



## Documentación Interfaz Rover Robocol

**Autor:**

**ROBOCOL**

**Profesores Asesores:**

Carlos Francisco Rodríguez - Departamento de Ingeniería Mecánica

Fredy Enrique Segura - Departamento de Ingeniería Electrónica

12 de mayo de 2020

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Herramientas Necesarias para correr el servidor</b>	<b>3</b>
2.1. Herramientas . . . . .	3
<b>3. Ventajas y Desventajas de la interfaz</b>	<b>4</b>
3.1. Ventajas . . . . .	4
3.2. Desventajas . . . . .	4
<b>4. Trabajo prioritario a realizar</b>	<b>4</b>
<b>5. Partes de la Interfaz</b>	<b>5</b>
5.1. Pestaña de Tracción . . . . .	5
5.2. Pestaña de Autonomía . . . . .	6
5.3. Pestaña de Brazo Robótico . . . . .	7
5.4. Pestaña de Sensórica . . . . .	8
5.5. Pestaña de Estatus del Rover . . . . .	9

## 1. Introducción

El siguiente documento muestra la interfaz actual, dando una breve explicación de cada una de la pestañas de la misma y explicando sus funciones, sus fallas y las partes que están inhabilitadas. Esta interfaz se desarrolló en Django y luego se incluyó como un nodo de ROS y es la mayor encargada de la interacción con el rover. Aunque cuenta de muchas funcionalidades aún carece de muchas otras que son esenciales. Requiere cambios en tanto agregar, eliminar cosas y de organización.

Entre las ventajas de esta interfaz está que al ser una página web es accesible fácilmente a través de cualquier dispositivo después de que el servidor está corriendo y de poder acceder desde varios dispositivos al mismo tiempo.

## 2. Herramientas Necesarias para correr el servidor

Para correr la interfaz el servidor de la misma requiere tener las siguientes herramientas instaladas y funcionales, es necesario revisar si hay algún requerimiento adicional que no esté mencionado. Los clientes solo requieren conectarse a la red y poder solicitar un servicio HTTP (a través de un navegador comúnmente) colocando la dirección IP del servidor con su respectivo puerto.

### 2.1. Herramientas

- ROS (Melodic recomendado)
  - <http://wiki.ros.org/melodic>
- Django Channels (versión ....)
  - <https://channels.readthedocs.io/en/latest/>
- Simple JSON
  - <https://simplejson.readthedocs.io/en/latest/>
- Numpy
  - <https://numpy.org/>
- Redis
  - <https://redis.io/>

### 3. Ventajas y Desventajas de la interfaz

#### 3.1. Ventajas

- Una vez que el servidor sube la interfaz es muy fácil acceder desde cualquier dispositivo sin necesidad de instalar ninguna herramienta adicional (solo se requiere conexión a una red y un navegador)
- Reúne varias funcionalidades del rover de una manera ordenada.
- Pueden entrar varios usuarios al tiempo y usar diferentes secciones (tracción, brazo, etc)
- Es muy veloz, fácil de recargar.

#### 3.2. Desventajas

- Se requiere gran cantidad de herramientas en el servidor para correr la interfaz.
- Se requiere un gran conocimiento de programación (Python, Django, Web, ROS) para entenderla y modificarla adecuadamente.
- Actualmente es completamente dependiente de una red TCP/IP (en el caso de manejo remoto del rover, una red WiFi).

