

Manual de usuario para el entorno ROS-MATLAB-Arduino

Documento Anexo B

Ernesto David CORTÉS GARCÍA

Universidad Nacional de Colombia, Departamento de Mecánica y Me-

catrónica, Colombia

erdcortesga@unal.edu.co



Introducción

Este manual de usuario es documento anexo a la tesis *Experimentación del control de actitud en un prototipo de CubeSat con ruedas de reacción*[1]. El cual sirve de guía de usuario para el uso del entorno ROS - MATLAB - Arduino, para la implementación de controladores en tiempo real.

Entorno ROS-MATLAB-Arduino

- Explicación de las partes
- **ROS**(Robot Operating System): Hace posible la lectura de los datos de la IMU Razor en tiempo real y los arroja en Cuaterniones. ROS solo se encuentra disponible en el sistema operativo (OS) de Linux.
 - **Arduino**: Envía la señal de control al ESC que hará mover los motores.
 - **MATLAB - Simulink**: Ejecuta el controlador deseado. Debe conectarse con la IMU para leer el estado del sistema y con Arduino para ejecutar el movimiento de los actuadores.

Configuración IMU - ROS

Para acoplar el sensor IMU Razor a ROS es necesario seguir el paso a paso proporcionado por la guía oficial de ROS.

Para verificar el funcionamiento basta con ejecutar un terminal con dos pestañas. En la primera pestaña ejecutar el código:

```
1 roscore
```

En la segunda ejecutar el código:

```
1 roslaunch razor_imu_9dof razor-pub-and-display.launch
```

Deben aparecer dos ventanas emergentes con la siguiente visualización:

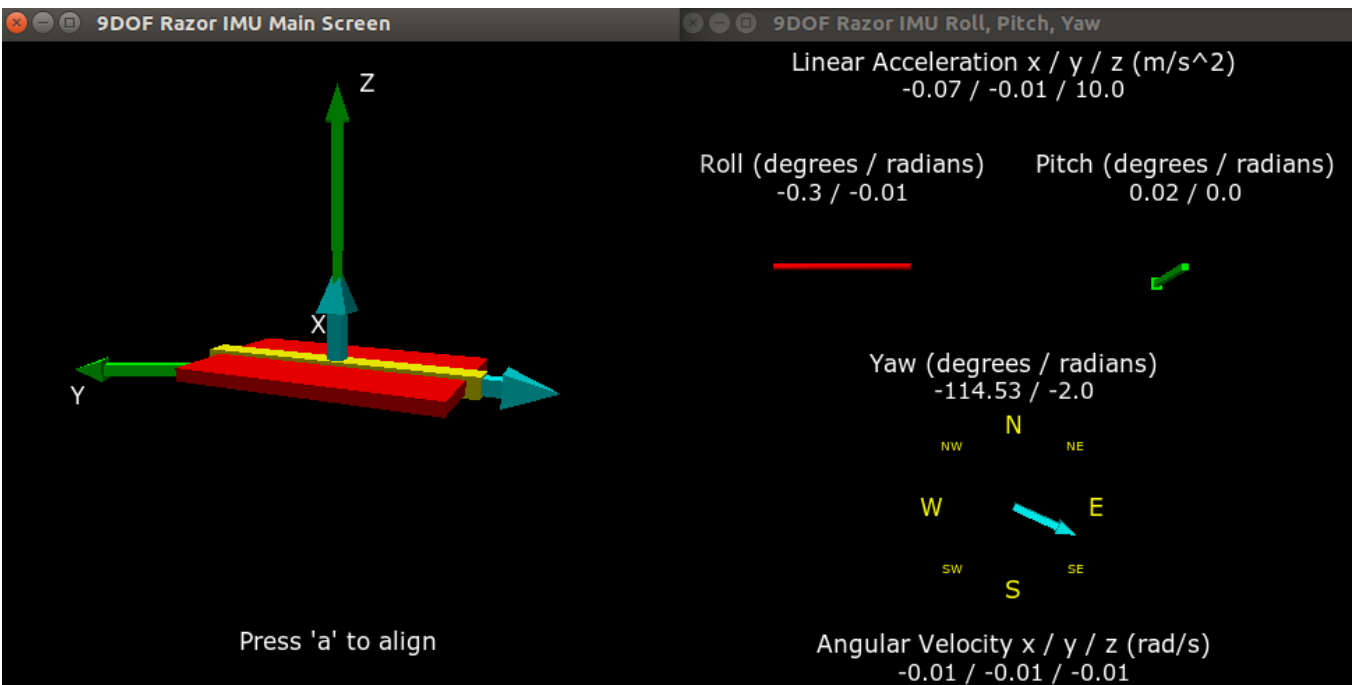


Figura 1: Interfaz IMU Razor - ROS

Configuración Simulink

Para abrir el entorno de simulink en MATLAB basta con escribir el siguiente comando en la ventana de comandos (*Command Window*):

```
1 simulink
```

Antes de empezar las funciones de simulink es necesario configurar el entorno de trabajo:

1. Una vez abierta la ventana de simulink, se crea un nuevo modelo (*Blank Model*). En la sección superior seleccionar el botón con forma de engranaje (*Model Configuration Parameters*) para configurar el entorno así como el *solver* de este.

