo, como por ejemplo saber su IP cosa que nos será necesaria. Este modelo es la tercera generación de LEGO MINDSTORMS, abajo aparece una tabla comparativa con las distintas especificaciones de cada modelo.

Tabla 1: Comparativa bricks LEGO

	RCX	NXT	EV3
Fecha de lanzamiento	1998	Julio de 2006	Septiembre de 2013
Pantalla	Segmentada LCD monocromo	100×64 pixel LCD monocromo	178×128 pixel LCD monocromo
Procesador	HitachiH8/300 @16 MHz	AtmelAT91SAM7S256 (ARM7TDMIcore) @48 MHz	TI SitaraAM1808 (ARM926EJ-S core) @300 MHz
Memoria principal	32 KB RAM 16 KB ROM	64 KB RAM 256 KB Flash	64 MB RAM 16 MB Flash
Memoria expansible	No	No	Ranura microSD
Puerto USB	No	No	Sí
Wifi	No	No	Dongle Opcional vía puerto USB
Bluetooth	No	Sí	Sí

2.1.5 Robot

Para el montaje del robot con piezas LEGO vamos a construir el ejemplo de robot de dos ruedas que viene en las instrucciones de la caja, acoplándole el giróscopo para futuras medidas.

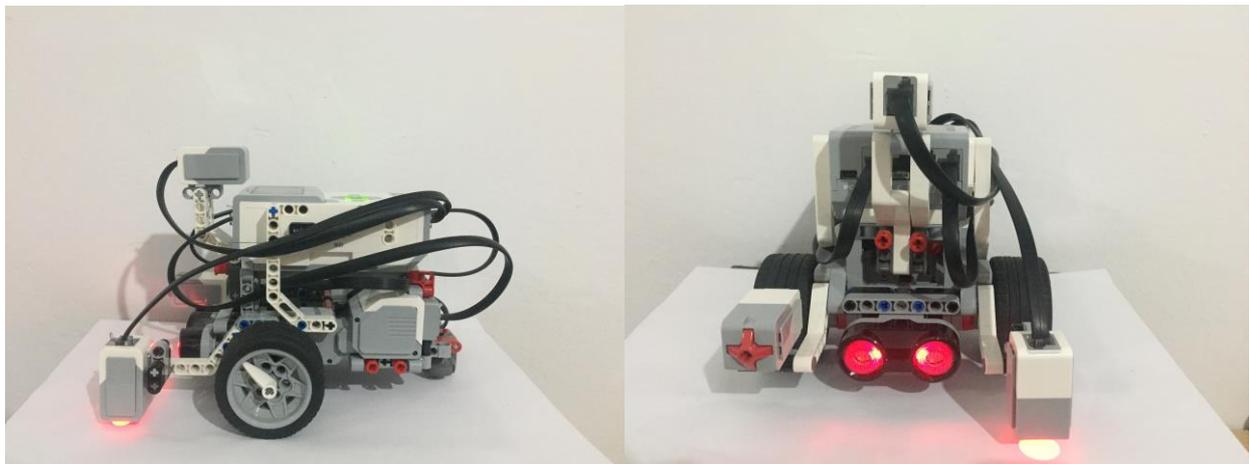


Figura 2: Montaje robot

2.1.6 Comunicación

La comunicación con Matlab para transmitirle el código del programa que generemos se puede hacer mediante el cable USB que viene incorporado, por Bluetooth, por Ethernet mediante el Apple USB Ethernet Adapter compatible o vía Wifi con el adaptador compatible. El bloque tiene algunas limitaciones para la conexión a una

red Wifi, ya que solo acepta encriptación WPA2 y la clave no puede contener caracteres extraños. Además, si la intensidad de señal no es muy alta porque tengamos el Reuter alejado la comunicación falla frecuentemente y se pierden paquetes.

Para solucionar este problema hemos convertido nuestro ordenador en un repetidor Wifi con la opción de Windows 10 de crear una zona de cobertura inalámbrica. Así podemos crear una red a la que nos sea más rápido conectarnos y la cobertura sea mayor. Una vez creada la zona de cobertura con una clave que nos permita introducirla rápido en con las teclas del brick necesitamos activar el Wifi dentro de la pestaña opciones, una vez activo le damos a conectar e introducimos la contraseña elegida. Dentro de las opciones tenemos también el apartado Brick Info donde debemos entrar para conocer la IP del dispositivo.

La parte interesante de conectarlo mediante Wifi es poder usar el external mode a la hora de correr el programa, esto te permite ajustar parámetros de entrada, monitorizar las salidas y generar gráficas de programas en ejecución. Si queremos ejecutar modelos de Simulink directamente en el hardware es suficiente con la conexión micro USB al ordenador.

Para parar una aplicación que se está ejecutando basta con pulsar el botón atrás en el EV3.

La configuración dentro de Simulink la encontramos dentro de Tools > Run on Target Hardware > Option. En la ventana que nos aparece seleccionamos la opción del LEGO en Hardware board y dentro de Target hardware resources > Host to Target Connection elegimos el tipo de conexión bien sea UBS, Wifi, Bluetooth o Ethernet y rellenamos la dirección IP y el ID del brick.

2.2 Ejemplos

2.2.1 Seguidor de línea

El robot seguidor de línea es un ejemplo sencillo clásico en el aprendizaje de la robótica que nos permite trastear con los bloques de Simulink vistos anteriormente mezclado con un algoritmo de control.

En un robot de dos pruebas que usa motores independientes para cada una aplicar la misma potencia a ambos no significa que el robot se vaya a mover en línea recta. Esto puede deberse a una distribución de peso no simétrica en el robot o a diferencias en la superficie por donde va cada rueda. Una mejor aproximación para reducir el cabeceo del robot es usar un controlador en bucle cerrado que ajuste la potencia basado en la diferencia de la rotación de las ruedas, dato que nos dan los encoder de los motores.

En este ejemplo se va a implementar un algoritmo de seguidor de línea en el robot que tenemos montado usando la retroalimentación del sensor de color para seguir la línea y la de los encoders para mantener una velocidad recta. Para la velocidad usaremos un controlador PI y el giro con un controlador PD que buscará el borde derecho de la línea. La acción derivativa estabiliza más rápidamente la variable controlada después de cualquier perturbación por eso nos interesa para controlar el giro, ya que aquí es más importante el transitorio, mientras que la acción integral está centrada en eliminar el error estacionario por eso lo usamos para que avance sin desviaciones.

El sensor de color lo tenemos que poner en el modo que refleja la intensidad de luz, este nos dará una salida entre 0 y 100 y tendremos que calibrar los valores de blanco y negro dependiendo de la claridad del espacio donde queramos llevar a cabo el seguimiento de línea e introducirlos como parámetros. El valor deseado que busca el controlador sería el valor medio entre blanco y negro para encontrar el borde de la línea. Si nos estamos acercando al color negro esto nos dará una diferencia positiva y como deseamos seguir la parte derecha de la línea este valor se resta a la velocidad del motor derecho y se suma al del izquierdo para que así gire hacia la derecha y se aleje del interior de la línea. Si nos estamos alejando del borde la diferencia entre el valor deseado y el valor del sensor va a ser negativa y el robot girará hacia la izquierda buscando el borde. Los codificadores de los motores izquierdo y derecho, que nos miden las vueltas que lleva dada la cada rueda en grados, se usan para calcular la velocidad derivando la posición de estas y esta estimación es lo que se usa para intentar llegar a la velocidad deseada que introducimos como parámetro. En los sensores y motores debemos configurar también dentro de sus opciones el número o letra de puerto a la que vamos a conectarlos físicamente en el brick.

Line Tracking

Copyright 2014-2015 The MathWorks, Inc.

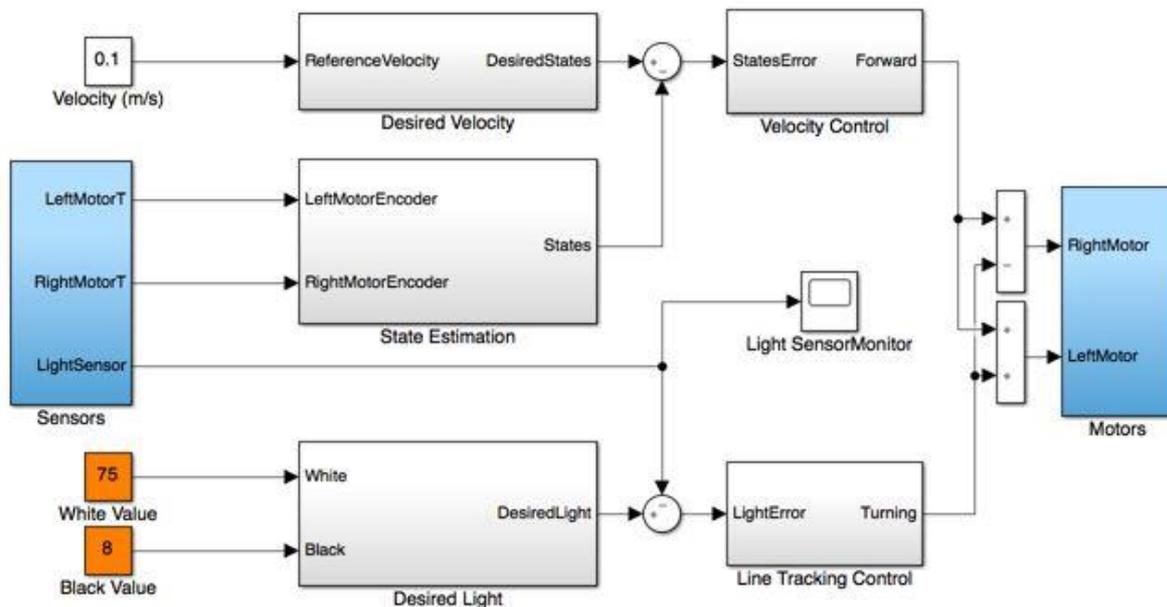


Figura 3: Seguidor de Línea

Una vez tenemos el diagrama de bloques, elegidos los parámetros de entrada y configurados los motores y sensores debemos elegir si ejecutar la aplicación en modo normal, para que sea autónoma, o external, para poder monitorizar las salidas y poder afinar en algoritmo. El modo de ejecución se puede seleccionar en la barra de herramientas. Si elegimos el normal pinchamos en Deploy to Hardware y el modelo se implementará en el brick, si optamos por el external mode pinchamos en run.

