HO CHI MINH UNIVERSITY OF TECHNOLOGY

VIETNAM NATIONAL UNIVERSITY

FALCUTY OF MECHANICAL ENGINEERING

---------------oOo----------------

MECHATRONICS DESIGN PROJECT

Instructor: Assoc. Prof. Vo Tuong Quan, PhD.

Student's name	ID
Huỳnh Bá Lộc	1952326
Nguyễn Hoàng Minh	1952335
Trương An Quốc	1952422
Phan Đăng Khôi Nguyên	1952366

Ho Chi Minh City - 2022

Table of Contents

List of Figures

List of Tables

CHAPTER I: OVERVIEW

1.1 Introduction

Line following robot is a type of mobile robots which move by wheels. This type of robot is designed to follow a line that is drawn, painted or glued to the ground. The trajectory of motion will be designed base on purposes of the user.

Line following robots are used in automatic transporting of cargo in warehouses, workshops, harbours,… and used in various studies about detection technology and designation of controllers, as well as being the object of many engineering competitions.

Most line following robots consist of these main parts: chassis and frame, wheels (driving wheels controlled by motors and passive wheels), sensors system, control system and controller, power supply and load.

1.2 Principle of working

Various principle diagrams of working can be used for designing a line following robot. To suffice the requirements of line tracking and weight carrying, principle diagrams of AGVs are best fit. However, some other line following robots might still be used to investigate the best principle for our robot.

Here are some robots which principles and mechanical specifications worth taken in to account:

a. Savant Automation's Model DC-10S

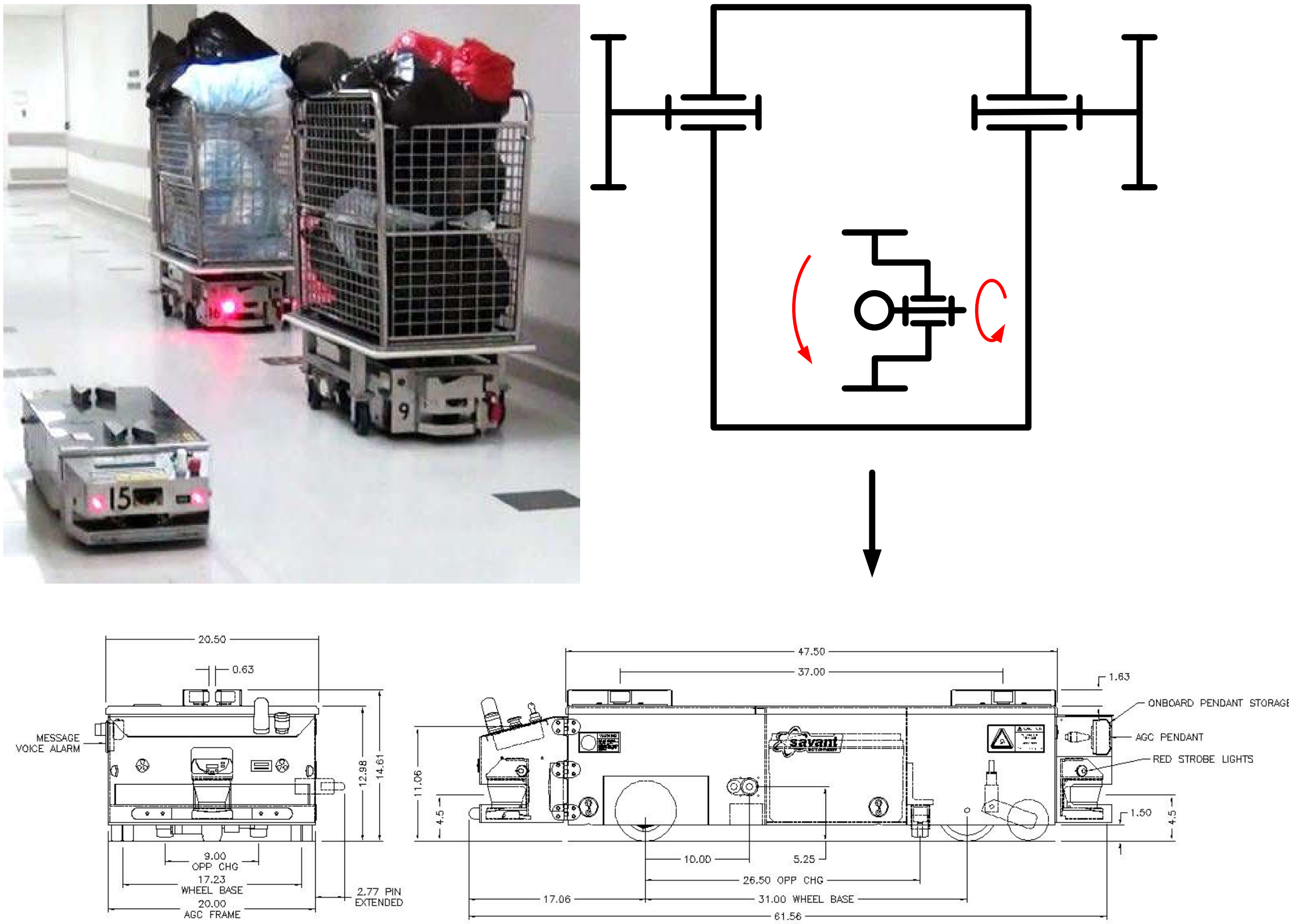


Fig 1.1 Savant Automation's Model DC-10S, its dimensions and its principle diagram [1]

Parameter	Value
Capacity	≈ 907 kg
Drive unit	Single wheel steer/drive
Maximum speed	≈ 1.03 m/s
Maximum battery cycle	8 hrs

Table 1.1 Specifications of Savant Automation's Model DC-10S [1]

Savant Automation's Model DC-10S is an AGV used to transport cargo and carts directly from station to station right in the manufacturing process. By using conveyors, rollers or pushing mechanisms,... cargo can be placed on AGVs without having to pause the process. It also has automatic cart elevation to lift carts.

Advantages:

+ Low profile ensure better stability, allows it to drive under prepositioned carts.

+ Stainless steel body.

+ Use "virtual path" navigation system so the workspace needn't be rehabilitated.

Disadvantages:

+ Heavy weight, approximately 182kg (400 lbs) when empty.

+ Single wheel steer/drive makes the robot expensive.

b. Rocla's AWT Reach Fork

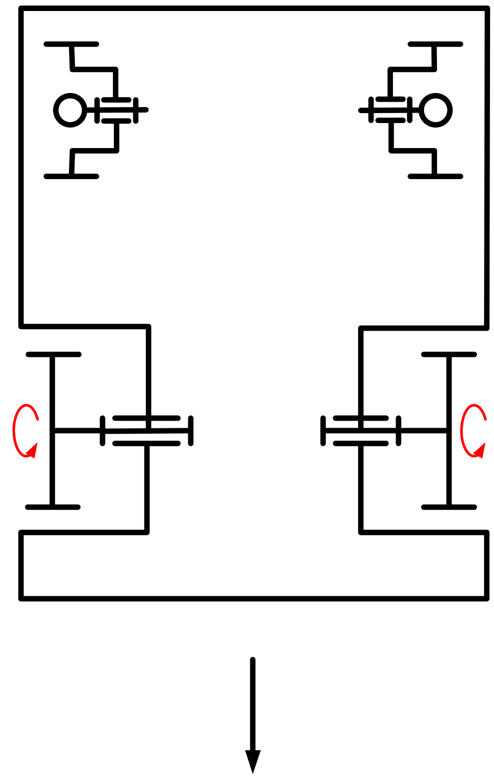


Fig 1.2 Rocla's AWT Reach Fork and its principle diagram [2]

Parameter	Value
Capacity	≈ 1200 kg
Drive unit	Single wheel steer/drive
Maximum speed	≈ 1 m/s
Maximum battery cycle	16 hrs

Table 1.2 Specifications of Rocla's AWT Reach Fork [2]

Rocla's AWT Reach Fork is an AGV that fully compatible with existing racks and warehouse layouts, which makes it easy to automate the existing warehouse.

