EIE3105 & EIE3123
Joint Lab Report
20040968D
Dinesh Gautam

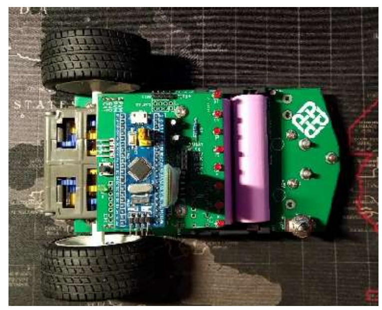
Joint Lab Report

Objective:

To design and implement a dynamic control system for the wheel on the robot car so that it moves in a straight line.

Equipment:

- Keil uVision (IDE)
- Robot Car with STM32F108C8 microcontroller

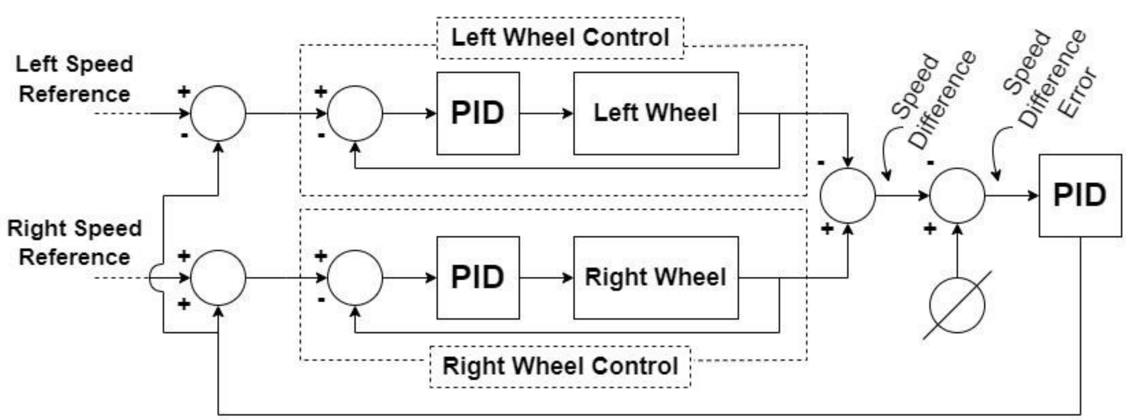


Procedure:

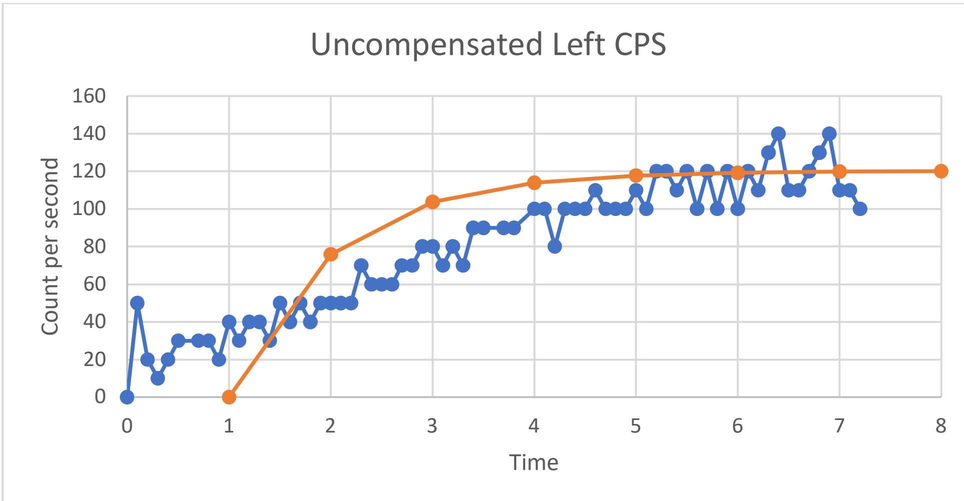
The hardware was already prepped with the necessary codes to move the car forward and get counter values from the encoder to assist in developing the control system. This report focuses on design process of the control system.