Para guardar datos en el workspace de Matlab y poder analizarlos a posteriori existe el bloque To Workspace o usar la opción log data to workspace dentro de un bloque scope. Para analizarlos y representarlos gráficamente vamos a ayudarnos del Simulation Data Inspector al que podemos acceder mediante el comando simplot en Matlab. Este inspector de datos de simulación permite inspeccionar y comparar datos de series temporales de una forma gráfica sencilla e intuitiva.



Figura 4: Montaje seguilínea

Resultados

Como podemos observaren la gráfica el seguidor de línea se comporta de la manera deseada, aumentando la velocidad del motor izquierdo a la vez que disminuye la del derecho cuando el sensor de luz detecta que se está acercando al negro al bajar la intensidad de luz y viceversa.



Figura 5: Sensor luz y velocidad

También vemos cómo funciona el control de velocidad adecuándola a la que le introducimos como deseada.

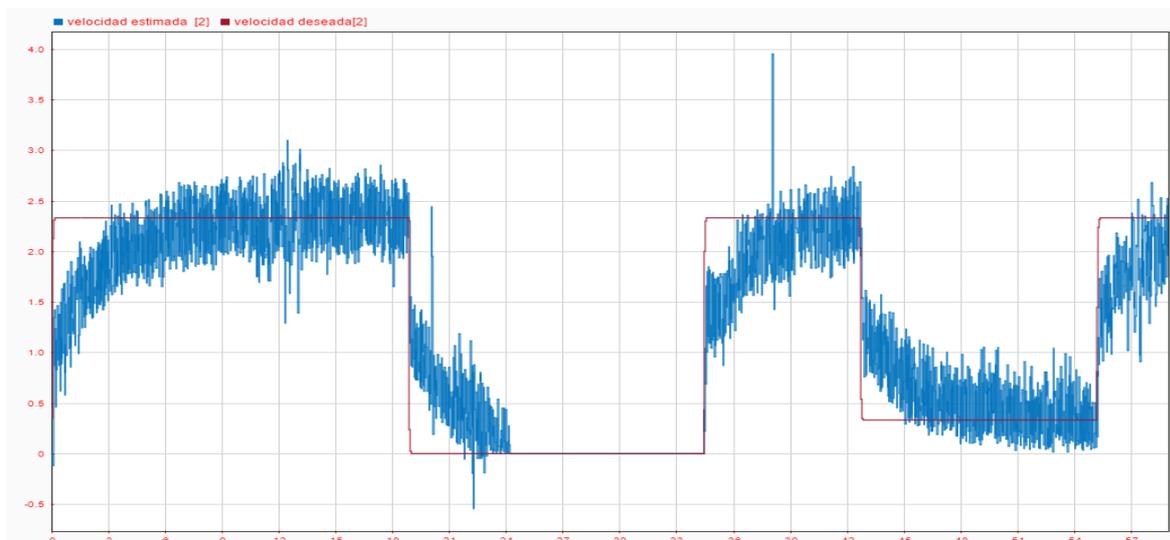


Figura 6: Control de velocidad

Con el modo de ejecución configurado en externo podemos variar los parámetros del control del seguimiento de línea mientras se está ejecutando la aplicación. De esta manera y gracias a la ayuda del giroscopo podemos ver como las modificaciones de estos influyen en la respuesta. Al principio teníamos una ganancia proporcional mayor, lo que se traducía en un comportamiento más oscilatorio y el robot cabeceaba. La última parte se disminuyó este valor a la mitad y vemos un control más suave y que se sigue una línea prácticamente recta.

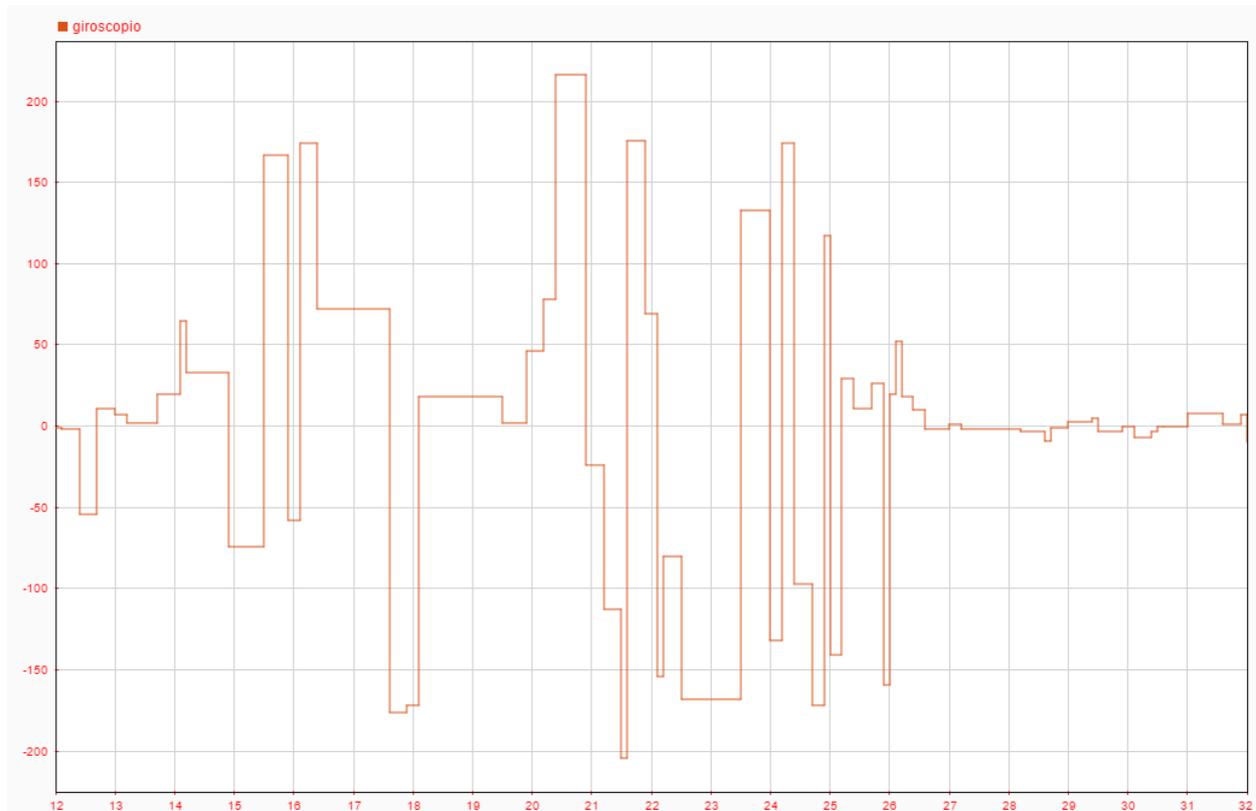


Figura 7: Giroscopo cabeceo

Modificaciones

Partiendo de este ejemplo podemos añadir la información de otros sensores para variar el comportamiento, una de las variaciones llevadas a cabo ha sido la inclusión del sensor de ultrasonidos, con el que se puede detectar la presencia de un objeto y la distancia a este en centímetros, reaccionando a él emitiendo una alarma y parándose. El bloque que hemos añadido detecta cuando la distancia a un objeto es menor a 10 centímetros, si es así se emite un sonido con el bloque del altavoz y la velocidad que se envía a los motores es cero. Mientras la distancia sea mayor al límite, la velocidad no se modifica. El bloque Switch de lo que se encarga es de pasar la señal 1 cuando la señal 2 cumple una determinada condición, en este caso que se active.

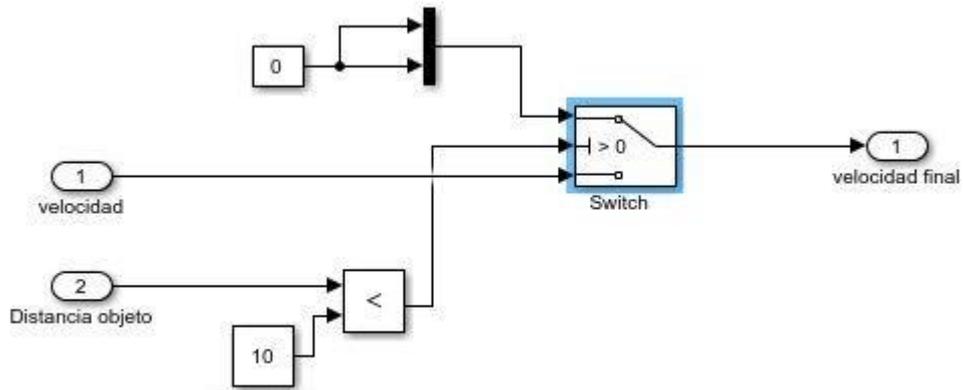


Figura 8: Detección del obstáculo

En la imagen siguiente podemos observar como cuando la distancia que es la línea amarilla es menor a 10 las velocidades de los motores pasan a cero.

