Remote Control Robot Car

# Summary

The robot car is a two-wheel differential-drive mobile robot. The wheels are actuated by TTL DC motors whose speed and direction are sensed using quadrature encoders. The linear and angular speeds of the car can be sent over USB from a pc or over Bluetooth using the associated phone app. In order to quickly reach and maintain the desired linear and angular velocities a PID control system was implemented. An Arduino Mega is used as the microcontroller for the car.

# Features

* Maximum linear velocity set to 130 rpm, given wheel radius that is 44.24 cm/s
* Maximum rotation velocity set to 200 deg/s
* A hardware switch to change velocity command source between associated phone app and pc with indicator lights
* Can see actual current linear and angular velocity of the robot on the app
* App sends velocity commands with start and stop bits so that erroneous commands are ignored

# Operation Parameters

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| PID | | Encoder | | | Communication | | |
| Proportional | 5.75 | Pulses/Rotation | 960 | Baud Rate | | 115200 |
| Integral | 7 | Trigger Type | Change | Bluetooth Time Interval | | 75 ms |
| Derivative | 0.02 | Triggers/Rotation | 1920 |  | |  |
| Time Delay | 20 ms |  |  |  | |  |

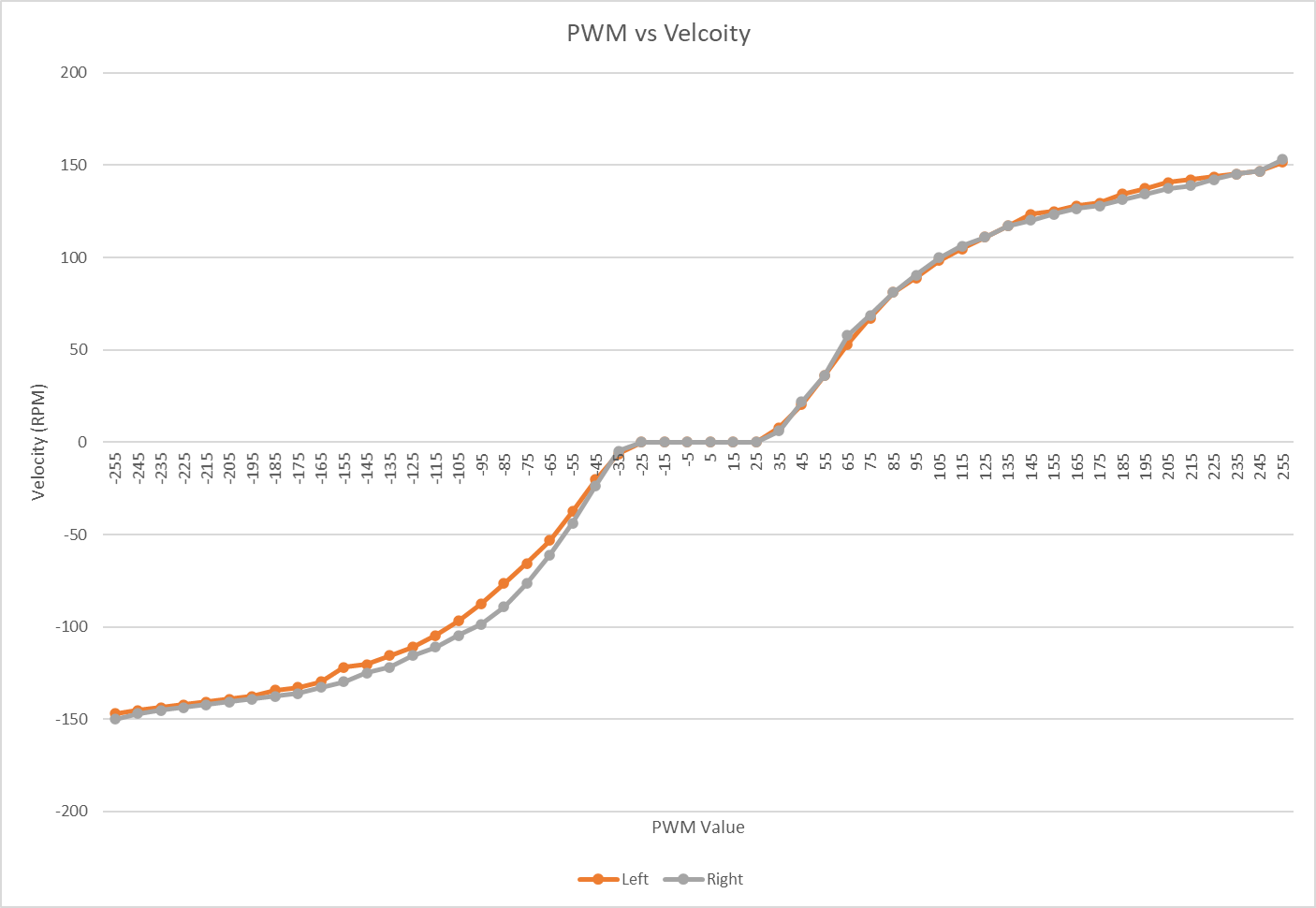
# Bill of Materials



**Table Shows parts that were bought and used over the project’s lifecycle. To see the cost of everything bought for the project look at “Bill of Materials.xlsx”.**