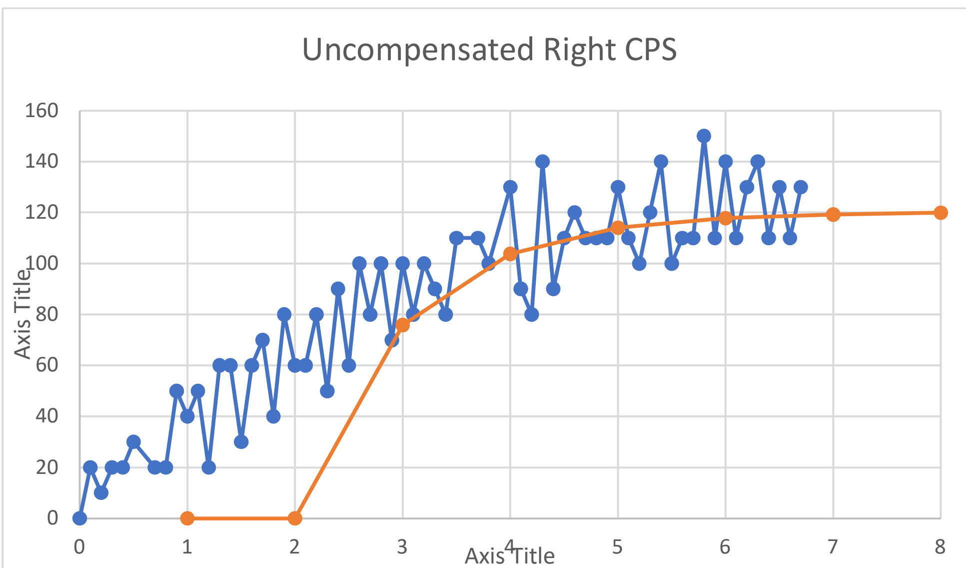
1. Design of the control system

   The diagram below shows the general control scheme of the system:

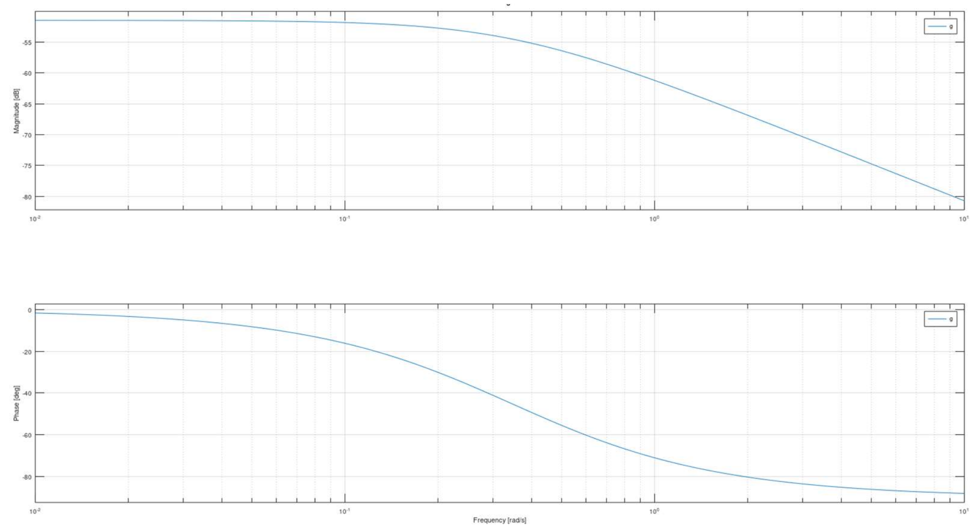


First, we have to obtain the counter values from each wheel on the car. By supplying a constant PWM of 12000 we obtained the following speed profiles for each wheel.



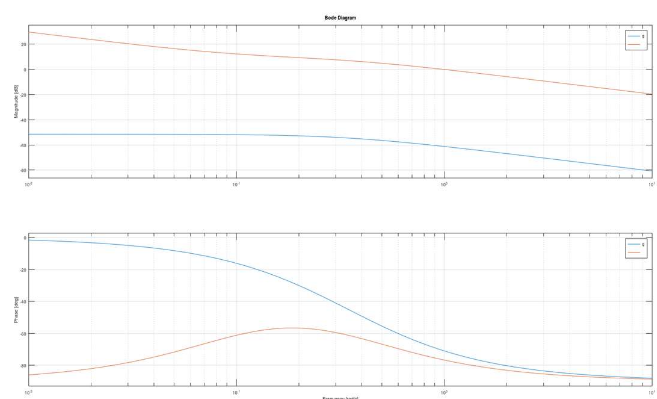


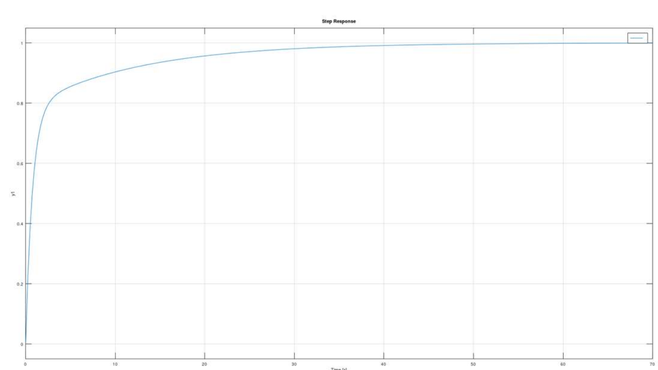
From these graphs we can see that system becomes stable when counts per second approaches 120 for both wheels this meant that we needed to develop one controller for both wheels. The system reaches 63% of that target count (80 cps) at 2.9 seconds. By using the formula $\alpha = \frac{1}{\tau}$, we get $\alpha\ =\ 0.3448$. thus, the transfer function of the of the speed control system is [$\boldsymbol{G(s) = 120(\frac{0.3448}{s+0.3448})(\frac{1}{44999})}$]. After that we find the bode plot of the transfer function