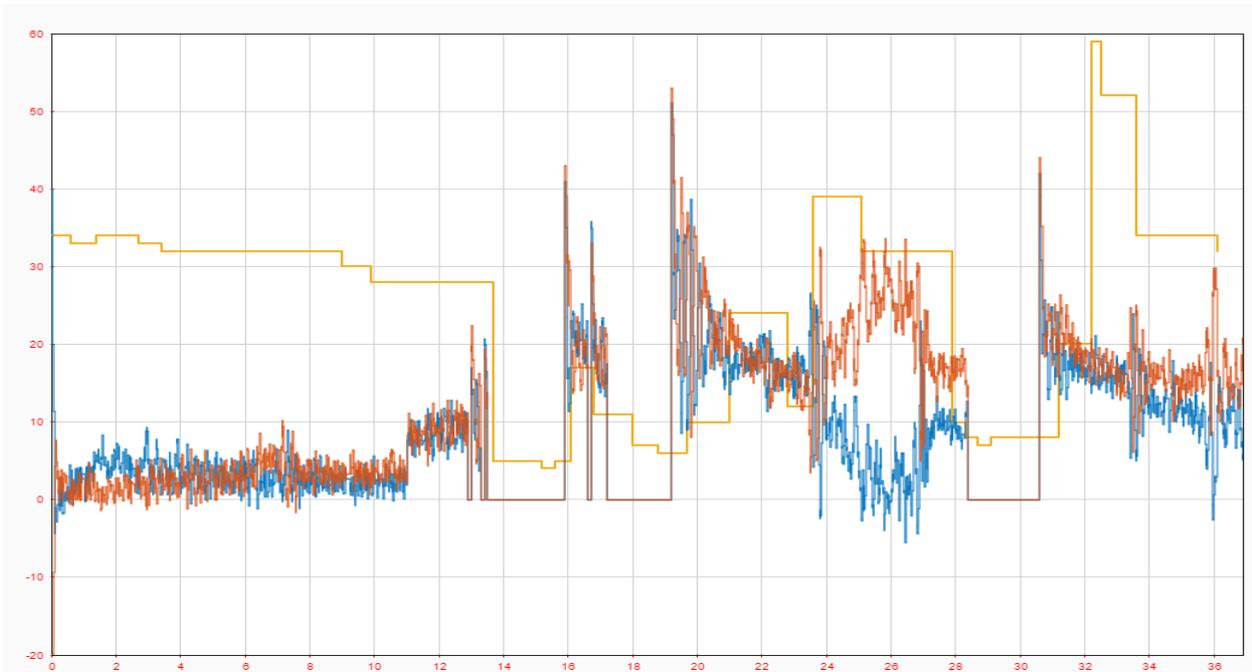


Figura 9: Distancia objeto frente a velocidad

2.2.2 Controlar EV3 desde ordenador

Este ejemplo muestra cómo crear un modelo de Simulink para comunicar el PC con el robot LEGO MINDSTORMS EV3 y que estos intercambien datos, este intercambio se lleva a cabo sobre UDP. Desde el modelo ejecutándose en el ordenador vamos a controlar los movimientos del robot. En el robot tendremos otra aplicación ejecutándose en paralelo que se encarga de recibir los datos y enviarlos internamente a los motores y actuadores.

El modelo del ordenador se compone de una serie de widgets del que su estado resultará en una velocidad para cada motor. Cada widget está asociado a un bloque constante que variará su valor al moverlo. Una vez decidida la velocidad de salida esta se visualizará por pantalla y se enviará mediante UDP. Para el envío usaremos el bloque de envío UDP que pertenece al Instrument Control Toolbox en el que debemos configurar introduciendo la dirección IP de nuestro EV3.

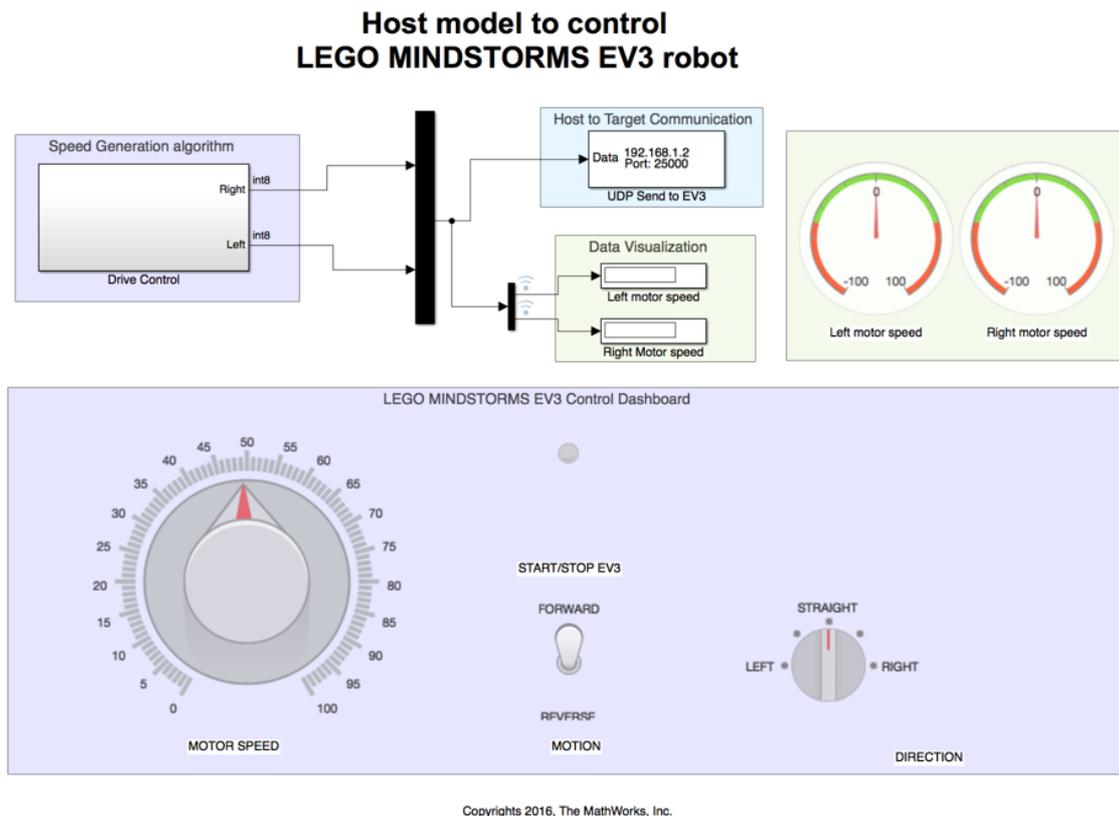


Figura 10: Control robot EV3 desde PC

La parte que decide la velocidad de cada motor en función de los valores que introducimos la realizaremos usando el Stateflow de Simulink. Este es un entorno para modelar y simular lógica de decisión combinatoria y secuencial basado en máquinas de estado y diagramas de flujo. Stateflow permite combinar representaciones gráficas y tabulares, lo que incluye diagramas de transición de estado, diagramas de flujo, tablas de transición de estado y tablas de verdad, con el fin de modelar la forma en que el sistema reaccionará ante los eventos, las condiciones basadas en el tiempo y las señales de entrada externas. Es una herramienta muy útil de cara al aprendizaje de robótica y mecatrónica ya que muchos problemas se resuelven mediante estos métodos de forma sencilla.

Las velocidades de los motores se van a controlar usando una máquina de estados cuyas entradas serán la velocidad, dirección, sentido y encendido y las salidas las velocidades que enviaremos a cada motor. Consta de cinco estados, cada uno asociado a la dirección en que se va a mover el robot y en cada cual se calcula una velocidad de los motores en función de las variables velocidad, sentido y encendido.

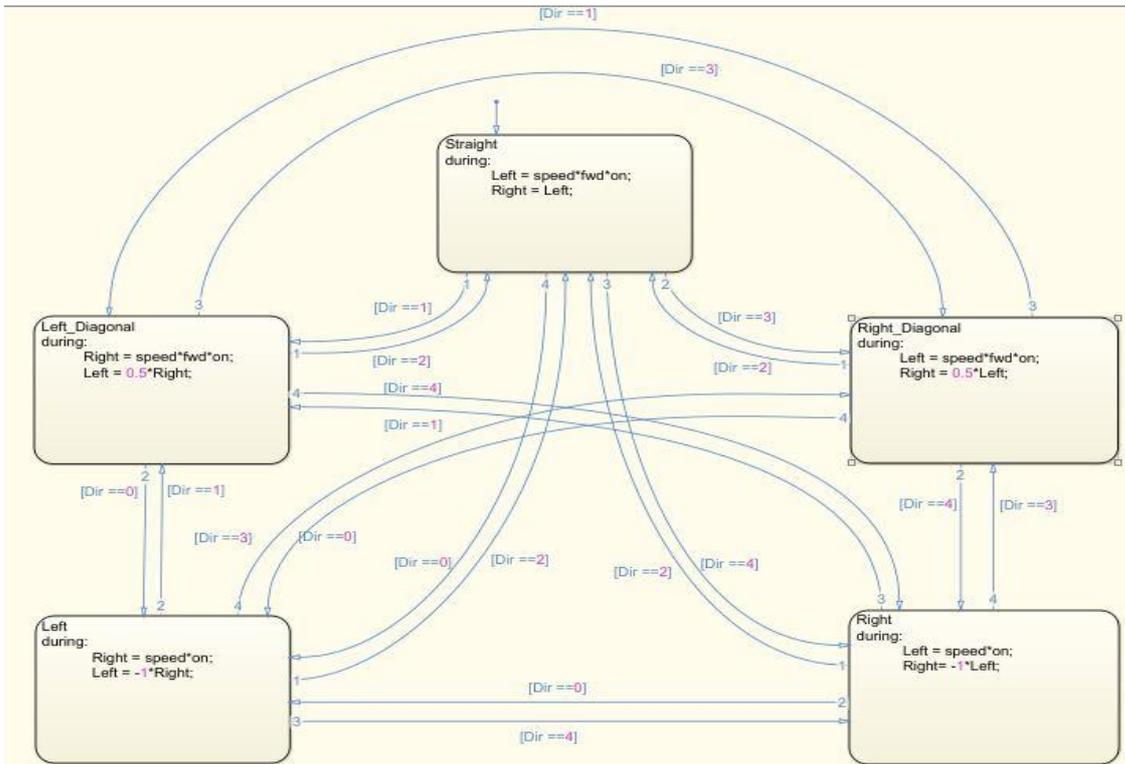
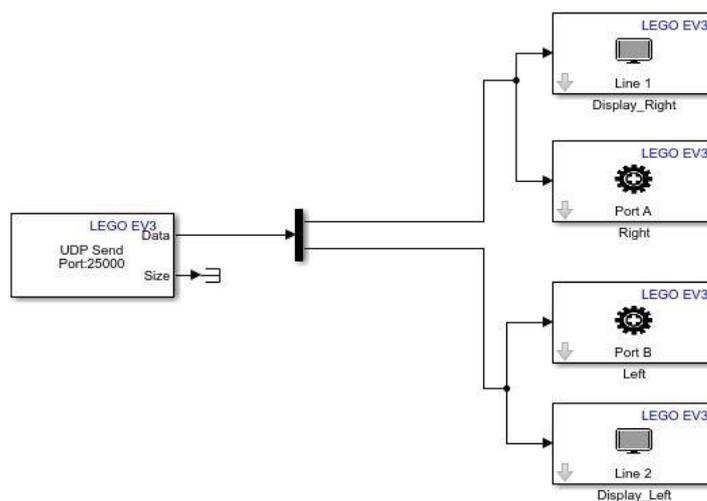


Figura 11: Máquina de estados Stateflow

Para la aplicación que correrá en el lego necesitaremos el bloque del receptor de UDP el cual debemos configurar introduciendo el puerto por el que debe escuchar, el número de datos que recibe y el tipo. Esto se conecta a los motores, también se ha añadido el display para que aparezcan las velocidades de cada motor en la pantalla del brick. Ambas aplicaciones se ejecutarán simultáneamente clicando run en la del ordenador y Deploy to Hardware, después de actualizar el Host to Target Connection, en la del robot.