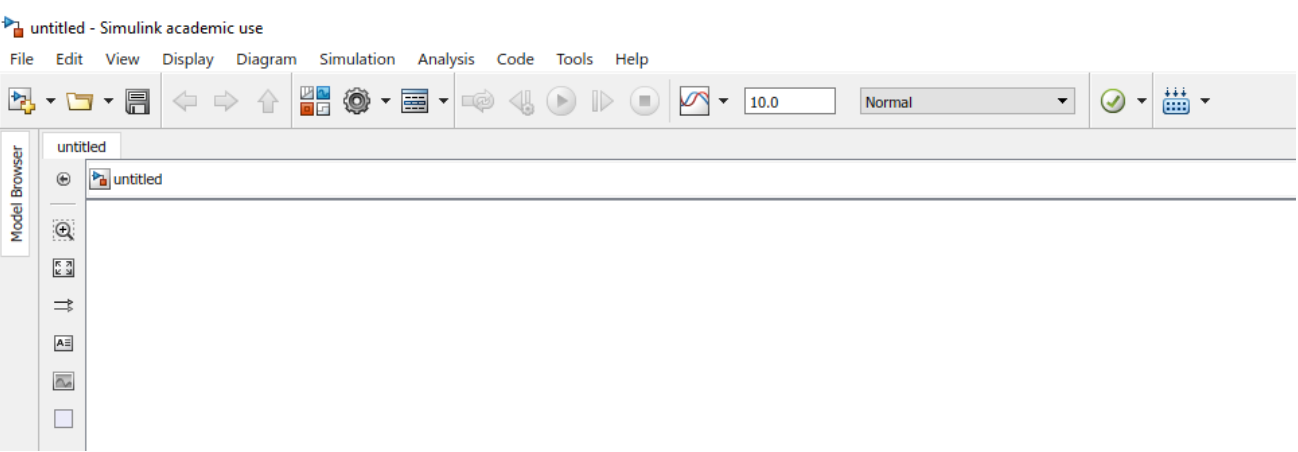


Figura 2: Entorno Simulink

2. En la ventana emergente cambiar en la sección de *Solver Selection* el modo de "Variable-Step" a "Fixed Step". Se recomienda el uso del solver de orden 4 o Runge-Kutta. Asimismo en *Solver Details* se recomienda cambiar el tiempo (*Step time*) de Auto a 0.01. Para casos de simulación en tiempo real es necesario cambiar el tiempo de parada de 10.0 a *inf*.