### 4. Trabajo prioritario a realizar

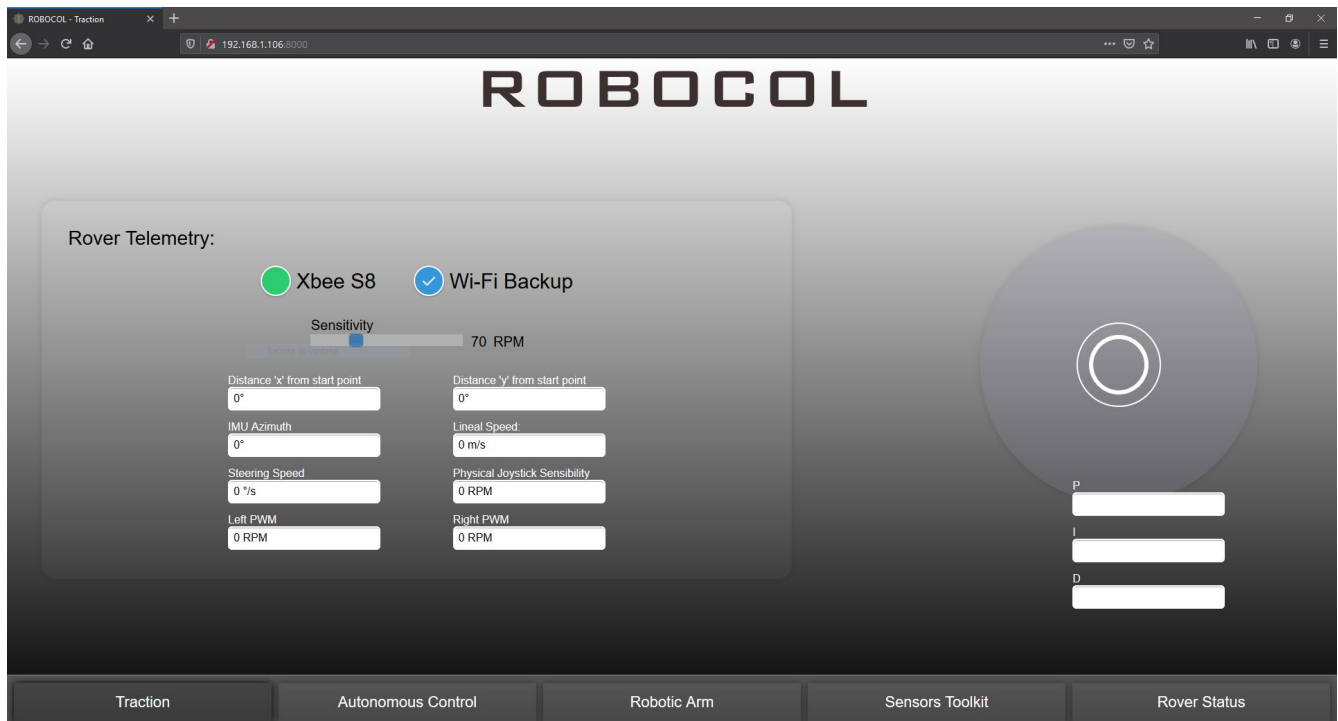
Entre las cosas que se requieren realizar con urgencia en la interfaz están las siguiente

- Conectarse con la base datos (a definir) del subsistema de sensórica y agregar los elementos necesarios para este subsistema, así como organizar la pestaña de la mejor manera con la información actual y la nueva.
- Permitir que la página sea responsive de manera que se pueda abrir en una tablet o celular de manera cómoda y usar toda su funcionalidad.
- Revisar las funciones inactivas y removerlas, arreglarlas o terminarlas según sea el caso.
- Hablar con el equipo de visión y definir si la interfaz tendrá cámaras, o al menos un acceso a ellas.
- Hablar con el equipo de tracción y de brazo para añadir las funcionalidades faltantes a la interfaz.

