3105 & 3123
Joint Lab Report
Wong Keith
19067393D

Joint Lab Report

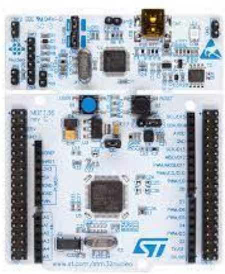
The Dynamic Control of a Robot Car

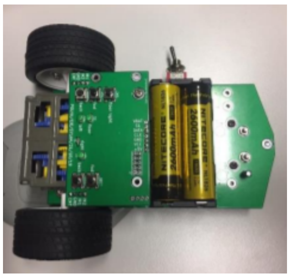
Objective:

To design and implement a dynamic control system for the wheels of a robot car so that the car can move forward on a straight line.

Equipment:

- uVision (software)
- Robot Car (hardware)
- STM32F103RBT6 (hardware)



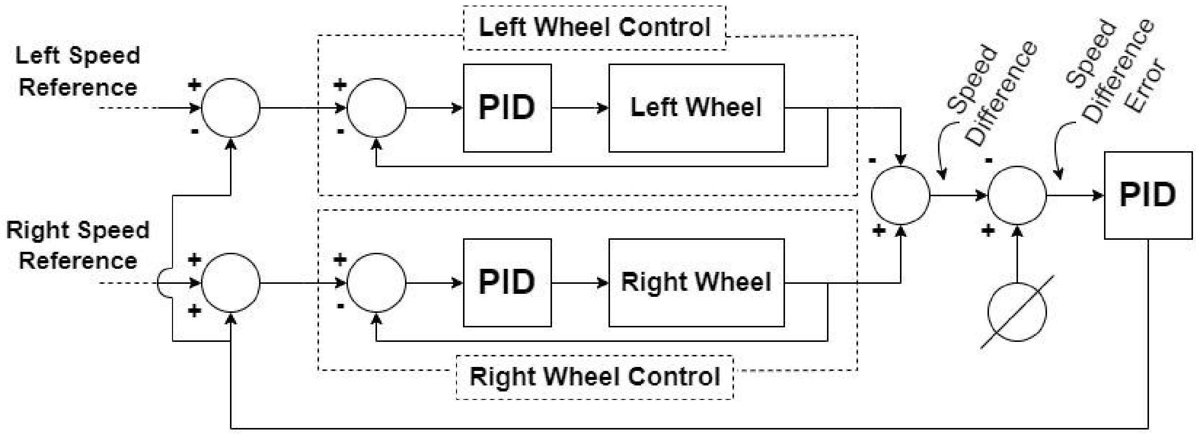


Procedures:

Hardwares (development board and robot car) have already been prepared and programmed for moving forward ideally. This report will mainly focus on the design of the dynamic closed-loop system of the robot car.

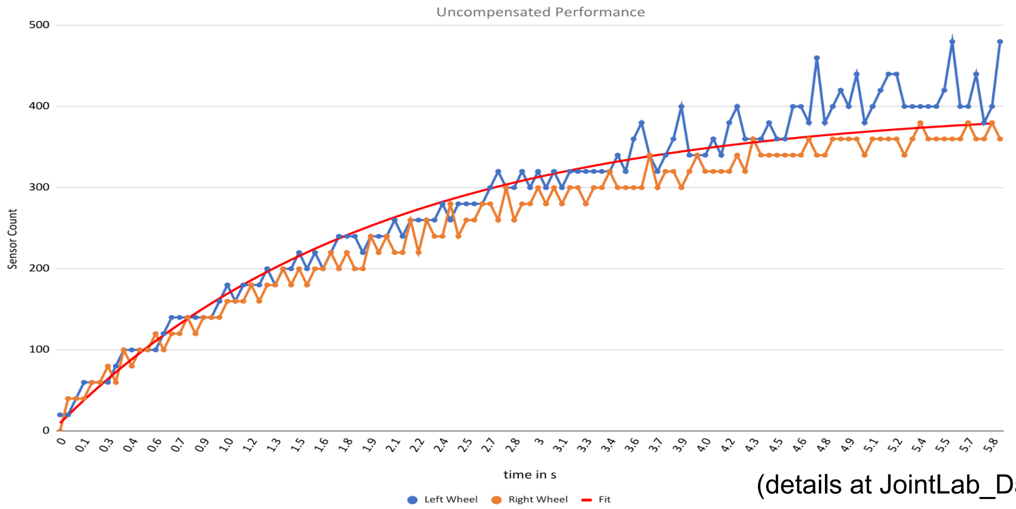
1. Design of the dynamic control system

The product of the system should like the graph shown below.