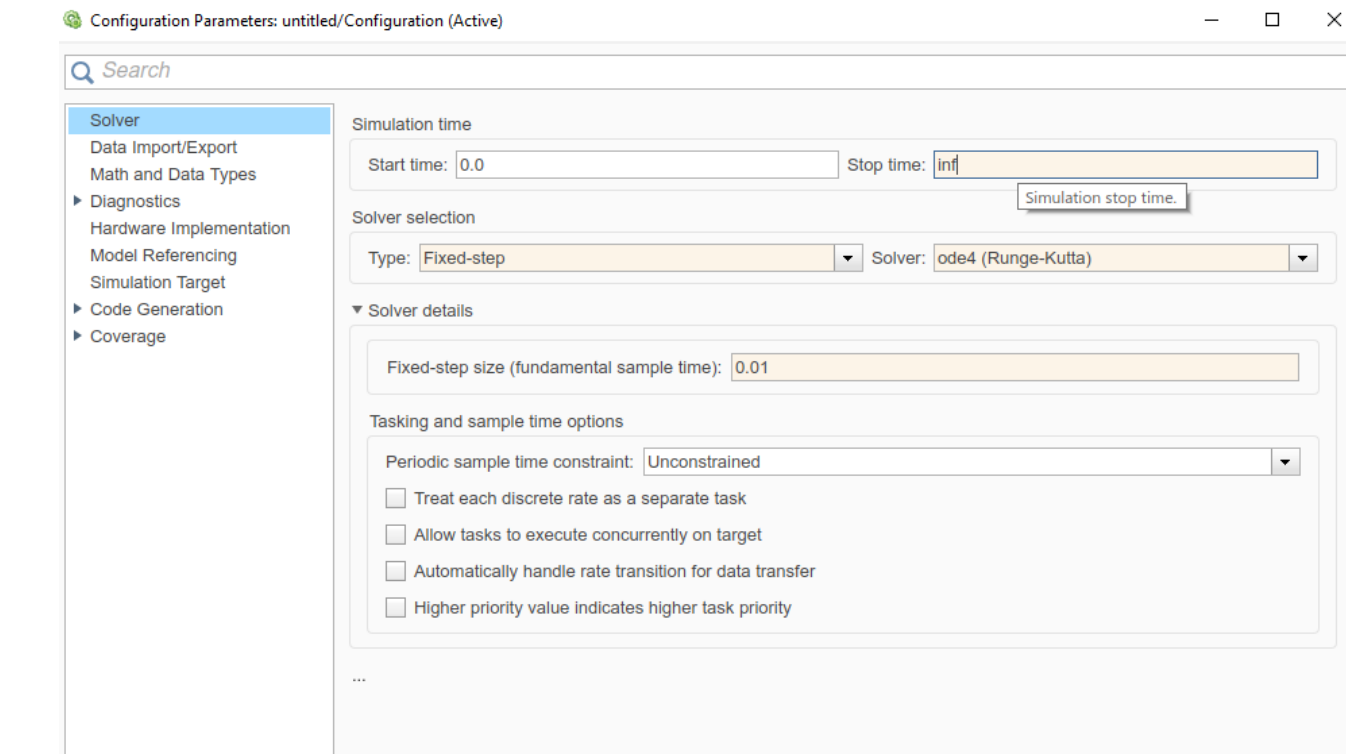


Figura 3: Entorno Simulink - Model Configuration Parameters

Una vez realizada esta configuración es posible empezar a usar e implementar las funciones que ofrece MATLAB - Simulink. Para ejecutar el programa basta con seleccionar el boton de play verde(*Run*) en la sección superior.

Configuración Arduino - Simulink

Para poder acoplar Arduino a Simulink es necesario instalar los siguientes paquetes disponibles en la *tienda de MATLAB (Add-ons)*.

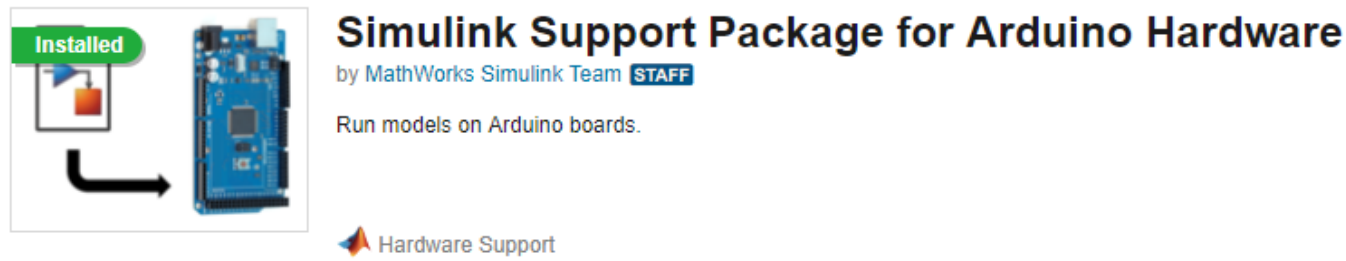


Figura 4: Paquetes de MATLAB para conectarse con arduino a)Simulink Support Package for Arduino Hardware



Figura 5: Paquetes de MATLAB para conectarse con arduino b)Matlab Support Package for Arduino Hardware

Una vez instalado, se verifica que en la librería (*Library Browser*, junto al engranaje), debe crearse una categoría (en el panel izquierdo) especial para Arduino. Simulink Support Package for Arduino Hardware. El bloque de PWM es capaz de mover los motores BLDC.