# Graph of Motor Speed and Velocity

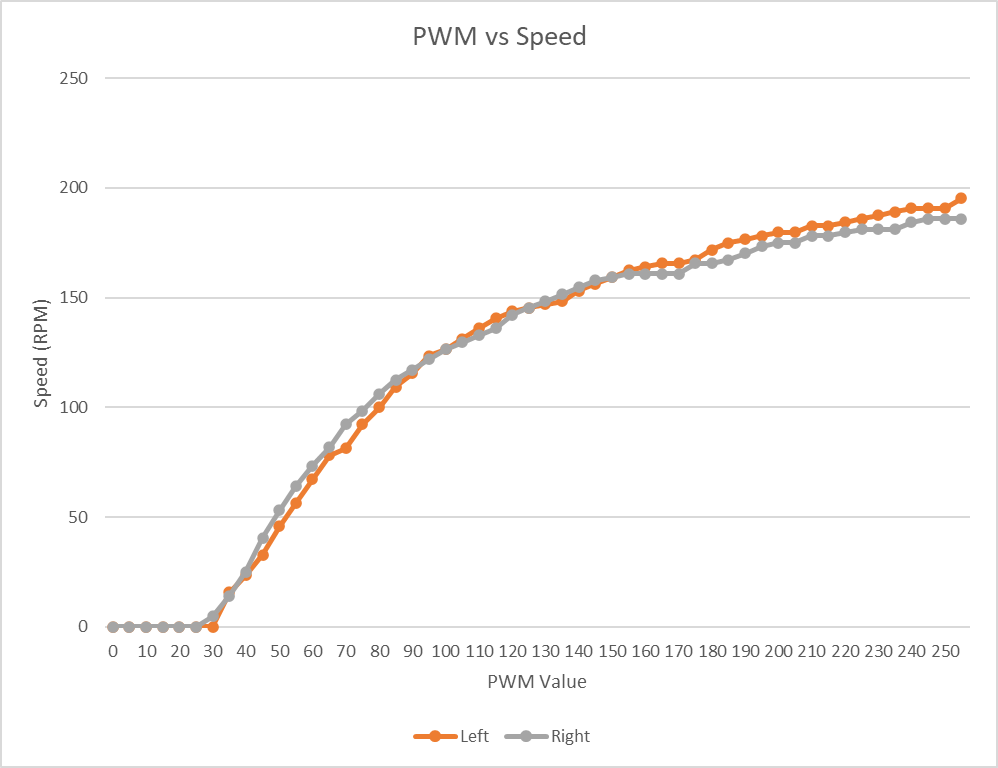
The following figure shows the velocity of the motor when the driver is powered by a 7.5V supply.



While gathering the data it was discovered that even if the PWM value is constant the velocity of the wheels can still vary by RPM over time. Also, stepping down/up to a PWM value can change what the resulting velocity will be. Differences between the absolute value of the speeds for negative and positive PWM commands were considered to be due to the aforementioned reasons. Therefore, when testing the PWM-RPM relationship at 9V only positive PWM values were considered.

The figure below shows the speed of the motor when powered by a 9V supply. The maximum speed of the motors is RPM faster than when they were powered by the 7.5V supply.

**The table that the plots are based on can be found in “Motor Speed Vs PWM and Voltages.xlsx”.**



# Step Response Test

The following figure shows the rise and settling times of the wheels when they were commanded to step to a desired speed from 0 rpm. These values were calculated after a 9-point moving average filter was applied on the speed data. Interestingly the slowest rise times (the step responses where it took more than 0.6 seconds to go from 10% to 90% of the desired value) were for desired speeds less than 40 rpm and the slowest settling times (the step responses where it took longer than 0.55 seconds to go from 0 to 95% of the desired value) occurred for desired speeds less than 70 rpm.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RPM** | **L-Tr (s)** | **R-Tr (s)** | **L-Ts (s)** | **R-Ts (s)** |
| 10 | 1.67 | 1.76 | 2.66 | 2.72 |
| 20 | 1.18 | 1.15 | 1.77 | 1.88 |
| 30 | 0.77 | 0.70 | 1.62 | 1.42 |
| 40 | 0.60 | 0.53 | 1.18 | 1.20 |
| 50 | 0.45 | 0.33 | 0.98 | 1.00 |
| 60 | 0.16 | 0.16 | 0.82 | 0.85 |
| 70 | 0.16 | 0.13 | 0.54 | 0.54 |
| 80 | 0.18 | 0.16 | 0.51 | 0.24 |
| 90 | 0.13 | 0.13 | 0.27 | 0.27 |
| 100 | 0.16 | 0.16 | 0.30 | 0.28 |
| 110 | 0.23 | 0.21 | 0.33 | 0.30 |
| 120 | 0.24 | 0.24 | 0.38 | 0.36 |
| 130 | 0.28 | 0.26 | 0.40 | 0.40 |
| 140 | 0.31 | 0.31 | 0.46 | 0.46 |
| 150 | 0.41 | 0.38 | 0.53 | 0.53 |

