TASK 5

PID Control - Implement PID control to offset unknown tracking errors. Use supplied K values. You can modify them later.

1. Install latest Arduino IDE, open “Sketch” menu, “Include Library”, then “Manage Libraries..”, and install “Romi32U4” library
2. Open “MotorsFinal.ino” found within the GitHub repository for Task 5. Note that there are several parameters established within the code which include tickDist, track, Kp, Ki, Kd, motorMaxCommand, leftTrim, rightTrim, lastSpeedCmdLeft, lastSpeedCmdRight, and accelMax. Note that these parameters represent the distance traveled per encoder tick, width of Romi, the proportional gain, the integral gain, the derivative gain, the max speed of the motors, motor offsets, left and right speed controls, and a maximum acceleration for the motors.
3. The distance traveled per encoder tick is found by simply dividing the wheel circumference by number of ticks per wheel revolution. From here, the parameters that can be tuned - if needed - are those the PID gains. These parameters are essentially error handlers that attempt to adjust for error, remove error, and predict error. There are many methodologies for identifying PID values that will produce values that are with a desired-tolerance of the set-speed. For our project, the default values were implemented as they produced values that reasonably able to match the set speed established within the code. An example of the left and right motors attempting to match the set-speed for a constantly changing speed is shown in Fig. 2. In this figure, the green line represents the set speed that left and right motors are attempting to match (shown with the blue and red line).
4. The same figure shown in Fig. 2 can be reproduced by running the “MotorsFinal.ino” code, opening the serial monitor, and pressing button A.

Images:

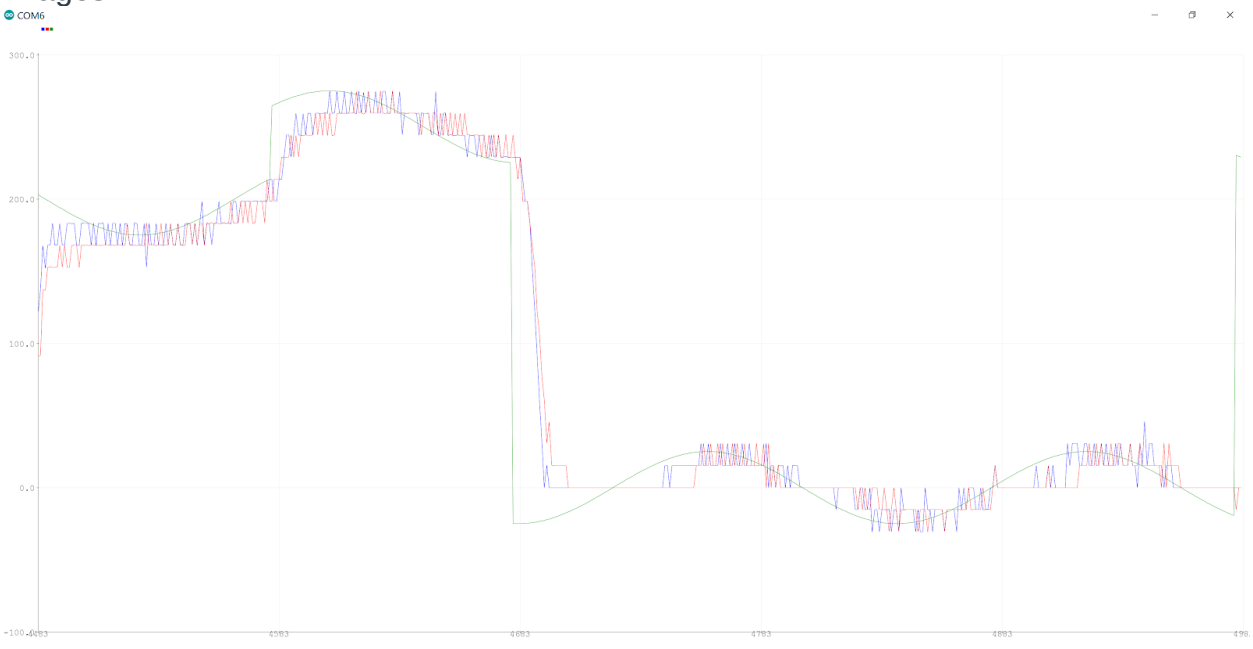

Fig. 2: Serial Plotter illustrating the left and right motor speeds (shown in blue and red) attempting to match the speed set by the code (represented by the green line