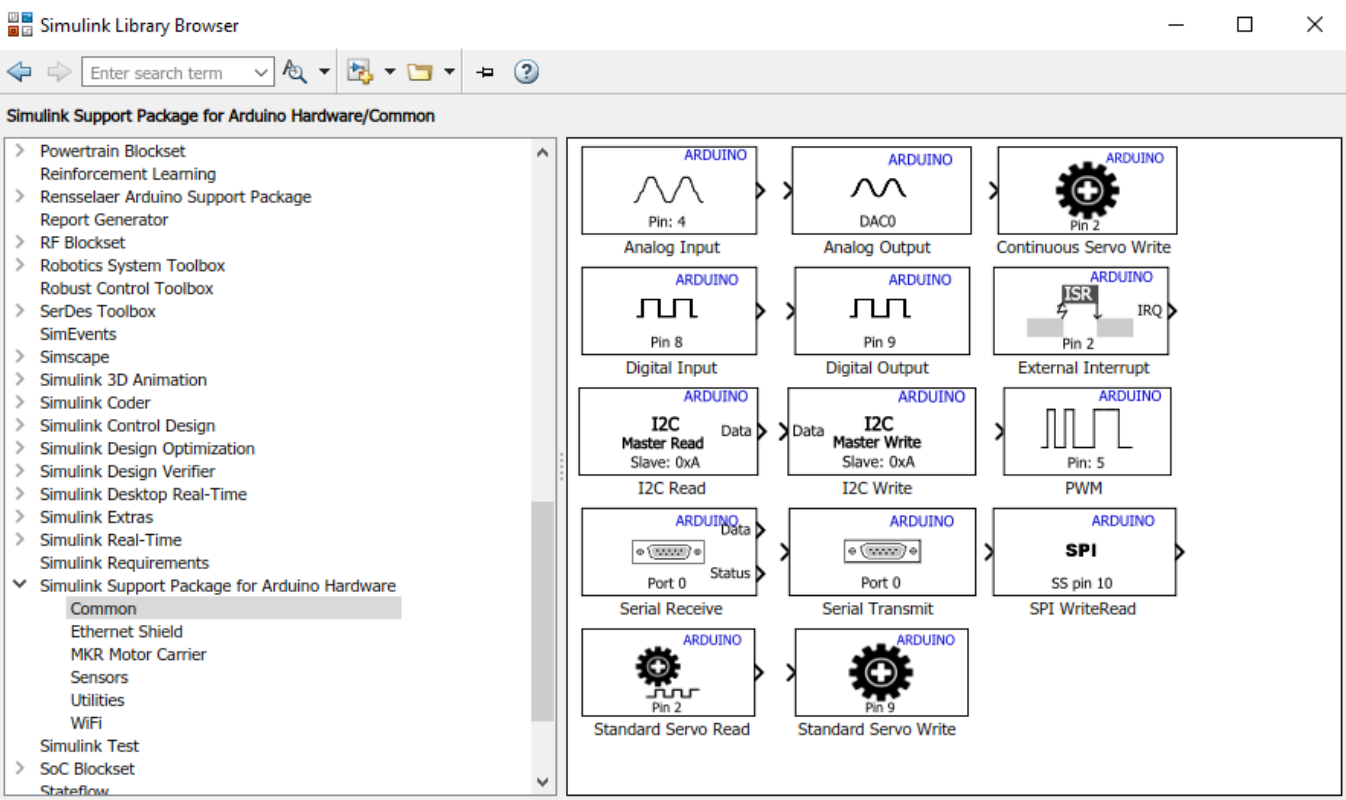


Figura 6: Entorno Simulink - Library Browser

Es necesario también configurar la tarjeta de Arduino que se espera usar para la implementación. Nuevamente en el engranaje, se selecciona la opción "Hardware Implementation" de la columna izquierda. En *Hardware Board* se escoge la tarjeta a implementar. En *Device Details* se recomienda indicar manualmente el puerto del dispositivo "dev/ttyACM*"

