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:

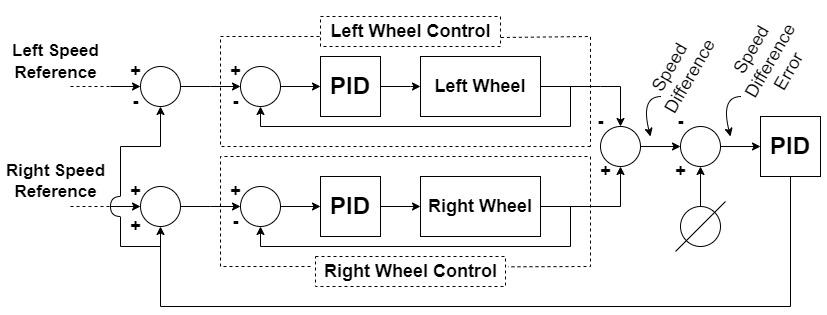
* uVision (software)
* Robot Car (hardware)
* STM32F103RBT6 (hardware)

## Procedures:

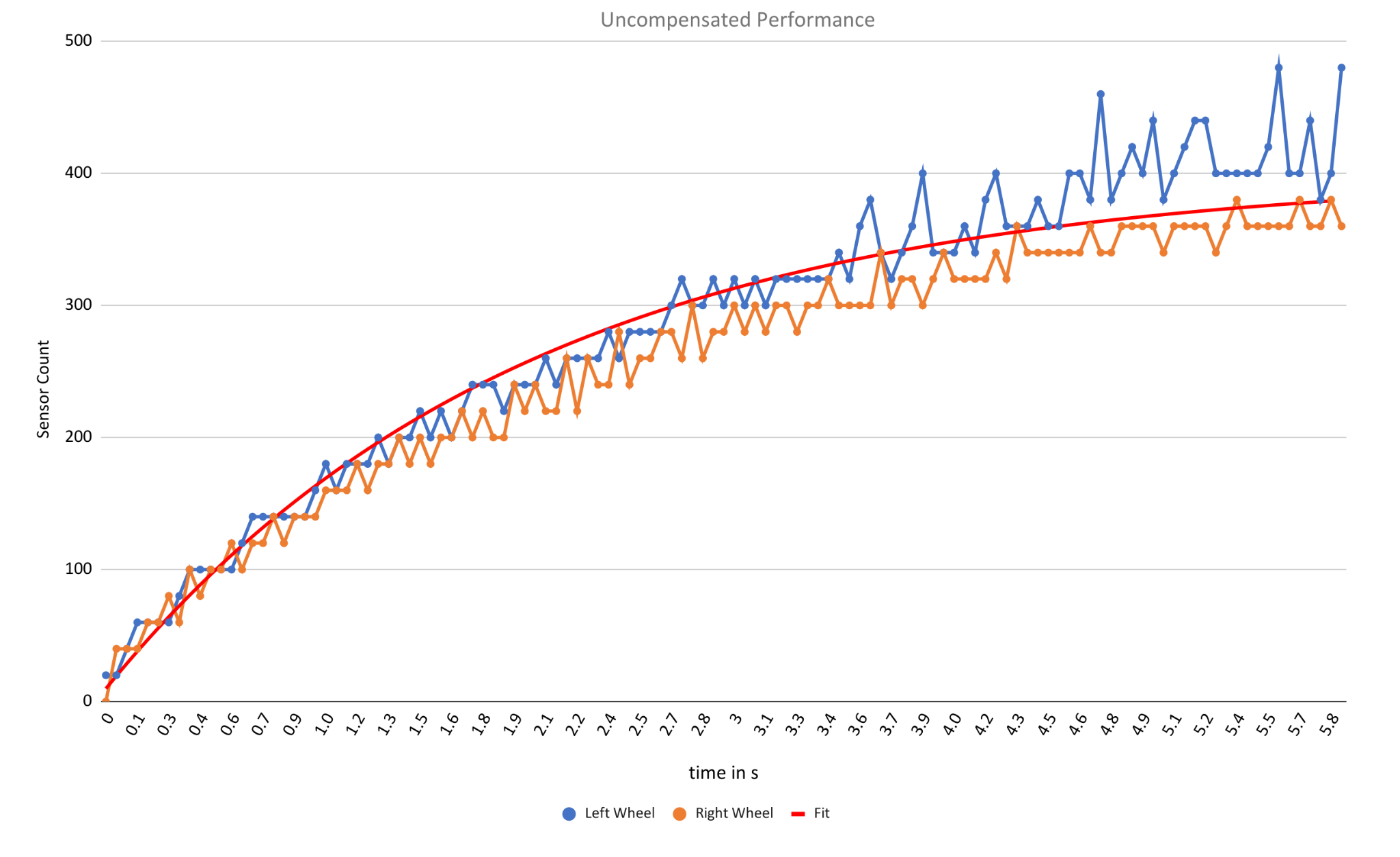
Hardwares (development board and robot car) have already been prepared and programmed for moving forward ideally. This report will mainly focus on the design of the dynamic closed-loop system of the robot car.