## 5. Partes de la Interfaz

### 5.1. Pestaña de Tracción

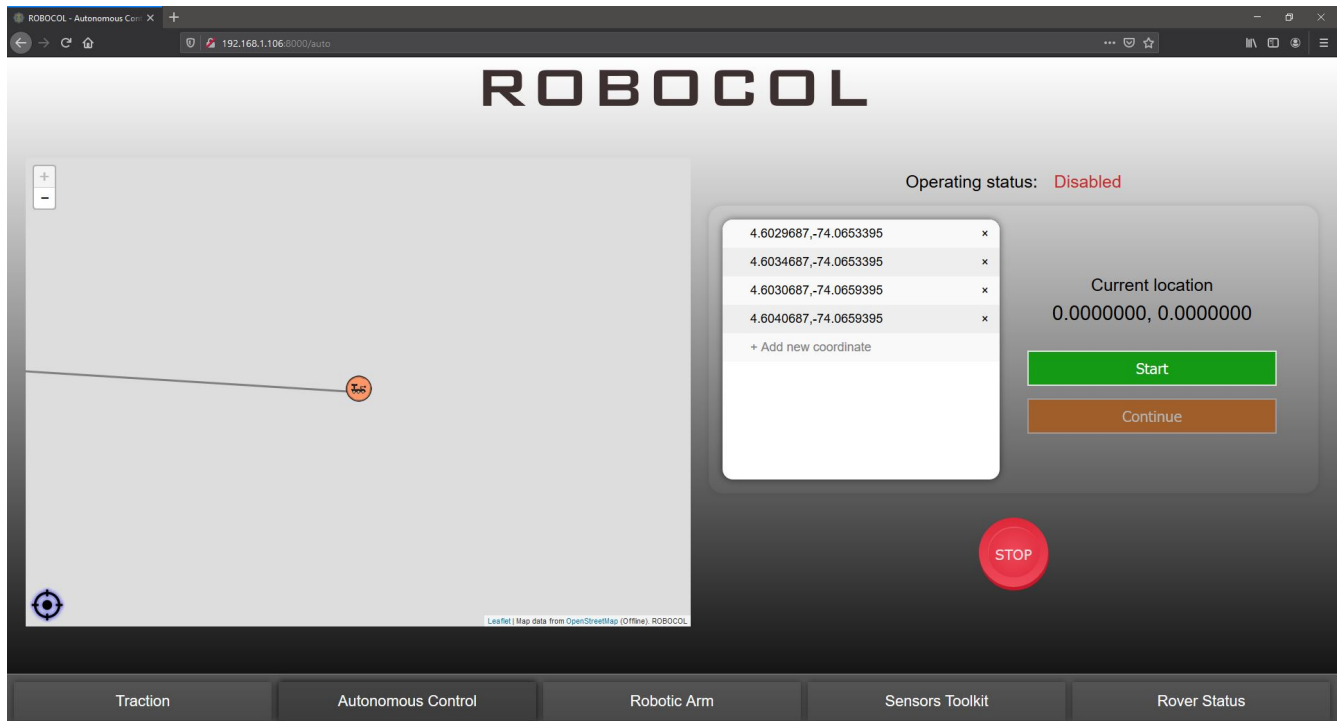
Permite controlar el movimiento del rover por medio de un joystick virtual, tiene un deslizador de sensibilidad para definir la velocidad máxima del joystick. Tiene 2 botones para definir el método de comunicación (si es por WiFi o por RF), esta función está actualmente deshabilitada aunque se usó en algún momento con una configuración anterior del rover. Los datos que entrega son los de odometría (posición relativa al punto de encendido del rover en x y en y, esto lo hace por medio de los datos de la IMU (Inertial Measurement Unit) la cual tiene un acelerómetro, un giroscopio y un magnetómetro para devolver el norte, la velocidad lineal y angular del rover. Por último cuenta con la velocidad en RPM enviada del lado izquierdo y derecho del rover que vienen dadas por la posición del joystick así como la sensibilidad al tener un joystick físico conectado al nodo de ROS. Los campos de PID eran de prueba para configurar los controladores y hay que quitarlos.