LEGO MINDSTORMS EV3 Robot Control



Copyrights 2016 The MathWorks, Inc.

Figura 12: Receptor UDP EV3

Resultados

Podemos observar como el lego reacciona consecuentemente con las instrucciones que introducimos en la aplicación del ordenador. Pulsando el botón derecho y abriendo la máquina de estados en una nueva ventana podemos ver el estado en el que se encuentra en cada momento y la transición entre estos en tiempo de ejecución a la vez que movemos el interruptor rotatorio en la otra ventana.

Modificaciones

Hemos probado a añadir dos direcciones más en las que el robot pase de ir en línea recta a girar levemente en una dirección, a girar más pronunciadamente y a girar sobre sí mismo en esa dirección, e igualmente para la otra dirección, para ello se han introducidos dos estados más en el interruptor rotatorio que es el que controla la dirección. Después hemos creado dos estados nuevos dentro de la máquina de estados con sus transiciones correspondientes.

3 PAQUETE DE APOYO ANDROID

Este paquete nos permite programar y ejecutar algoritmos en dispositivos Android. Incluye una librería de bloques de Simulink para configurar y acceder a los sensores, capturar y reproducir audio y video, enviar datos a la nube, crear UI widgets y comunicarse con otros dispositivos mediante una interfaz de red.

3.1 Hardware y software

3.1.1 Matlab, Simulink y otros softwares de adquisición de datos

Existen varias aplicaciones en Android que te permiten transmitir a un ordenador los datos de los sensores en tiempo real o registrarlos para su posterior análisis. Por ejemplo, Sensor fusion app, Physics Toolbox Sensor Suite o phyphox realizan esa función. Matlab también ofrece su propia aplicación que instalada en el móvil se comunica con el programa ejecutándose en el ordenador. Nosotros vamos a escoger la opción que ofrece Simulink con su toolbox de Android debido a la facilidad de integración en otros proyectos y a las posibilidades de programación y análisis de datos que tiene. En este caso creamos la aplicación que necesitamos que se ejecute en el móvil o Tablet mediante la programación por bloques de Simulink. Una ventaja que tiene la App de Matlab frente a Simulink es que podemos acceder a la información del sensor magnético y al vector de orientación del dispositivo en la versión con la que estamos trabajando, pero en la última actualización han añadido estos sensores a Simulink también.

3.1.2 Requisitos

Se necesita Matlab versión 2014a o superior, en nuestro caso trabajamos con la versión 2017b, e instalar el Android Support from Simulink. También necesitamos un dispositivo Android con la versión de software Android 4.2 (JellyBean) o superior. Nosotros vamos a usar dos dispositivos Android para las pruebas, un Moto G de 2ª generación con Android 6.0 y un Sony Xperia X con Android 8.0.

3.1.3 Sensores y actuadores

A continuación, aparece una tabla comparativa de las especificaciones de los sensores que encontramos en ambos dispositivos.

Tabla 2: Comparativa móviles

	Moto G	Sony Xperia X
Fecha de lanzamiento	Septiembre 2014	Mayo 2016
Peso	149g	153g
Dimensiones	70.7 x 141.5 x 11 mm	69.4 x 142.7 x 7.9 mm

Procesador	Qualcomm Snapdragon 400 Quad-Core 1200 MHz	Qualcomm Snapdragon 650 Hexa-Core 1800 MHz
Memoria RAM	1GB LPDDR2	3GB LPDDR3
SO	Android 6.0 Marshmallow	Android 8.0 Oreo
Cámara	8 MPX f/2.0	23 MPX f/2.0
Giróscopo	BOSCH BMC150 Rango: 0-2000 rad/s Resolución: 0.003	BOSCH BMI160 Rango: 0-34.907 rad/s Resolución: 0.001
Acelerómetro	BOSCH BMC150 Rango: 0-156.88 m/s ² Resolución: 0.019	BOSCH BMI160 Rango: 0-78.453 m/s ² Resolución: 0.001
Magnetómetro	BOSCH BMC150 Rango: 0-1600μT Resolución: 0.3	AKM AK09915 Rango: 0-4912μT Resolución: 0.15
Proximidad	Sensortek STK3310 Rango: 0-100 cm Resolución: 100	Avago APDS-9940 Rango: 0-5 cm Resolución: 0.101
Luz	Sensortek STK3310 Rango: 0-27000 lux Resolución: 1	Avago APDS-9940 Rango: 0-30000 lux Resolución: 1
Presión	No	HSPPAD042A Rango: 0-1100 hPa Resolución: 1

En la figura inferior podemos observar los diferentes bloques que nos facilita Simulink para trabajar con estos sensores y otros para interactuar con la pantalla, visualizar contenido, reproducir o grabar audio, acceder a la cámara o establecer comunicaciones mediante red Wifi con otros dispositivos. A partir de la versión de Matlab R2018a también existe el bloque del magnetómetro, que nos dice el campo magnético en cada eje, y el bloque orientación, que mide la rotación del dispositivo respecto a los ejes X, Y y Z. También están los bloques ToApp y FromApp que nos permiten enviar o recibir datos de un método de una aplicación. Por último mediante el bloque ThingSpeak Write podemos publicar datos de nuestro equipo en internet creándonos una cuenta en su web. ThingSpeak es un servicio web para el Internet de las Cosas (IoT) que proporciona recolección de datos en tiempo real, procesamiento de datos, visualizaciones de estos, apps y plugins.

Simulink

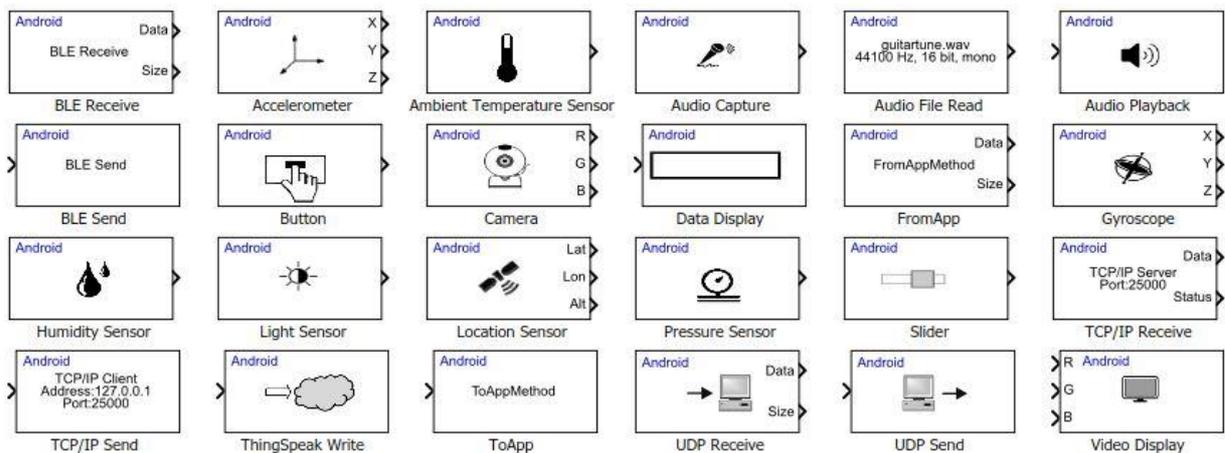


Figura 13: Sensores y actuadores

3.1.4 Comunicación

Para poder cargar una aplicación en nuestro Android primero activar las opciones de desarrollador en este, además debemos activar también la depuración por USB ya que el ordenador carga y ejecuta las aplicaciones en el dispositivo Android mediante la herramienta Android Debug Bridge y esta se comunica usando el cable USB. Para conectar el dispositivo a nuestro PC necesitamos que estén conectados mediante USB y que ambos estén conectados a la misma red Wifi, como comentamos en el apartado homónimo del toolbox del LEGO vamos a crear una zona de cobertura inalámbrica desde Windows 10 para conectar ahí los dispositivos por sencillez de configuración y mejorar la cobertura. Mediante el comando `targetupdate` en Matlab podemos establecer el link de comunicación siguiendo los pasos que salen para poder cargar las aplicaciones en el móvil. Estas aplicaciones una vez cargadas pueden ejecutarse desde el móvil cuando se desee sin necesidad de que exista conexión con el computador.

En el caso de este toolbox también se puede utilizar la opción de external mode para modificar parámetros y monitorizar el efecto de dichos cambios en tiempo real mientras se está ejecutando el modelo de Simulink en el dispositivo Android.

3.2 Ejemplos

3.2.1 Recepción y filtrado de señales

En este ejemplo vamos a ver cómo podemos ajustar parámetros y monitorizar las señales en tiempo real de un algoritmo ejecutándose en Android. En este caso vamos a modificar las ganancias de un filtro simple que va a suavizar la señal de salida de nuestro acelerómetro. Esto lo vamos a hacer usando el external mode, que nos permite cambiar algunos parámetros mientras el modelo está ejecutándose en nuestro dispositivo y podemos observar el efecto de estos cambios en la salida desde el ordenador.