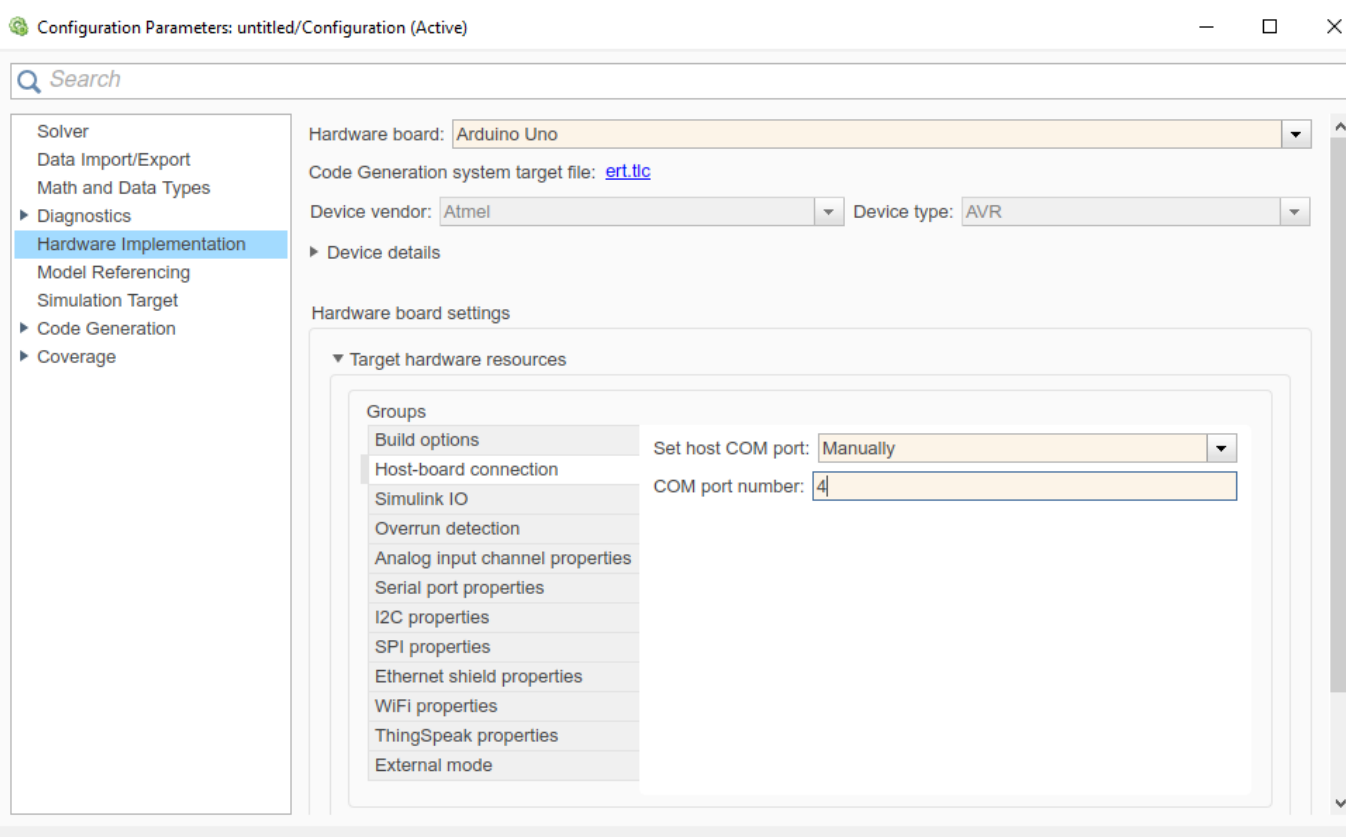


Figura 7: Entorno Simulink - Hardware Implementation

Configuración ROS - Simulink

De igual manera deben instalarse dos paquetes para el funcionamiento de ROS en Simulink.

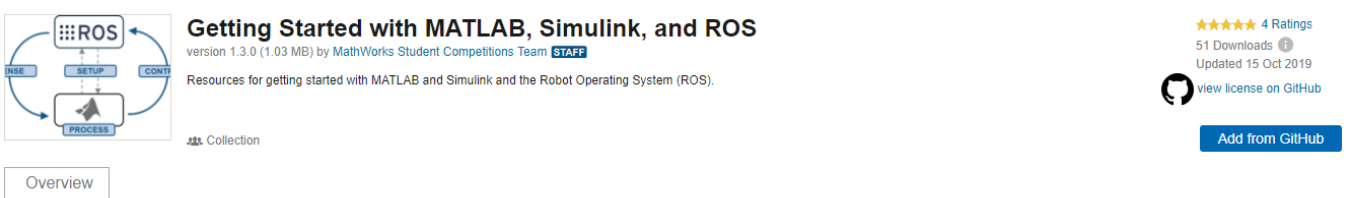


Figura 8: Paquetes de MATLAB para conectarse con ROS a) Getting Started with MATLAB, Simulink, and ROS