**Figure 1.** Pestaña de tracción. Permite moverlo y obtener información relacionada con el movimiento.

## 5.2. Pestaña de Autonomía

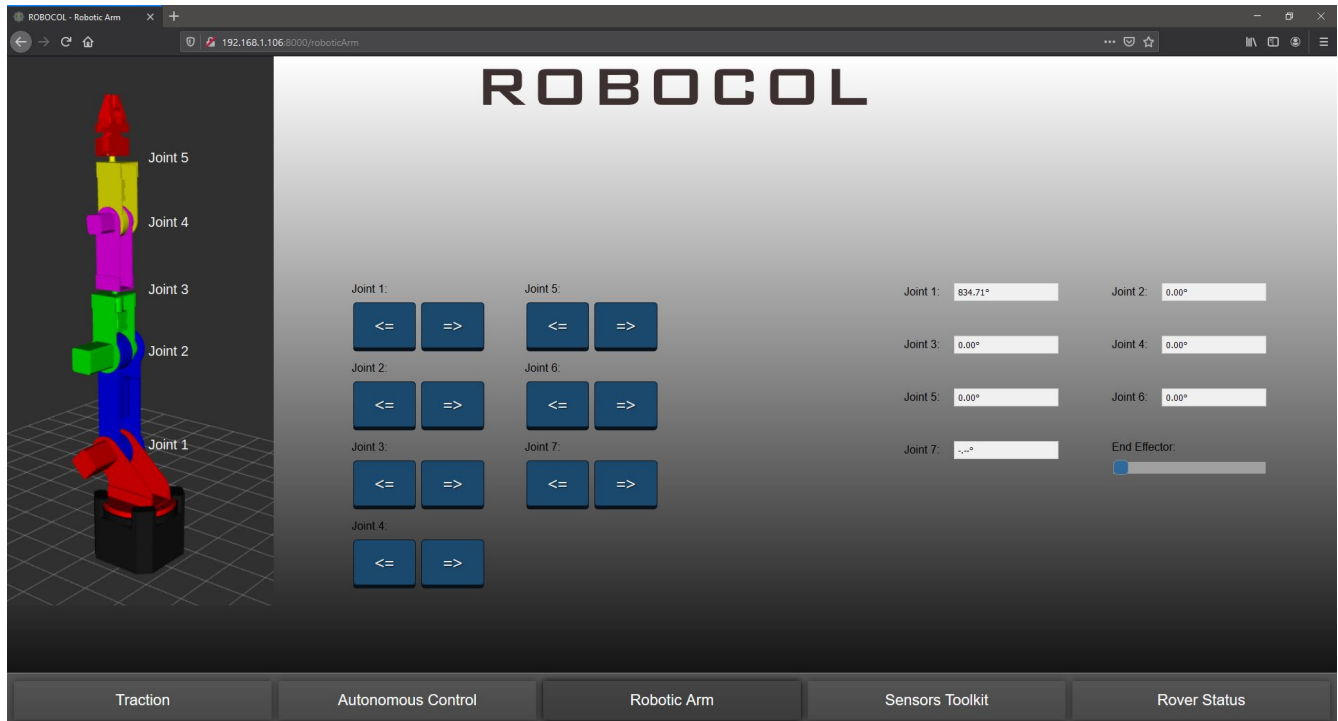
Esta pestaña fue diseñada para la movilidad autónoma del rover, tiene el mapa de Bogotá cargado por defecto (aunque en la imagen no aparece) y muestra la ubicación del rover en Bogotá dada por el GPS cuando este entrega los datos, actualmente el GPS no se ha migrado a ROS pero se hará. Además de eso permite seleccionar varios puntos de llegada dada por sus coordenadas (latitud y longitud), botones Start y Continue para empezar o seguir con la siguiente trayectoria de la lista y un botón de STOP de emergencia que se encargaría de enviar una gran cantidad de comandos de parada al rover para asegurar que este se detenga.



**Figure 2.** Pestaña de autonomía. Permite hacer un tracking del rover por medio de GPS para conocer su ubicación actual y asignar puntos de llegada con coordenadas.