Además del filtro que nos proporcionan en el ejemplo vamos a usar otro filtro paso bajo que incorpora Simulink y vamos a representar la salida sin filtrar para comparar los resultados.

Al usar los bloques scope, en el móvil vamos a ver las señales representadas gráficamente, podemos customizarlas a nuestro gusto y representar todas las señales que deseemos.

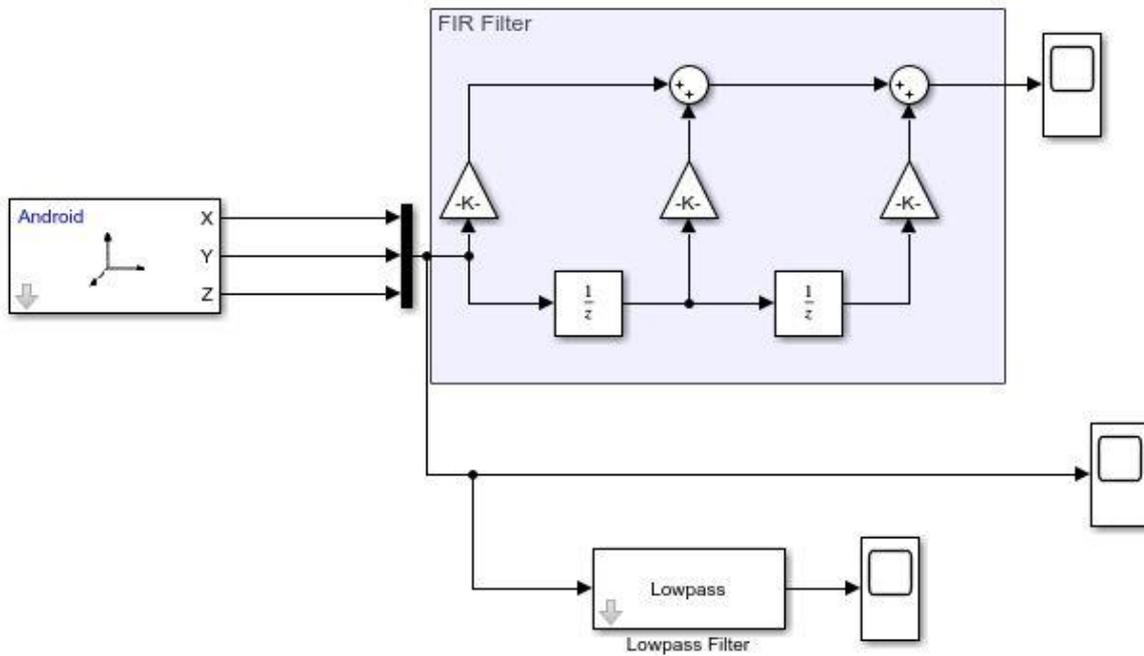


Figura 14: Filtros Android

Resultados

Podemos observar como la señal varía menos rápidamente y disminuye el ruido al ir variando las ganancias del filtro FIR. En la imagen inferior a la izquierda tenemos la salida de la señal en bruto, en el centro después de pasar por el filtro FIR y a la derecha después de pasar por el bloque del filtro paso bajo de Simulink.

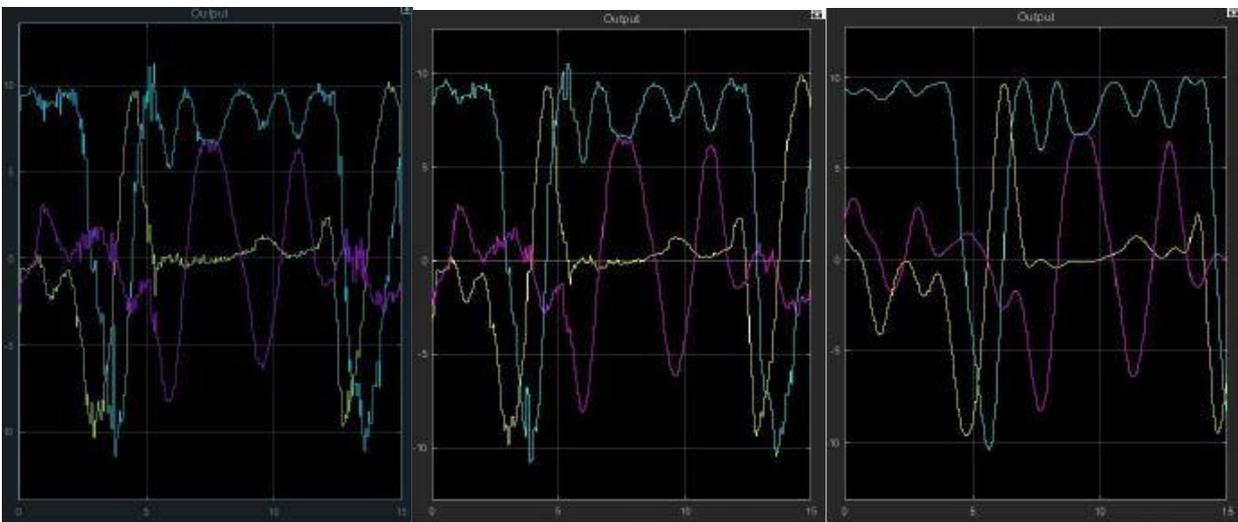


Figura 15: Acelerómetro filtros

El inconveniente del filtro paso bajo sería que ya hay un desfase de tiempo notable, aunque la señal sea mucho más suave. Dependiendo de la aplicación que para la que vayamos a usarlos puede interesarnos usar algún tipo de filtrado o no.

Modificaciones

Primero hemos probado cambiando el bloque del acelerómetro por el del giróscopo, que nos proporciona una señal más ruidosa como podemos ver en la imagen inferior. A la izquierda tenemos la salida de la señal en bruto, en el centro después de pasar por el filtro FIR y a la derecha después de pasar por el bloque del filtro paso bajo de Simulink.

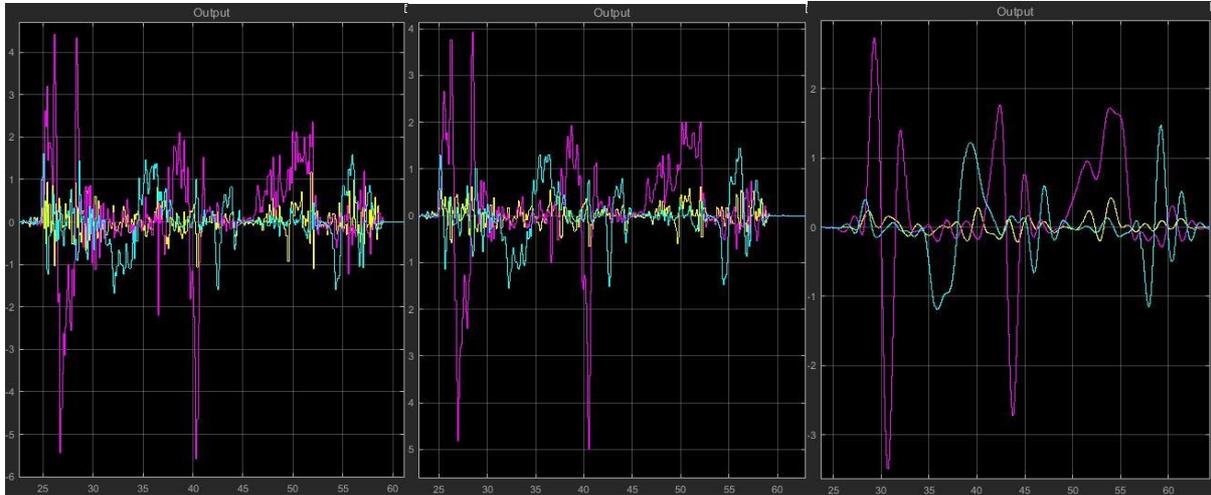


Figura 16: Giróscopo filtros

La segunda que hemos llevado a cabo ha sido comunicar al ordenador y el móvil sin necesidad del external mode. Se ha añadido un bloque UDP send del toolbox Android en las salidas que queremos recibir en nuestro ordenador, en el seleccionamos 255.255.255.255 como dirección IP remota para enviar a difusión y un puerto diferente para cada señal que queramos transmitir. Elegimos el modo de ejecución normal y pulsamos en Deploy to Hardware. Para la recepción en el ordenador crearíamos una sencilla aplicación como la mostrada en la imagen inferior:

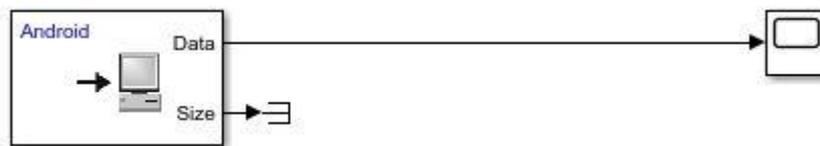


Figura 17: Receptor

El bloque usado sería el UDP Receive del toolbox de Android donde tenemos que configurar el puerto a escuchar el tipo de dato y el tamaño del vector. Para ejecutarlo seleccionamos el modo de ejecución a normal y pulsamos en Run, cuando la aplicación de nuestro Android se esté ejecutando en esta recibiremos los datos que hemos configurado.

3.2.2 Control EV3 mediante Android

En este ejemplo vamos a controlar el movimiento del robot LEGO mediante un dispositivo Android como si fuera un mando, usando los datos del acelerómetro de este. Ambos van a estar conectados a través de una red wifi.

Lo primero que hay que hacer es transformar matemáticamente los datos del acelerómetro a valores del pitch y roll para una determinada posición de Android. Después relacionamos estos con el movimiento del motor, el roll representaría velocidad hacia adelante y atrás y el pitch el movimiento hacia los lados. En la imagen inferior podemos ver los ejes X, Y y Z de nuestro dispositivo y los ángulos pitch y roll.