### Design of the dynamic control system

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

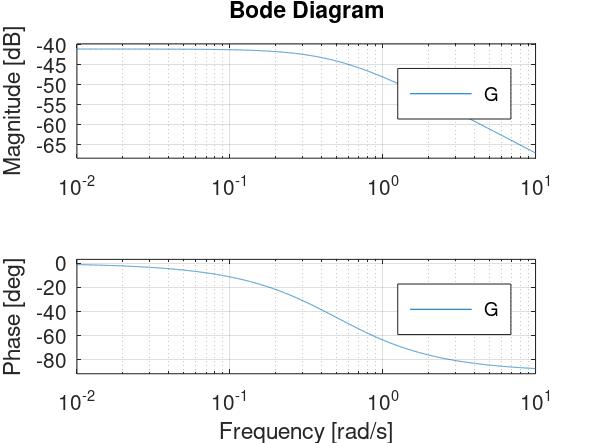


Firstly, design the part that is for Left/Right Wheel Control.

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

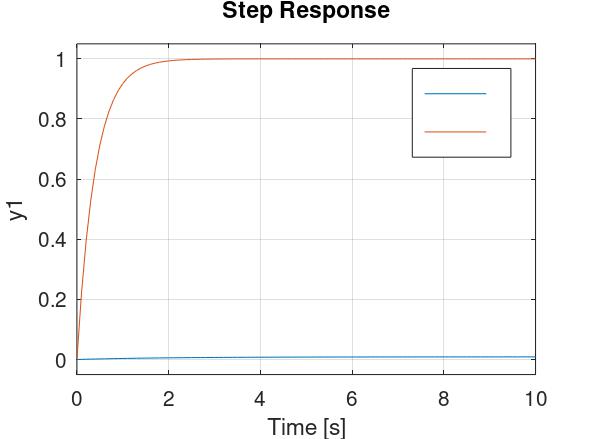
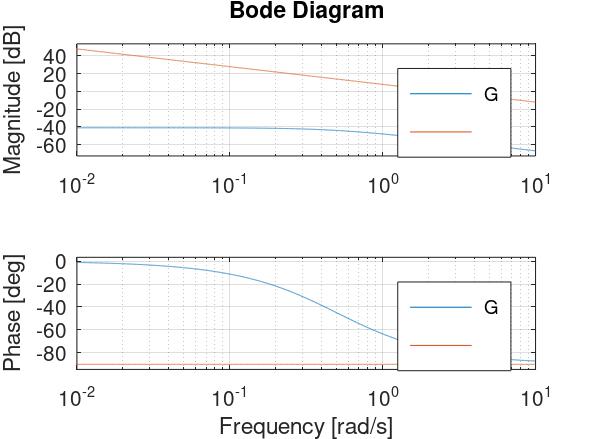
(details at JointLab\_Data.xlsx)

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 , we get . We can draw the ‘Fit line’ **[ ]** on the graph above. Thus, estimating the transfer function of the system **[ ]**.

Then, find the bode plot of the system **[ ]**

(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 **[ ]** for the wheel control, we can find Kp from the desired 0-dB frequency and formula **[ ]**, 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 **[ ]**. Here is the performance of the compensated system.