### 5.3. Pestaña de Brazo Robótico

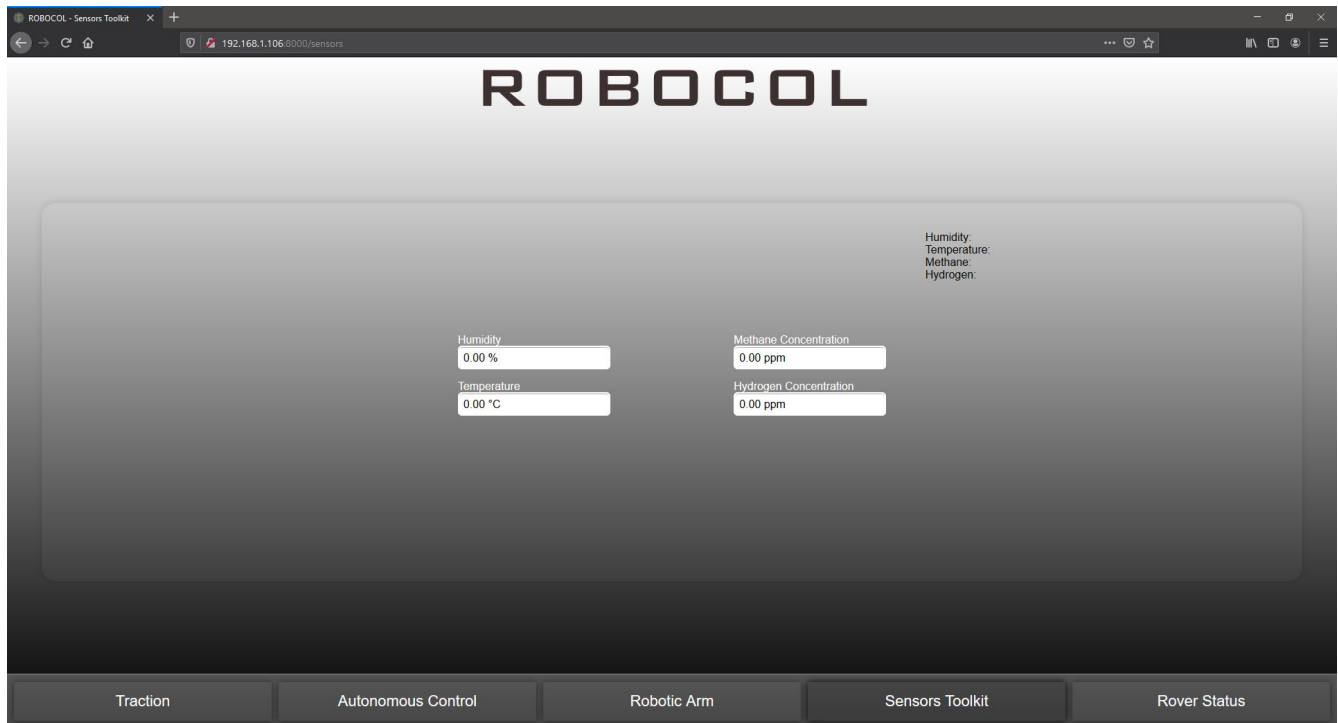
En esta sección se tienen varios botones que permiten controlar cada uno de las juntas (joints) del brazo por separado (esto es cinemática directa) el brazo que se muestra en la figura 3 es solo una referencia para saber que el número de la junta que se está moviendo, los botones lo mueven en el sentido de la manecillas del reloj o en contra. Además de esto se tiene una retroalimentación del ángulo de cada junta en grados e igual un deslizador que permite abrir y cerrar la garra (end effector).



**Figure 3.** Pestaña del brazo robótico. Permite controlar el brazo de manera remota y conocer su estado.

## 5.4. Pestaña de Sensórica

En la pestaña de sensórica se tiene actualmente los datos de humedad (en  $\%RH$ ), temperatura (en  $^{\circ}C$ ), concentración de hidrógeno y de metano (en  $PPM$ ). Esta pestaña es la que más requiere trabajo ya que se necesita comunicarse con una base de datos que contiene información de las muestras al ser medidas.