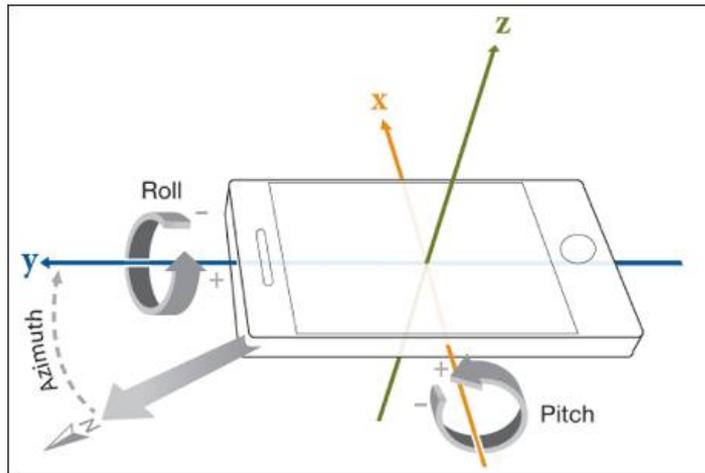


Figura 18: Ángulos pitch y roll

De esto se encargará una función de Matlab, que recibe como entrada la información del acelerómetro y como salida da la velocidad del motor izquierdo y derecho del robot. Estos datos junto a un botón de on/off los enviaremos mediante una conexión TCP/IP al LEGO. También vamos a recibir en nuestro dispositivo la información del sensor de ultrasonido del robot para ver la distancia a los objetos que tiene delante y generar una alarma si se encuentran a menos de 10 cm.

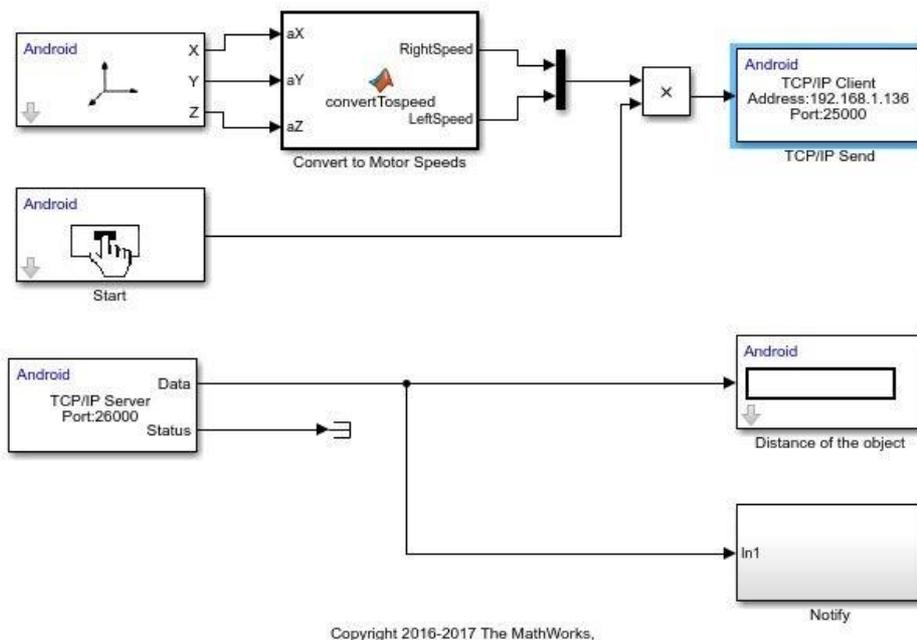


Figura 19: Control robot LEGO EV3 mediante Android

Simulink

En el lado del lego vamos a recibir las velocidades mediante la conexión TCP/IP para mandárselas al motor y vamos a transmitir la medida del sensor de ultrasonido.

El robot que vamos usar es el que construimos anteriormente. Para configurar los bloques de envío TCP/IP necesitamos saber las direcciones IP correspondientes y para la recepción necesitamos el tipo y tamaño de dato.

Resultados

Podemos observar como el robot se mueve acorde a la posición en la que se encuentra el móvil y reacciona de forma rápida a los cambios de esta.

4 CALIBRACIÓN, INTEGRACIÓN Y VALIDACIÓN EXPERIMENTAL

Podemos aprovechar los sensores extra que nos ofrecen los dispositivos Android para recibir más información del robot LEGO montando el móvil en este. Vamos a tener acceso a acelerómetro, giróscopo, cámara RGB, sensores de temperatura, humedad, presión y GPS. Estos datos podemos usarlos para hacer una realimentación en el control del robot o para recibir la telemetría de este.

4.1 Montaje dispositivo Android sobre EV3

Lo primero que vamos a hacer es añadir un soporte en la parte superior del robot para sujetar el móvil a la altura del eje de giro, así alineamos los ejes del robot con los del móvil para trabajar mejor con los datos de sus sensores.

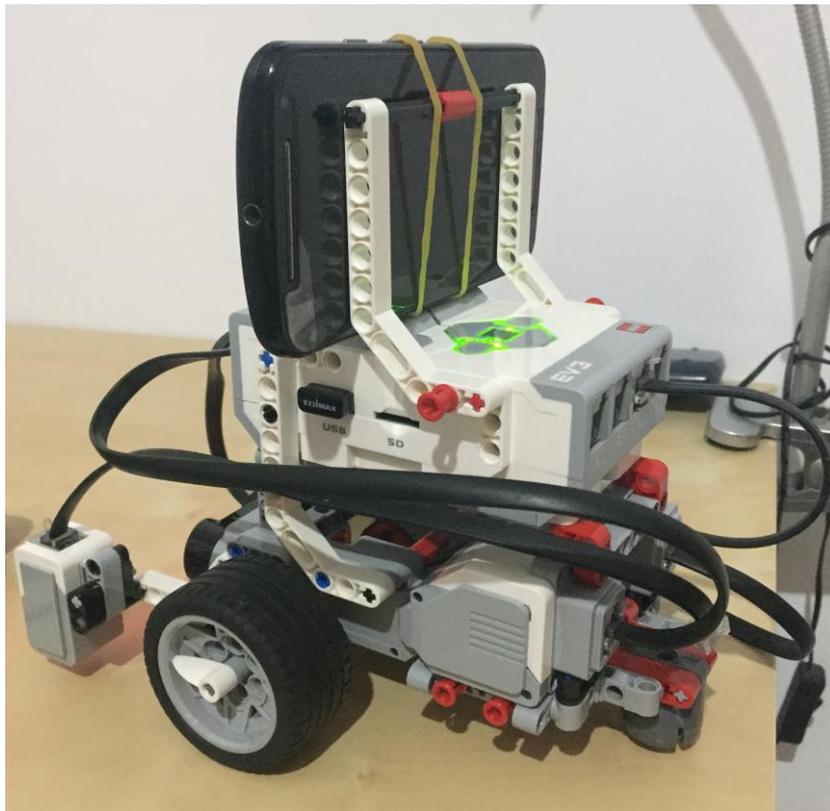


Figura 20: Móvil integrado con el robot

4.2 Telemetría robot

En este ejemplo vamos a transmitir los datos de los sensores que nos interesen a nuestro ordenador. Estas señales podemos procesarlas, representarlas o enviarlas al workspace de Matlab para trabajar con ellas después. Para el envío vamos a transmitir los datos de los sensores deseados, giróscopo y acelerómetro en este caso, mediante UDP a difusión. Esto lo vamos a hacer con dos móviles simultáneamente, por si queremos hacer una comparativa entre ambos, ejecutando una sencilla aplicación en cada uno.

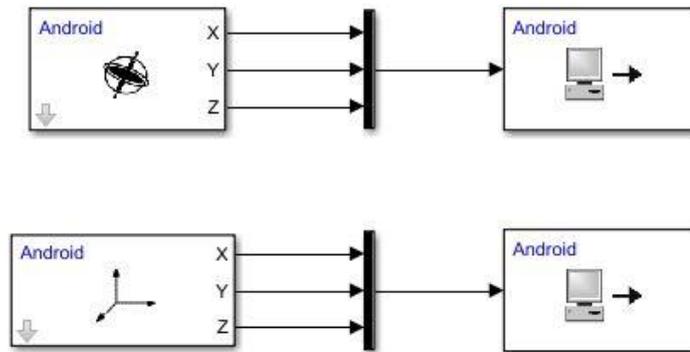


Figura 21: Envío Android

Desde el programa que estemos ejecutando en el robot también añadiremos un bloque que envíe los datos del giróscopo de este, recordemos que el este solo registra la velocidad angular en un eje. Para el programa que se encarga de la recepción de datos ejecutándose en el ordenador tenemos que tener en cuenta que los giróscopos de Android y el del robot miden la rotación en unidades diferentes así que vamos a convertir los primeros para comparar los resultados. Este sería el programa del ordenador, los bloques scope están configurados para que guarden una copia de los datos en el workspace de Matlab para futuros análisis.

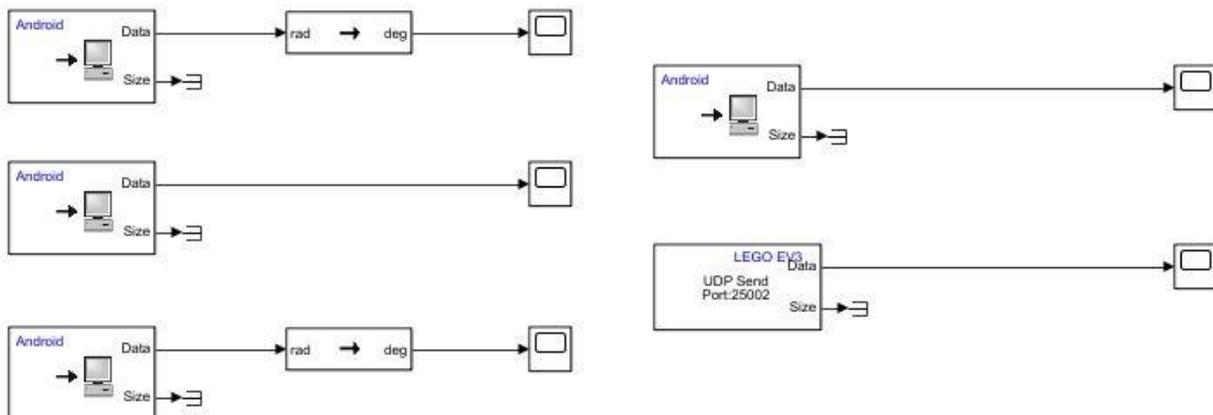


Figura 22: Receptor ordenador

Resultados

Primero hemos recibido la telemetría mientras el robot va siguiendo una línea. Para que no sea un lío de señales todas en la misma gráfica vamos a separarlas. En la imagen inferior tenemos los datos de los giróscopos, como los móviles nos dan la información de los tres ejes podemos observar como hay dos señales prácticamente a cero que no varía excepto por el ruido. También tenemos tres señales bastante similares que siguen los giros del robot a ambos lados, que son dos de los móviles y otra del giróscopo del LEGO.