Advantages:

+ Very high reach, small turning radius ensures working in compact warehouse.

+ Long capacity batteries.

Disadvantages:

+ High cost.

+ Long batteries life.

c. KIVA AGVs used by Amazon

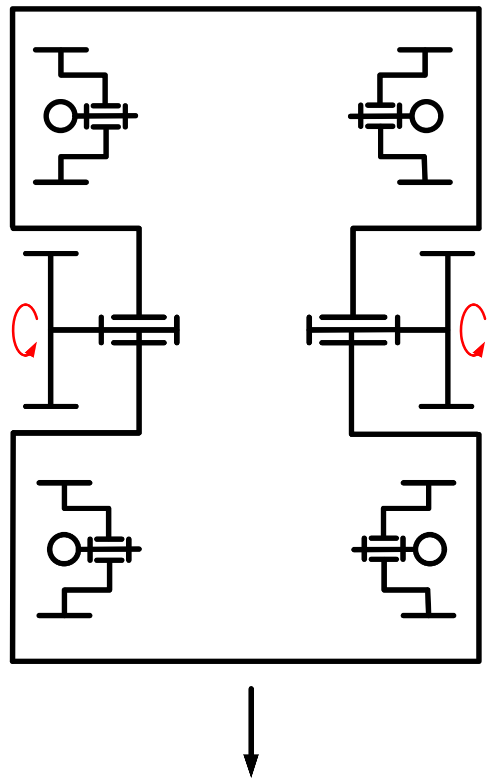


Fig 1.3 "The Pegasus" KIVA AGVs and its principle diagram [3]

Parameter	**Value**
Capacity	≈ 560 kg
Drive unit	2 wheels drive differential steering
Maximum speed	≈ 1.38 m/s
Maximum battery cycle	24 hrs

Table 1.3 Specifications of "The Pegasus" KIVA AGVs used by Amazon [3]

KIVA AGVs are mobile robots that frequently used by e-commerce companies like Amazon, Alibaba, and recently used in medical.

Advantages:

+ Low profile (allows it to drive under prepositioned carts).

+ High speed, high flexibility due to the use of differential drive wheels.

+ Long batteries cycle.

+ 6-wheels principle ensure better weight distribution.

Disadvantages:

+ Small capacity.

d. Toyota Autopilot TAE050

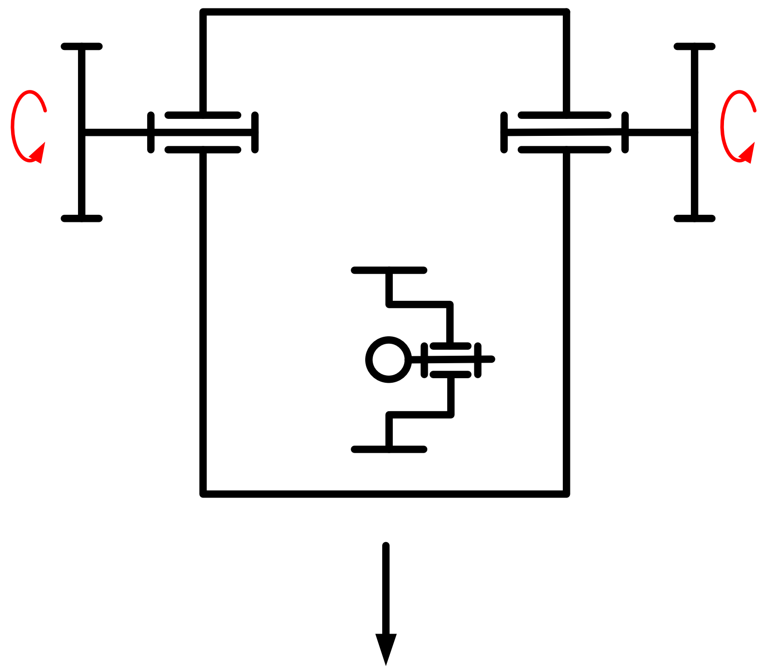


Fig 1.4 Toyota Autopilot TAE050 and its principle diagram [4]

Parameter	**Value**
Capacity	≈ 500 kg
Drive unit	2 wheels drive differential steering
Maximum speed	≈ 0.84 m/s
Maximum battery cycle	16 hrs

Table 1.4 Specifications of Toyota Autopilot TAE050 [4]

Toyota Autopilot TAE050 is an AGV that mostly used to tow cargo loaded on wagons or carts.