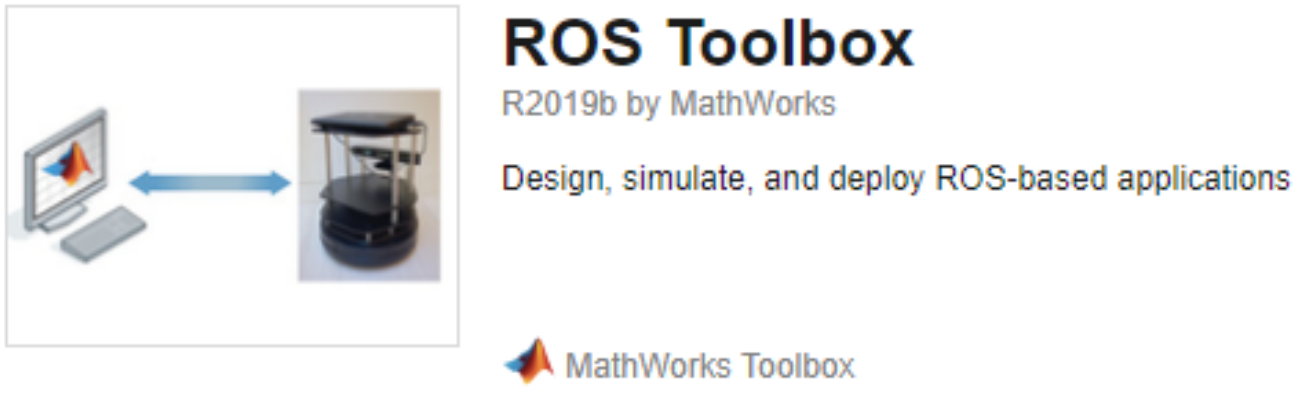


Figura 9: Paquetes de MATLAB para conectarse con ROS b)ROS Toolbox

Se verifica nuevamente mediante la librería. Se puede disponer del siguiente diagrama de bloques

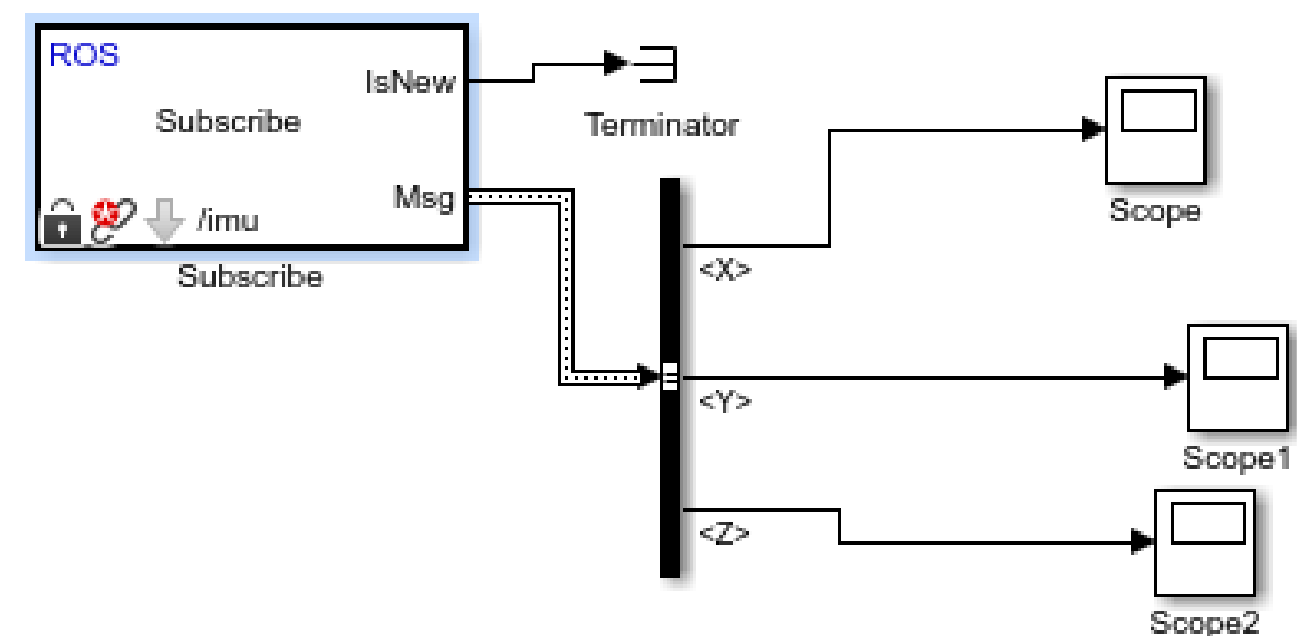


Figura 10: Verificación diagrama de bloques ROS-Simulink

Configurando el diagrama de bloques de Subscribe y el bus de la siguiente forma:

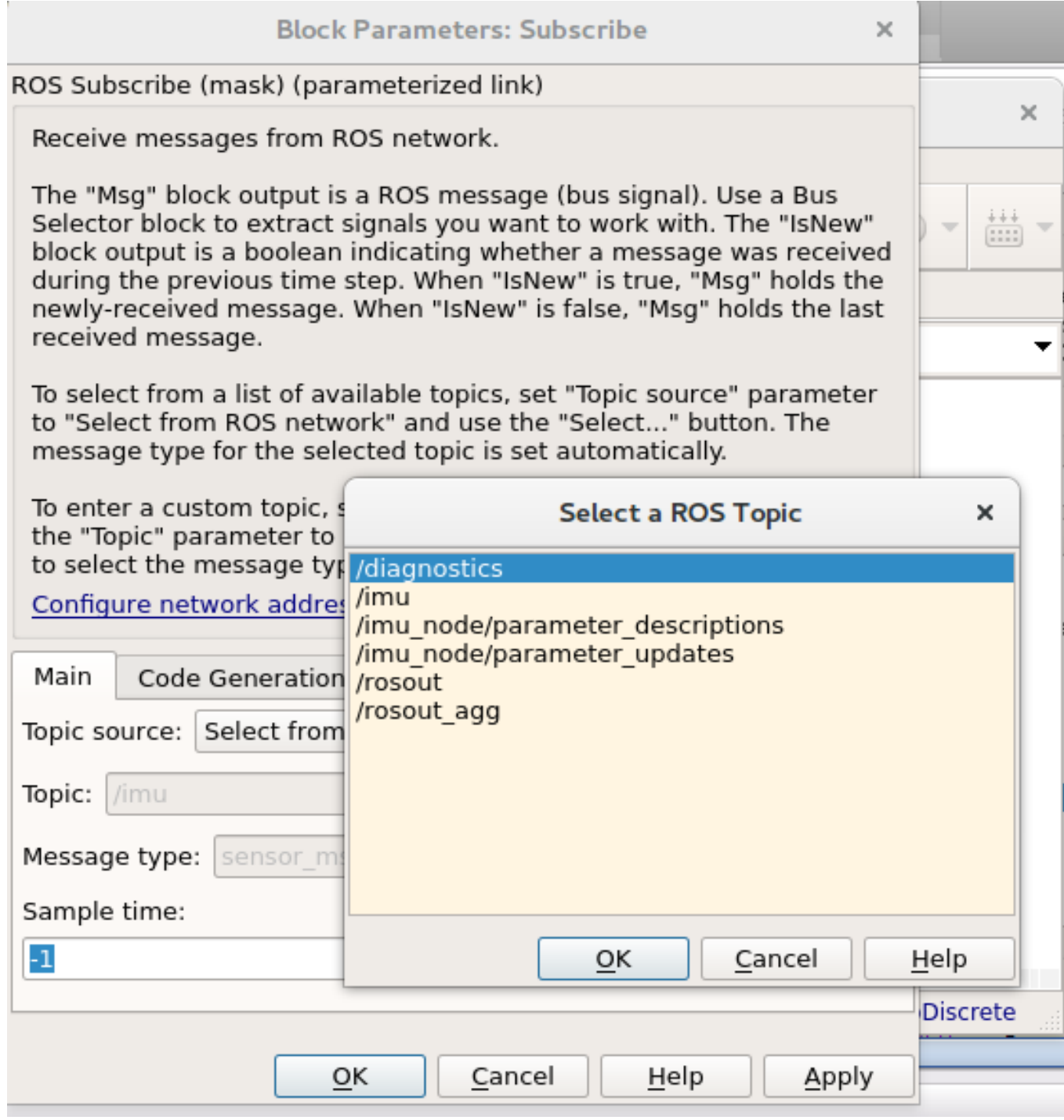


Figura 11: Subscribe

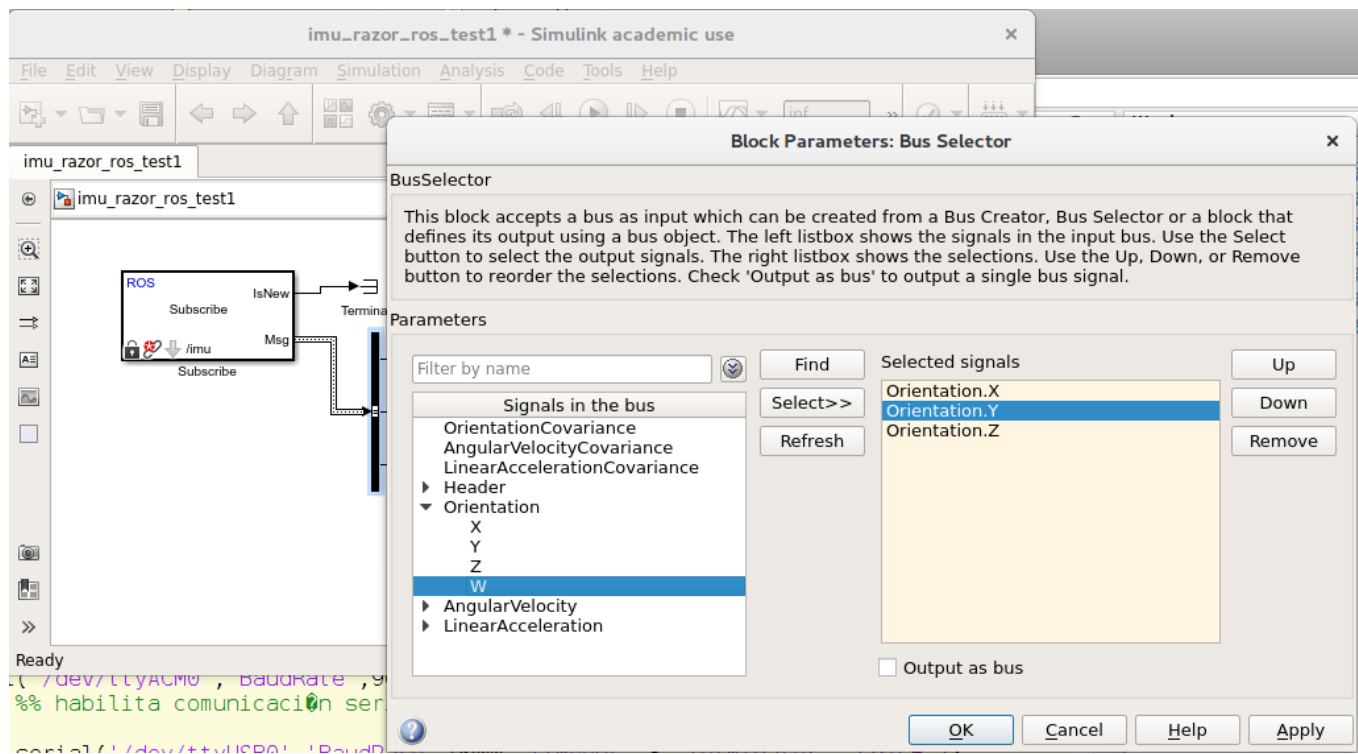


Figura 12: Bus de datos

Se observan los resultados en tiempo real una vez se corra el programa.

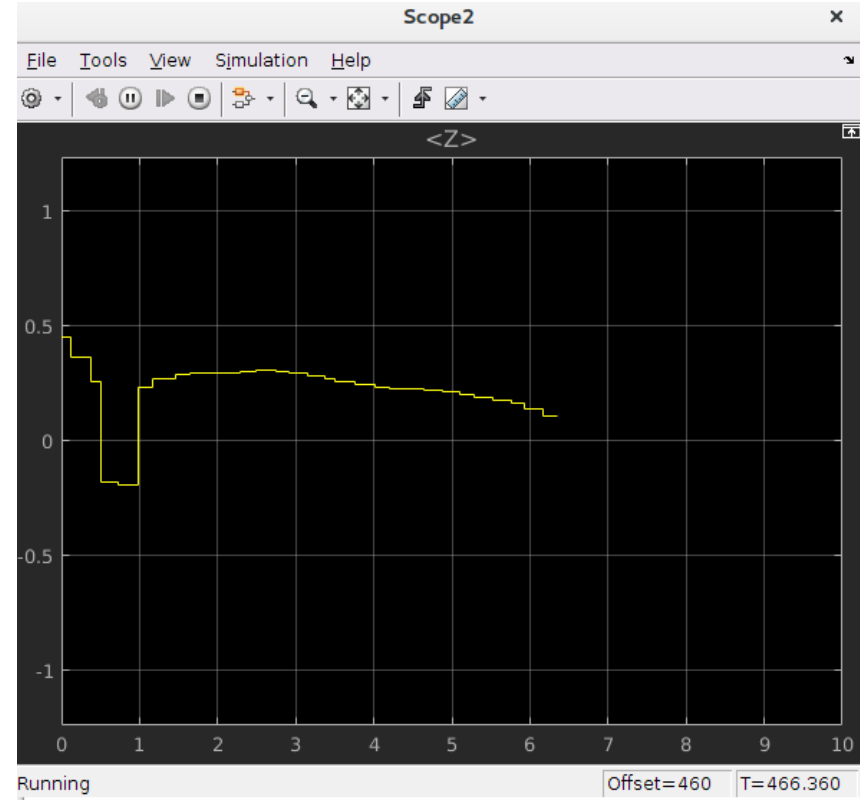


Figura 13: Observación en tiempo real del eje Z de la IMU

Referencias

[1] Ernesto David Cortés García. *Experimentación del control de actitud en un prototipo de CubeSat con ruedas de reacción*. PhD thesis, Univerisdad Nacional de Colombia, 2019.