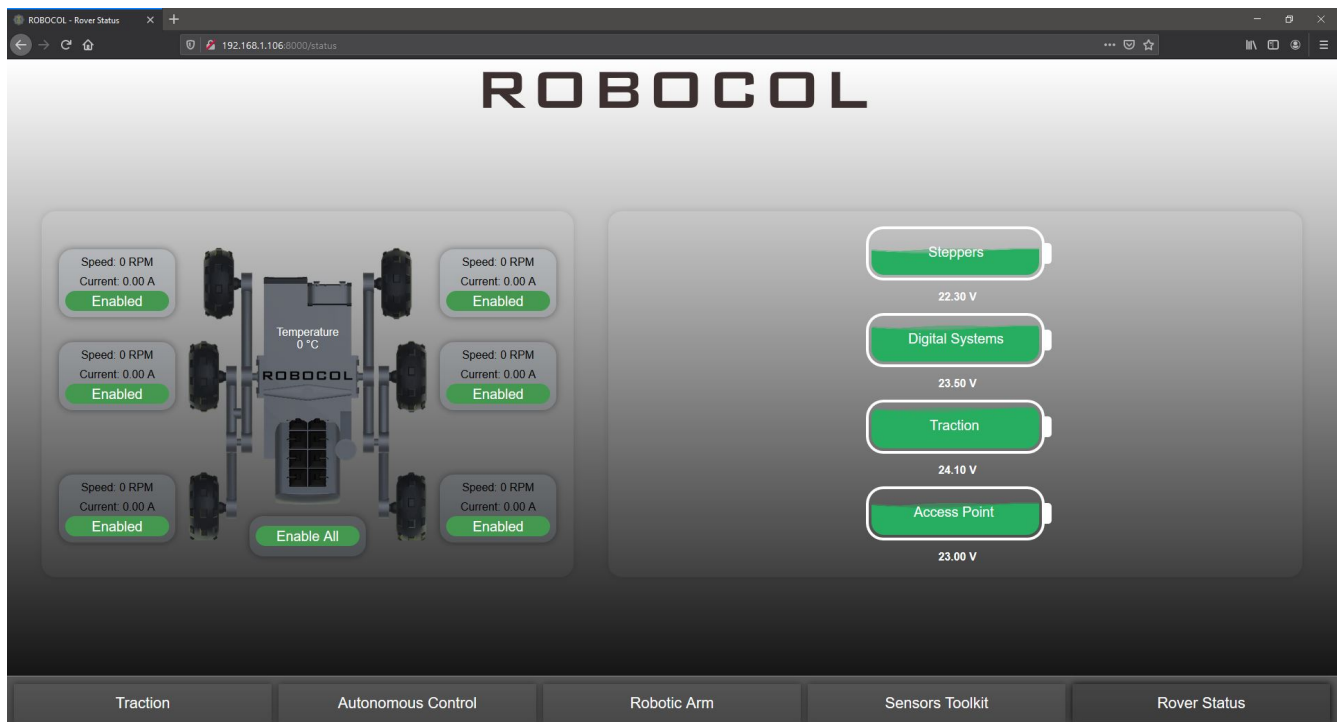


**Figure 4.** Pestaña de sensórica. Permite obtener datos de muestras recolectadas.



## 5.5. Pestaña de Estatus del Rover

La sección del estatus del rover muestra la información del voltaje de las pilas (cambie de color de verde a amarillo y a rojo al descargarse la pila). Así como la corriente y RPM en cada uno de los motores de tracción, de igual manera permite habilitar o deshabilitar la salida de cada motor de manera independiente, esto con el objetivo de realizar pruebas a ciertos motores, o desconectar alguno (o varios) en caso de fallas. El botón de *Enable All* no está funcionando pero se quiere este apague o prenda todas las llantas con un solo click. (Puede también incluirse uno de *Disable All*)



**Figure 5.** Pestaña de estatus del rover. Permite conocer el estado de las pilas, la corriente consumida por los motores y la velocidad de cada llanta. Además también se puede habilitar y deshabilitar los drivers de cada llanta de manera individual.