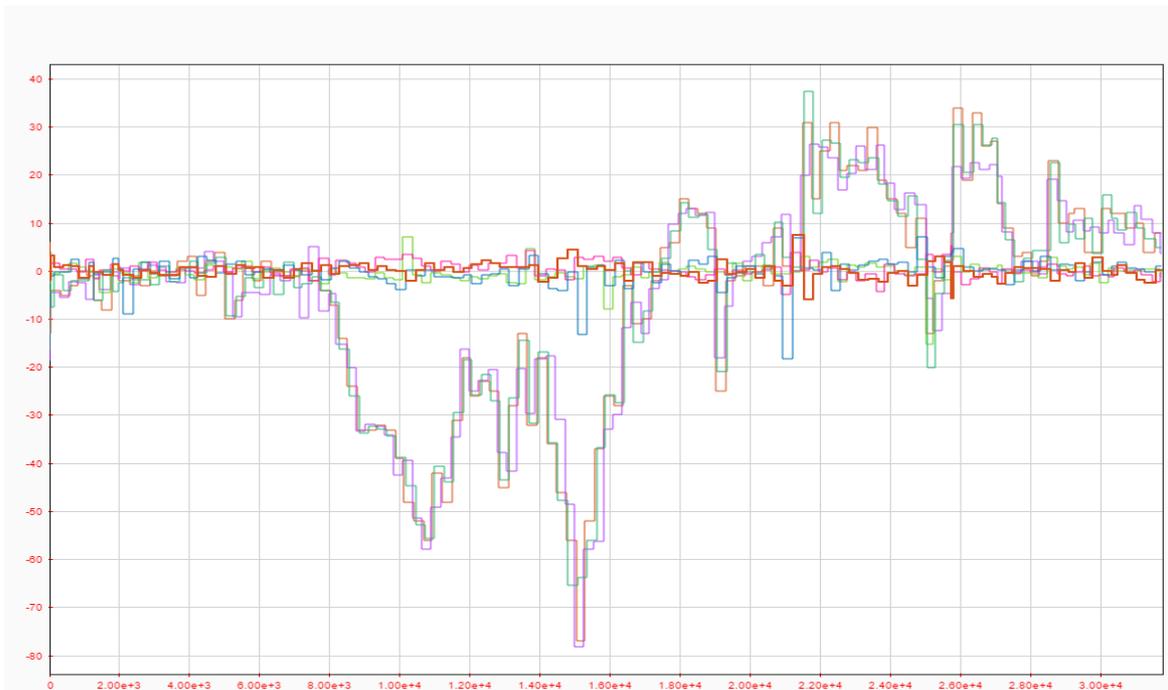


Figura 23: Telemetría giróscopo seguidor línea

Por otro lado, hemos recibido las aceleraciones en los tres ejes de cada Android. Estas, aunque siguen siendo similares, si difieren algo más. Por ejemplo, en su valor medio, la señal del Xperia se acerca más a los 9.8 de la gravedad.

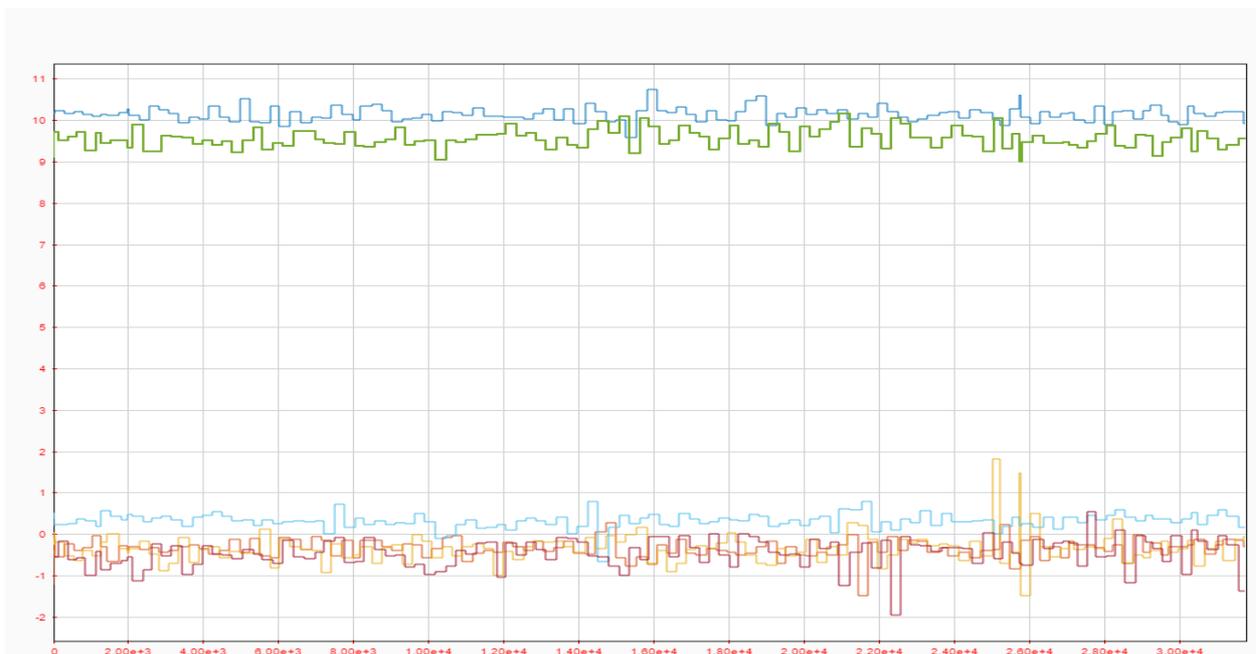


Figura 24: Telemetría acelerómetro seguidor línea

También se han hecho pruebas girando el robot sobre sí mismo a distintas velocidades y en ambas direcciones. En los datos de los giróscopos de la siguiente imagen volvemos a ver que las tres señales son muy parejas y vemos una relación entre la velocidad de giro y el ruido de las señales que deberían estar a cero.

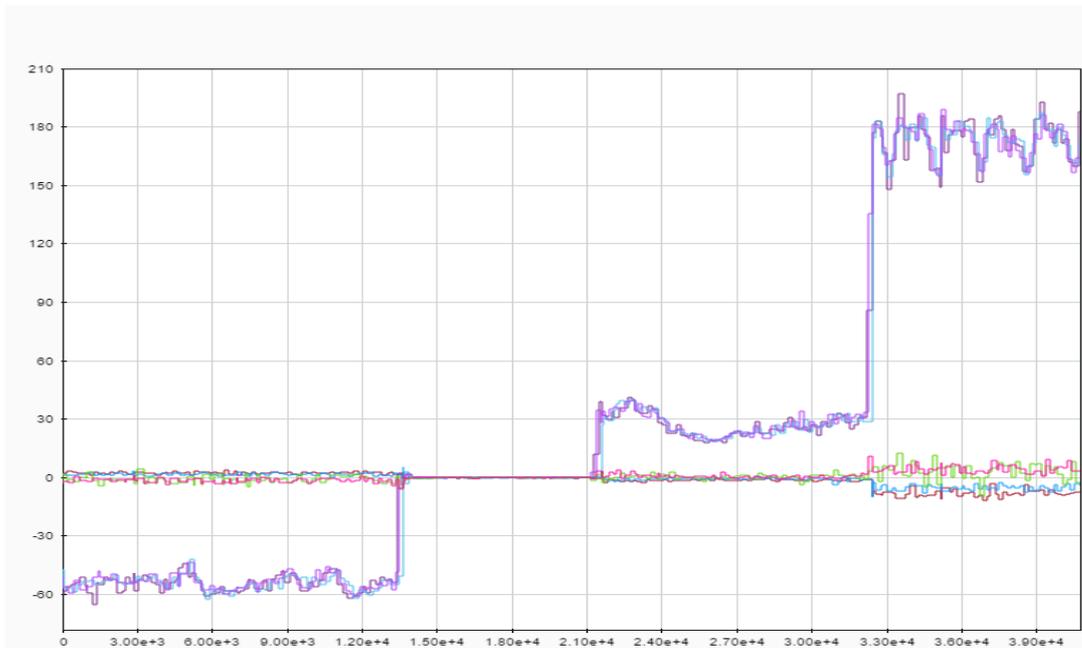


Figura 25: Giro ambas direcciones giróscopo

Estos cambios de velocidades de giro y su influencia en la amplificación del ruido se aprecian con claridad en los acelerómetros también.



Figura 26: Giro ambas direcciones acelerómetro

4.3 Calibración sensores

Para comprobar la calibración de los sensores hemos creado una aplicación que hace que el robot gire sobre sí 90 grados. Para la observación de este giro tomamos líneas perpendiculares en el suelo como referencia. Una vez implementado en el LEGO colocamos el móvil arriba de este y ejecutamos unos cuantos de giros. En la imagen inferior podemos observar como los tres giróscopos han captado la información de tres giros.



Figura 27: Giros 90°

Esto realmente lo que está midiendo es la velocidad angular del giro en grados/segundos. Si queremos saber los grados que llevamos girados los podemos aproximar integrando la señal del sensor que queramos comprobar. Abajo vemos el resultado de haber hecho un par de giros.