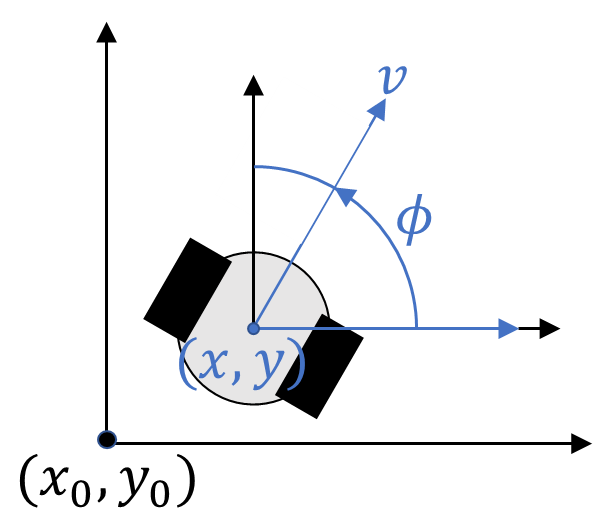
**To see the raw and filtered step response data as well as the original table detailing when the rise and settling times were reach look at “7\_5V Response Step Responses.xlsx” as well as “Time Metrics.xlsx” respectively.**

# Differential Drive Equations

time interval

Number of triggers/rotation

number of triggers in time interval



# Phone App

The phone app has sliders for controlling the robot’s linear and angular velocity and displays the current and desired velocity values. There are also buttons to stop linear movement, stop angular movement, or completely stop the robot car from moving.

Graphical user interface

Description automatically generated with medium confidence

The following image shows an example output of when both a linear and rotational velocity command is sent.

A picture containing text, electronics, screenshot

Description automatically generated

The following image shows an example output of when only a linear velocity command is sent.

Graphical user interface

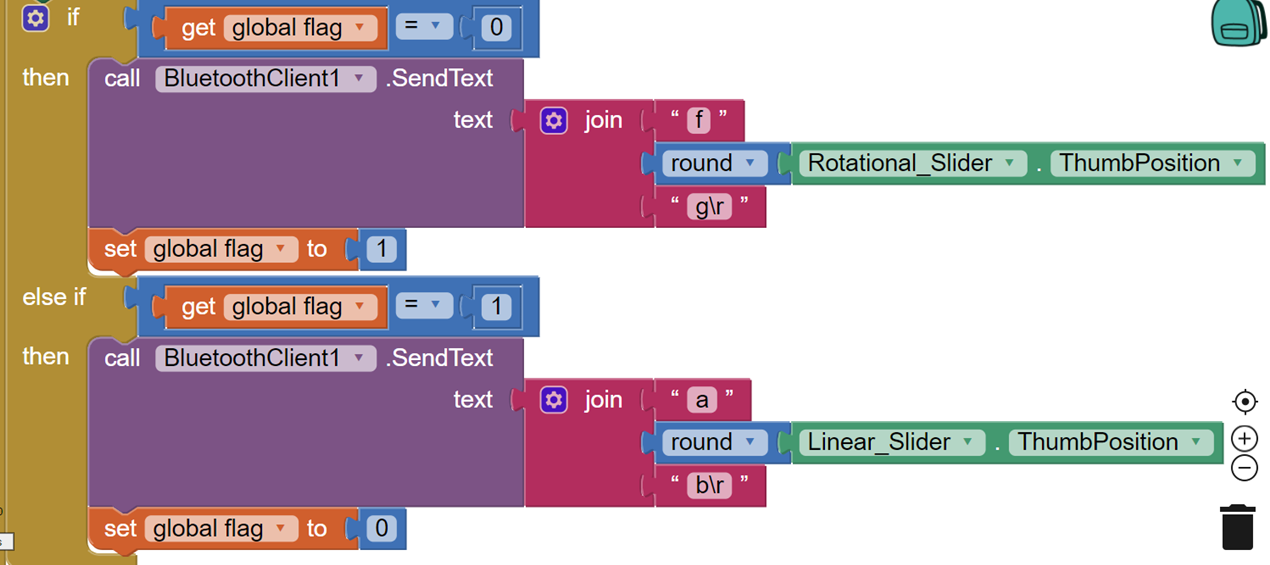
Description automatically generated

The following image shows an example output of when only a rotational velocity command is sent.

A screenshot of a computer

Description automatically generated with medium confidence

The following image shows the logic used to send the linear and rotational velocity commands to the robot as well as the how the data was packaged to allow for error detection.



# Simulink

Following image shows the Simulink model used to test the PID controller. The block code can be found in the Git repository for the project.Graphical user interface, application, Word

Description automatically generated