Firstly, design the part that is for Left/Right Wheel Control.

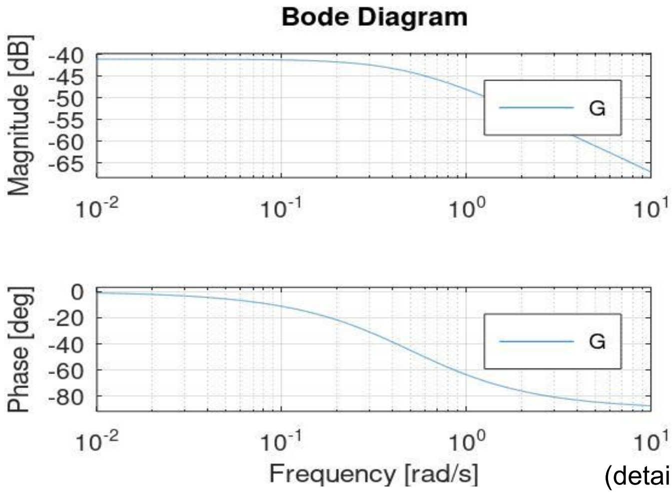
Extract datas (Sensor value from each wheel) from the robot car with a constant PWM (12500/44999 in this try).



(details at JointLab_Data.xlsx)

From the graph, we can find that the system becomes stable at around 400 counts, so we find the time that the system gets through around 63% of that of the target count (~252counts), we can get a result of around two seconds. By formula $\alpha = \frac{1}{\tau}$, we get $\alpha = 0.5$. We can draw the 'Fit line' [$f(t) = 400(1 - e^{-0.5t})$] on the graph above. Thus, estimating the transfer function of the system [$G(s) = 400(\frac{0.5}{s+0.5})(\frac{1}{44999})$].

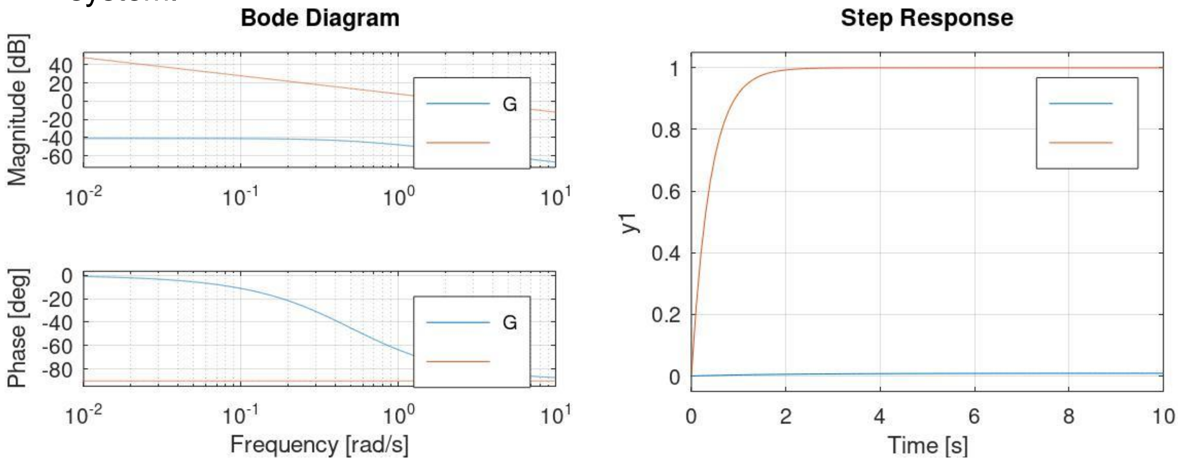
Then, find the bode plot of the system [$G(s) = 400(\frac{0.5}{s+0.5})(\frac{1}{44999})$]



(details at JointLab_Data.xlsx)

So, we can take the desired 0-dB frequency at (2.5rad/s, -55dB) and the phase margin has the final phase of -90deg, the system does not need a derivative controller.

For designing the PI controller [$Kp(\frac{s+Ki/Kp}{s})$] for the wheel control, we can find Kp from the desired 0-dB frequency and formula [$20log_{10}Kp = 55$], Kp of the system is 562.34 and Ki/Kp equals to 0.5, hence Ki equals to 281.17. Therefore, the resultant PI compensator of the wheel control system is [$Gc(s) = 562(\frac{s+0.5}{s})$]. Here is the performance of the compensated system.