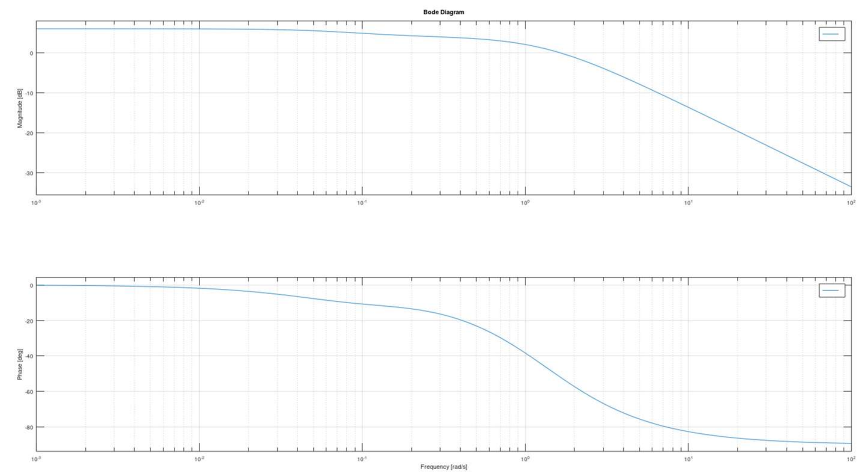
From this plot we can take the desired 0-dB frequency at (1 rad/s, -60dB). Since the phase margin has the final phase of -90 degree, we don't need a derivative controller.

For the PI speed controller **[** $Kp(\frac{s+Ki/Kp}{s})$ **],** we can fine Kp using the formula **[** $20log_{10}Kp = \mathbf{60}$ **].** The Kp of the system is 1140 and Ki/Kp is equal to 0.1, thus Ki equals 114. Therefore the PI speed controller transfer function is **[** $Gc(s) = \mathbf{1140}(\frac{s+0.1}{s})$ **].** The performance is as follows:



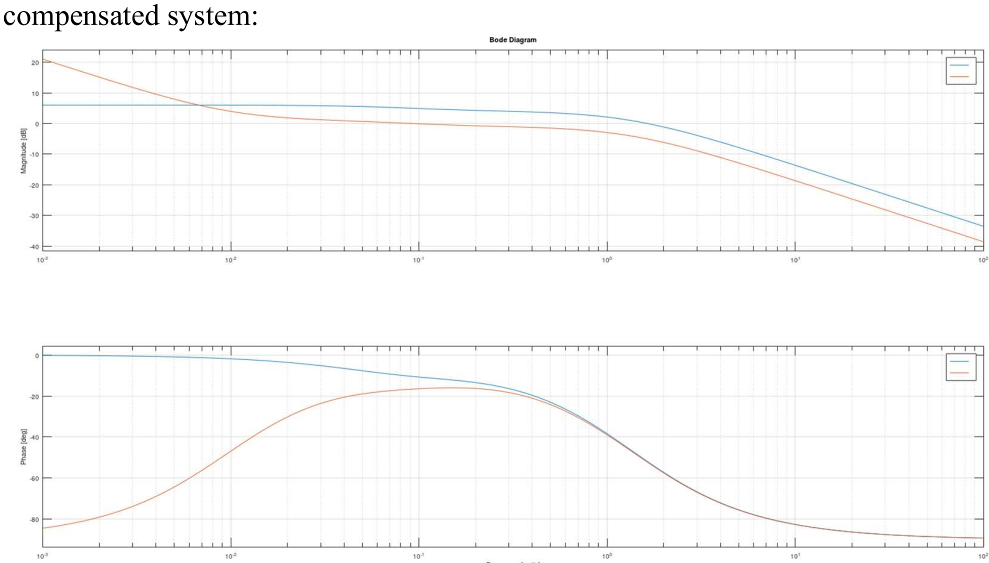


The next part is to design the line tracking loop of the system. By simplifying the closed-loop function we get $T(s) = \frac{1.048s + .1048}{s^2 + 1.393s + .1408}$. As the closed loop functions of both wheels are the same so the forward function of line tracking is 2T(s). the bode plot of the system is as follows:



We get the desired 0dB frequency at (0.1rad/s, 5dB). Since the phase margin has the final phase of -90 degrees, we don't need a derivative controller.

