

# Robot Differential Drive RPM Simulation with IPC

This project simulates the RPM (Revolutions Per Minute) control of a differential drive robot using real-time data exchange and inter-process communication (IPC). Built using ROS2, it implements a system for controlling and monitoring wheel RPMs, handling differential drive kinematics, and exchanging data between processes without relying on ROS nodes.

## Installation & Setup

### ➤ Prerequisites

Make sure you have the following installed:

- ROS 2 (Humble)
- C++17 & Python3
- Colcon Build System
- Matplotlib, Seaborn (for visualization)
- Httplib (C++ library)
- Requests, Dash, Plotly, Numpy (Python Libraries)

### ➤ Install ROS Dependencies

Install ROS 2 (if not already installed)

```
"sudo apt update && sudo apt install -y ros-humble-desktop"
```

Source ROS 2 setup file

```
"source /opt/ros/humble/setup.bash"
```

Create a ROS 2 workspace

```
"mkdir -p ~/ros_ws/src && cd ~/ros_ws/src"
```

Clone the repository

```
"git clone https://github.com/yashbhaskar/robot-differential-drive-rpm-simulation-with-ipc.git"
```

Navigate to the workspace

```
"cd ~/ros_ws/"
```

Install dependencies

```
"rosdep install --from-paths src --ignore-src -r -y"
```

Build the package

```
"colcon build --packages-select rse_assignment"
```

Source the workspace

```
"source install/setup.bash"
```

### ➤ **Install C++ Dependencies**

Install C++ Compiler and Build Tools:

```
"sudo apt update && sudo apt install -y build-essential cmake g++ gcc"
```

Run the following command to install necessary C++ libraries:

```
"sudo apt update && sudo apt install -y libhttpplib-dev libjsoncpp-dev libboost-all-dev  
cmake"
```

If libhttpplib-dev is not available in your package manager, install it manually:

```
"git clone https://github.com/yhirose/cpp-httpplib.git"
```

```
"cd cpp-httpplib"
```

```
"mkdir build && cd build"
```

```
"cmake .."
```

```
"make -j$(nproc)"
```

```
"sudo make install"
```

### ➤ **Install Python Dependencies**

Install Python and Pip:

```
"sudo apt install -y python3 python3-pip python3-venv"
```

Use pip to install the required Python packages:

```
"pip install requests dash plotly numpy matplotlib"
```

## **Usage**

### **1. Launch the System**

To run all components together:

```
"ros2 launch rse_assignment launch_all.py"
```

This will:

**Play a ROS bag file** with recorded motion data.

**Run script\_a** to calculate RPM values from cmd\_vel.

**Run script\_b** to handle IPC-based data exchange.

**Run script\_c** to visualize the data on Restful API (Dash).

**Run seaborn.py** to visualize the data on seaborn GUI.

## 2. Running Components Individually

### Play Bag File

"ros2 bag play ~/ros\_ws/src/rse\_assignment/bag\_files/rse\_assignment.db3"

### Run RPM Calculation

"ros2 run rse\_assignment script\_a"

### Run IPC Data Exchange

"ros2 run rse\_assignment script\_b"

### Run Visualization on Web through dash

"ros2 run rse\_assignment script\_c.py"

### Run Visualization on GUI through seaborn

"ros2 run rse\_assignment seaborn.py"

## Visualization

The script\_c.py script generates real-time plots:

- Wheel RPM vs. Time
- Velocity Commands vs. RPM

## Visualize Data on the Web

You can monitor the real-time wheel data using the following endpoints:

- REST API Data (JSON Format) Open in browser: [http://localhost:8080/get\\_wheel\\_data](http://localhost:8080/get_wheel_data)
- Live Dashboard Open in browser: <http://0.0.0.0:8050/>

## Links

- GitHub Link : <https://github.com/yashbhaskar/robot-differential-drive-rpm-simulation-with-ipc>
- Video Link : [https://drive.google.com/drive/folders/1Xuhy8\\_Vr9Cjo5vpVSsak\\_QN1-PQaBgmw](https://drive.google.com/drive/folders/1Xuhy8_Vr9Cjo5vpVSsak_QN1-PQaBgmw)