(Blue: Without PI Control; Orange: With PI Control)

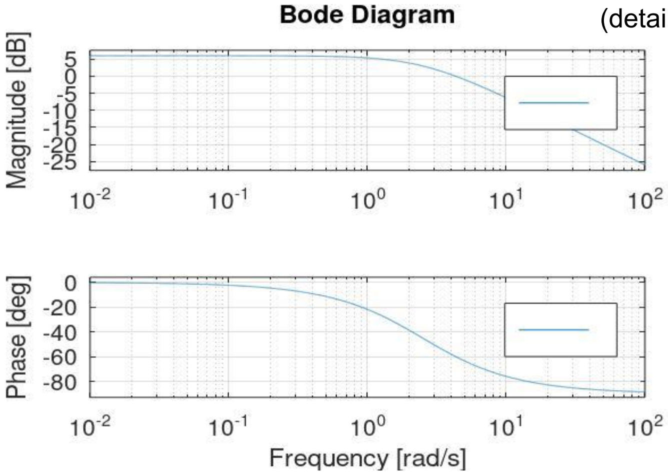
(details at JointLab_Data.xlsx)

Second, design the part for Cine Tracking of the robot car.

(Outside loop of the diagram)

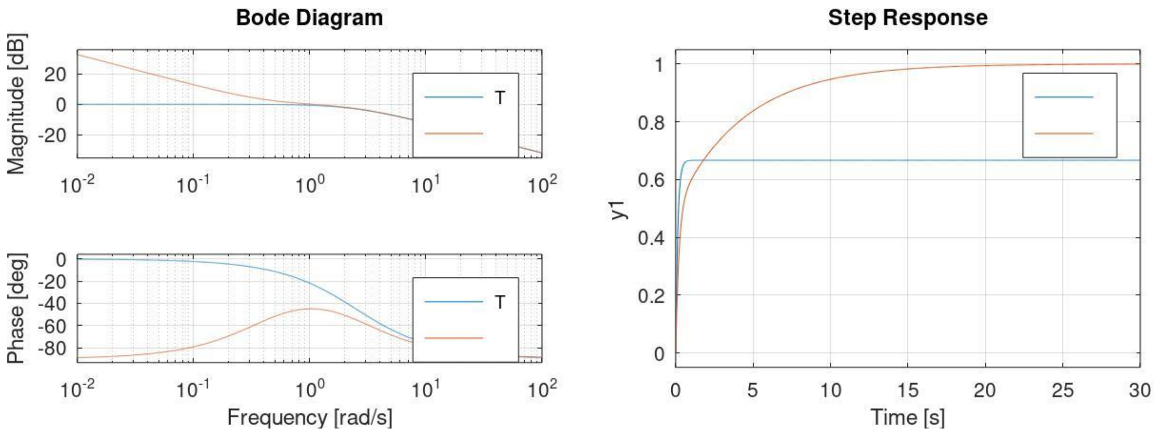
By simplifying the closed-loop function [$T(s) = \frac{2.498s+1.249}{s^2+2.998s+1.249}$] of the two wheel control, as the closed-loop function of the two wheels are the same, so the forward function of the Cine Tracking equals to $2T(s)$. Here is the bode plot of the forward function.

(details at JointLab_Data.xlsx)



So, we can find the 0-dB frequency at 4.2607rad/s, dividing the frequency by 10 (for smaller bandwidth), we can get the value of (0.42607rad/s, 5.9513dB) and the phase margin has the final phase of -90deg, the system does not need a derivative controller.

For designing the PI controller [$Kp(\frac{s+Ki/Kp}{s})$] for the Cine Tracking, we can find Kp from the measured 0-dB frequency and with the formula [$20log_{10}Kp =- 5.9513$], Kp of the system is 0.504 and Ki/Kp equals to 0.42607, hence Ki equals to 0.2147. Therefore, the resultant PI compensator of the wheel control system is [$Tc(s) = 0.504(\frac{s+0.42607}{s})$]. Here is the performance of the compensated system.



(Blue: Without PI Control; Orange: With PI Control)
(details at JointLab_Data.xlsx)

This is all for the dynamic system design of the robot car.