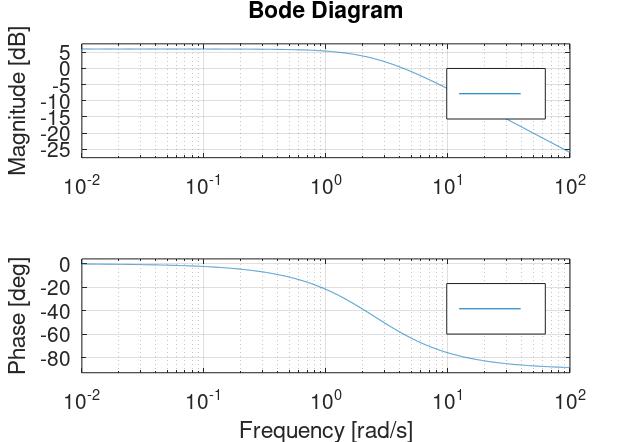
(Blue: Without PI Control; Orange: With PI Control)

(details at JointLab\_Data.xlsx)

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

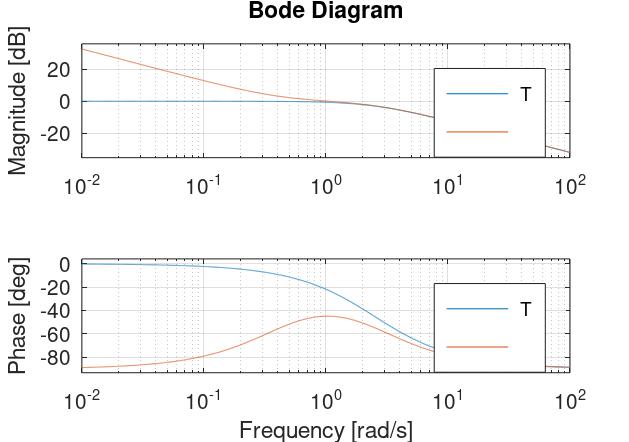
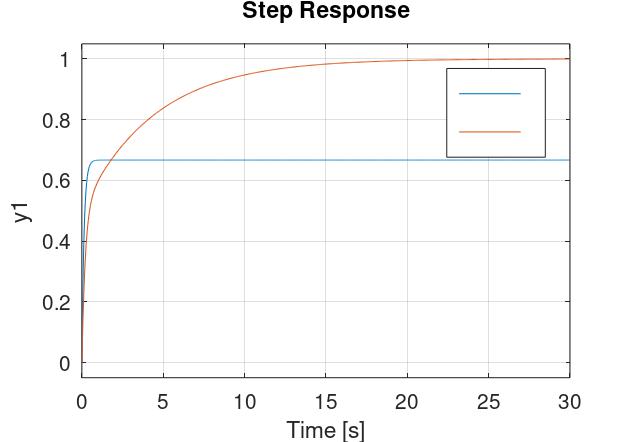
(Outside loop of the diagram)

By simplifying the closed-loop function **[ ]** 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 . 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 **[ ]** for the Cine Tracking, we can find Kp from the measured 0-dB frequency and with the formula **[ ]**, 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 **[ ]**. Here is the performance of the compensated system.

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

(details at JointLab\_Data.xlsx)

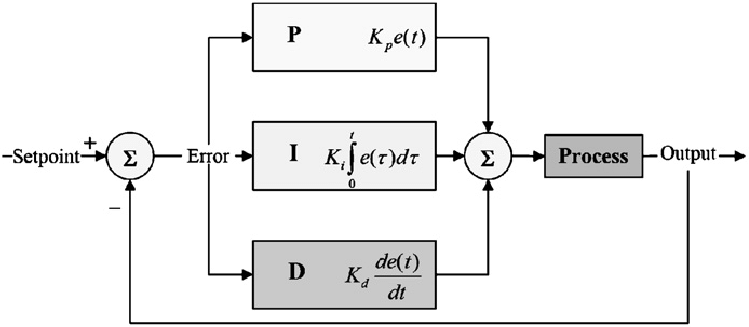
This is all for the dynamic system design of the robot car.

### 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.

**[ ]** => **[ ]**

**[ ]** => **[ ]**

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

| **// 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; | **// Proportional**  left\_e = target\_left + diff - left\_sum;  right\_e = target\_right - diff - right\_sum;  out\_e = 0 - out\_sum; |
| --- | --- |
| **// Integral**  left\_i += left\_e;  right\_i += right\_e;  out\_i += out\_e; |
| **// Derivative (NOT needed)** |

### Experiment Result圖表

| **Left Total Count** | **13420** |
| --- | --- |
| **Right Total Count** | **12780** |

(details at JointLab\_Data.xlsx)

### Difficulties

1. 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.

1. 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).