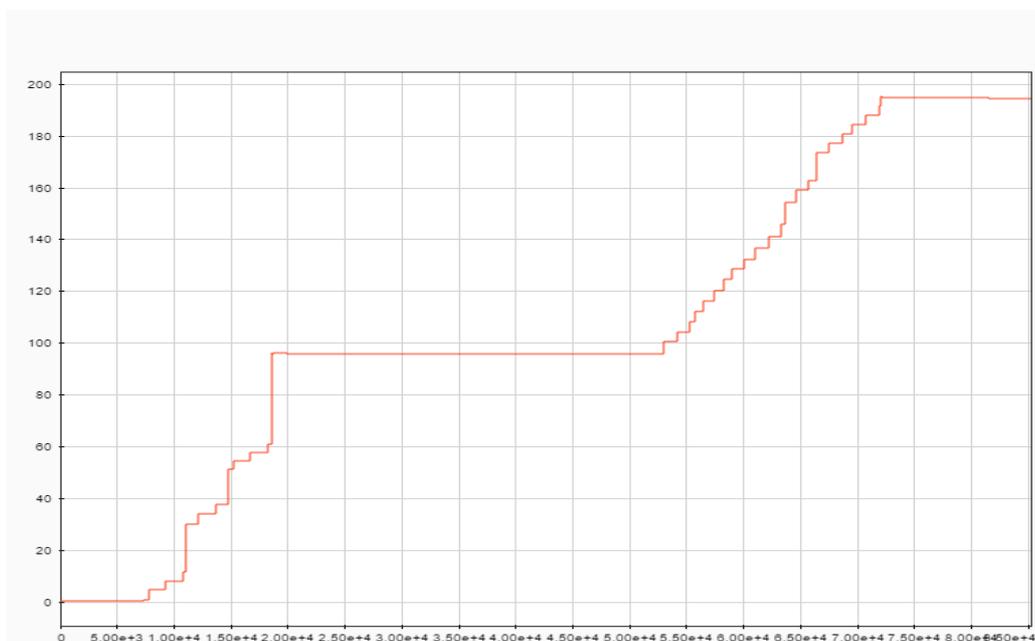


Figura 28: Calibración giro 90°

Existe un error de unos 5° debido a la forma en que calculamos ese ángulo ya que con la integración se acumulan los errores y el ruido de la medición. Para mediciones que necesiten ser más exactas se mezclan los datos de giroscopos y acelerómetros ya que los primeros funcionan muy bien para movimientos cortos o bruscos y los segundos no tienen deriva a medio o largo plazo, pero no son fiables a corto. Esto se suele conseguir implementando un filtro de Kalman, o de manera más sencilla con un filtro complementario.

4.4 Control EV3 mediante información sensores Android

En este apartado queremos comprobar que se puede controlar un proceso en el LEGO realimentándolo con la información de los sensores del móvil. Esto lo podemos conseguir transmitiendo datos de sus sensores entre ellos o transmitiendo variables auxiliares que se activan bajo determinadas condiciones. Queremos conseguir que el robot gire un determinado ángulo, pero en vez de sacar esa información de los encoder de los motores como se usó en el apartado anterior vamos a usar el giroscopo de Android. El móvil lo colocaremos encima del LEGO como ya hemos hecho en otros ejemplos, este calculará el ángulo que lleva girado y cuando llegue al ángulo deseado activará una señal que se le envía al EV3 mediante TCP/IP.

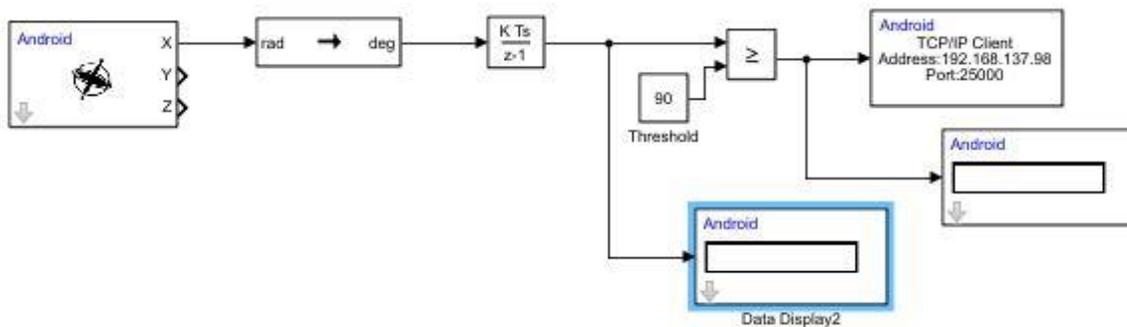


Figura 29: Control Android

El robot por su parte solo tiene que estar girando hasta que reciba esa señal activada y se pare.

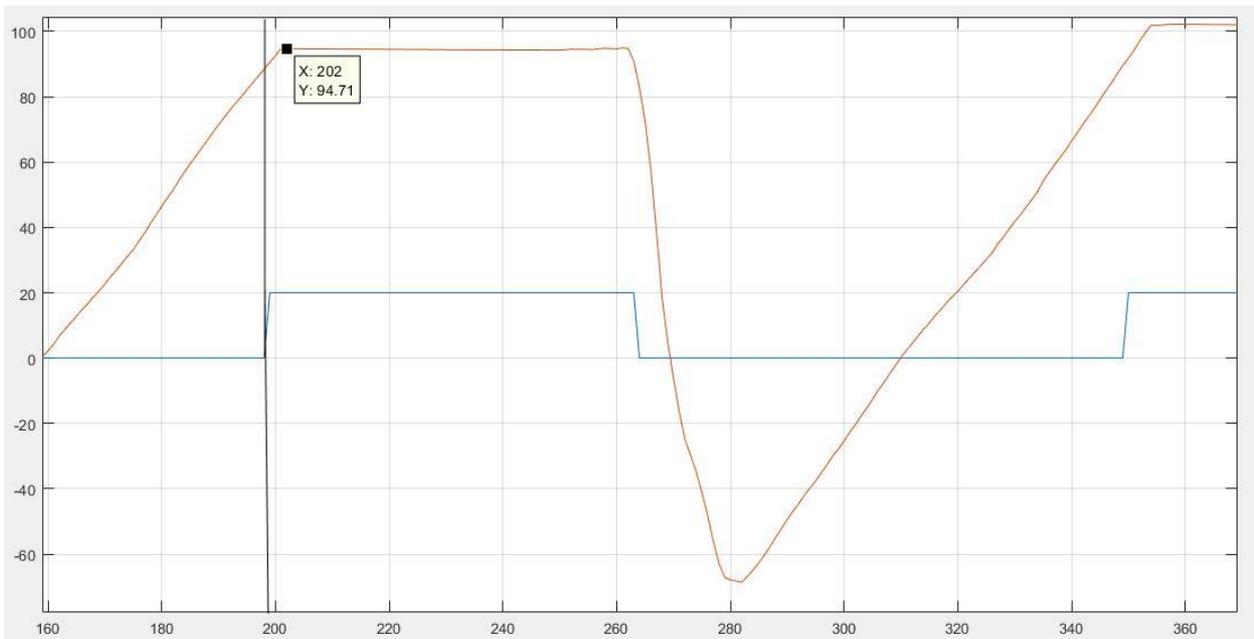


Figura 30: Control giro mediante Android

Simulink

Como se puede apreciar en la imagen anterior existe un cierto desfase entre que la señal se activa y el robot se para, por lo tanto, al final giramos en torno a 95° . Este desfase se debe a la comunicación por Wifi entre ambos equipos. También hay que tener en cuenta que el cálculo del ángulo girado se ha obtenido mediante la integración de los datos del giróscopo por lo tanto no es totalmente exacto.

5 CONCLUSIONES

Hemos observado como el toolbox del LEGO MINDSTORMS EV3 nos ayuda a crear diferentes aplicaciones de control, visualización y transmisión de datos de nuestro robot. Por su parte el toolbox de Android nos permite trabajar de forma sencilla con los sensores de nuestro móvil, procesar estos datos y enviarlos a otros dispositivos. Gracias a esto podemos ampliar los sensores con los que trabaja el LEGO, incrementando así la variedad de aplicaciones que se pueden diseñar para este. Esto, junto con su reducido precio comparado con otras, hace que la plataforma LEGO MINDSTORMS sea una opción interesante para usarse con fines docentes en disciplinas relacionadas con la ingeniería.

BIBLIOGRAFÍA

1. EV3 user guide
2. <https://es.mathworks.com/hardware-support/lego-mindstorms-ev3-simulink.html>
3. <https://www.lego.com/en-us/mindstorms/support>
4. <http://thetechnicgear.com/category/mindstorms/lego-mindstorms-tutorials-howto/>
5. <https://www.ev3dev.org/>
6. https://es.wikipedia.org/wiki/Set_de_Lego_mindstorms_EV3
7. Building Instructions for Robot Educator driving base
8. http://www.inpharmix.com/jps/PID_Controller_For_Lego_Mindstorms_Robots.html
9. https://www.mathworks.com/examples/simulink/mw/legomindstormsev3_product-ev3_linetracking-line-tracking
10. https://www.mathworks.com/examples/simulink/mw/legomindstormsev3_product-ev3_robot_hostcontrol-controlling-lego-mindstorms-ev3-robot-from-host-computer
11. <https://es.mathworks.com/products/stateflow.html>
12. <https://es.mathworks.com/hardware-support/android-programming-simulink.html>
13. <http://specdevice.com/?lang=es>
14. <https://thingspeak.com/>
15. <https://es.mathworks.com/matlabcentral/fileexchange/?term=tag%3A%22android%22>
16. <https://es.mathworks.com/help/supportpkg/android/ug/tune-parameters-and-monitor-data-in-a-model-running-on-target-hardware.html>
17. <https://es.mathworks.com/help/supportpkg/android/examples/communicating-with-android-devices.html>
18. <https://es.mathworks.com/help/supportpkg/android/examples/working-with-android-devices-block-library.html>
19. <https://es.mathworks.com/help/supportpkg/android/examples/scope-with-android-devices.html>
20. <https://es.mathworks.com/help/supportpkg/android/examples/control-lego-mindstorms-ev3-robot-using-android-device.html>