For the PI controller we find Kp using the formula $20log_{10}Kp = -5$ giving us Kp equal to 0.56 and Ki/kp is equal to 0.01 giving us Ki equal to 0.0056. hence the compensator of the line tracking is $Tc(s) = 0.5\mathbf{6}(\frac{s+0.\mathbf{01}}{s})$**.** the performance of the compensated system:

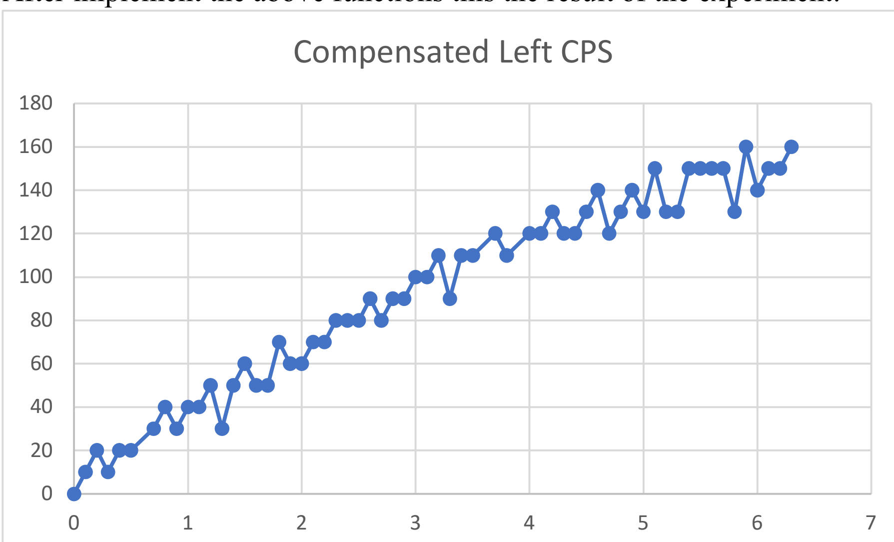


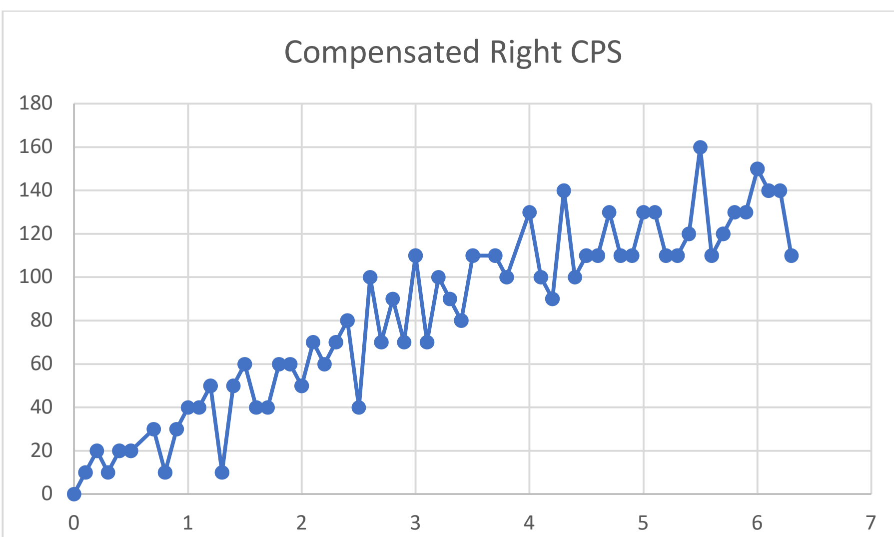
To implement the implement the dynamic control system to the robot car, we need to perform a Z-transform (with 2.5ms interval) on the designed systems above. Here are the results of the transformation.

$$Gc(s) = \mathbf{120}\left(\frac{s+0.5\mathbf{0.1}}{s}\right)] \qquad => \qquad [\ Gc(z) = \frac{\mathbf{1140}z-\mathbf{1140}}{z-1}\ ]$$

$$[\ Tc(s) = 0.56\left(\frac{s+0.\mathbf{01}}{s}\right)] \qquad => \qquad [\ Tc(z) = \frac{\mathbf{0.56}z-\mathbf{0.56}}{z-1}\ ]\backslash$$

After implement the above functions this the result of the experiment:





Difficulties:

- The connection between the Bluetooth modules was very unreliable so collecting data was difficult.
- The car still moved to the right some time due to environmental factors.
- At slow speeds the left wheel moved faster than the right wheel so some extra compensation was need for that.