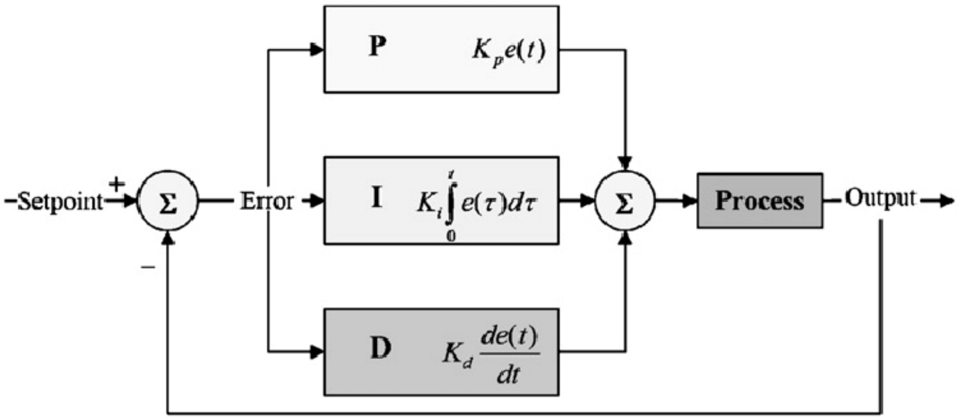
2. Implement the dynamic control system on the robot car.

In order to 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.

[$Gc(s) = 562(\frac{s+0.5}{s})$] => [$Gc(z) = \frac{562.4z-561.6}{z-1}$]

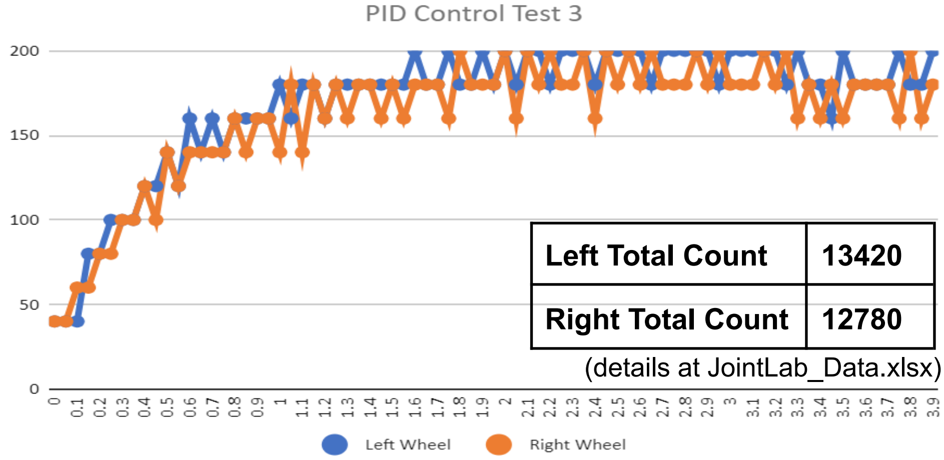
[$Tc(s) = 0.504(\frac{s+0.42607}{s})$] => [$Tc(z) = \frac{0.5043z-0.5037}{z-1}$]

For implementing the system on the program, here is the basic concept of the program designed.



<table>
<tr><td rowspan="3">// Initialization
// PI controller gain value, left wheel
float k_p_l = 562.72, k_i_l = 562;
// PI controller gain value, right wheel
float k_p_r = 562.72, k_i_r = 562;
// PI controller gain value, speed difference
float k_p_o = 0.5043, k_i_o = 0.5037;
// Output
left_pwm = k_p_l * left_e + k_i_l * left_i;
right_pwm = k_p_r * right_e + k_i_r * right_i;
diff = k_p_o * out_e + k_i_o * out_i;</td><td>// Proportional
left_e = target_left + diff - left_sum;
right_e = target_right - diff - right_sum;
out_e = 0 - out_sum;</td></tr>
<tr><td>// Integral
left_i += left_e;
right_i += right_e;
out_i += out_e;</td></tr>
<tr><td>// Derivative (NOT needed)</td></tr>
</table>

3. Experiment Result



4. Difficulties

a. In slow speed, the left wheel will go a little bit faster than that of the right, which cannot be compensated by the PID control, maybe there are hardware errors. So, sometimes it needs to multiply ~0.95 on the left PWM to maintain a straight line.
b. Although sometimes the graph seems the car going a well straight, but the car bends in a considerable angle, it may be caused by the environment errors (not even floor).