Advantages:

+ Long batteries cycle.

+ Reliable way of tracking indoor.

Disadvantages:

+ Outdated principle of robot.

e. CarryBee (Dolly type) of Aichikikai Techno System

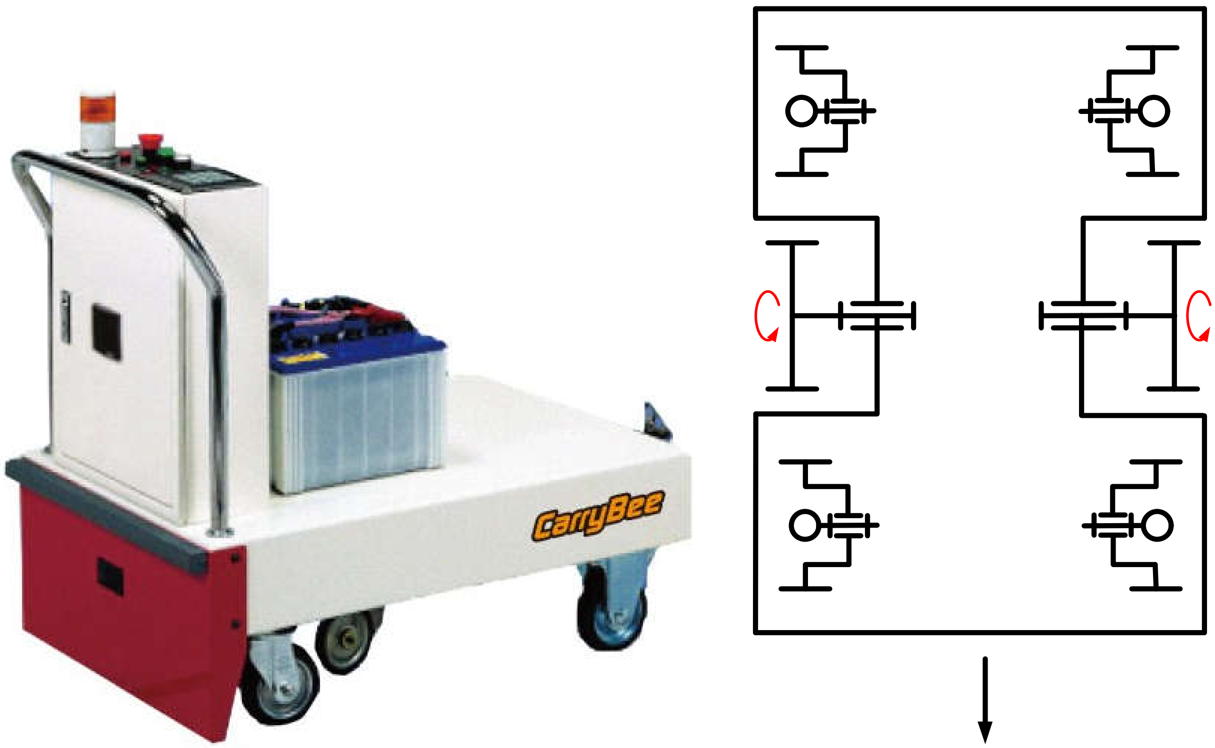


Fig 1.5 CarryBee (Dolly type) AGV and its principle diagram [5]

Parameter	**Value**
Capacity	≈ 1500 kg
Drive unit	2 wheels drive differential steering
Maximum speed	≈ 0.45 m/s
Maximum battery cycle	8 hrs

Table 1.5 Specifications of CarryBee (Dolly type) AGV [5]

Advantages:

+ Built in module type, easily fit in many kinds of robot frame.

+ High capacity, can carry enormous loads.

Disadvantages:

+ High cost.

+ Low speed.

+ Space consuming.

f. Innok Heros of Innok Robotics

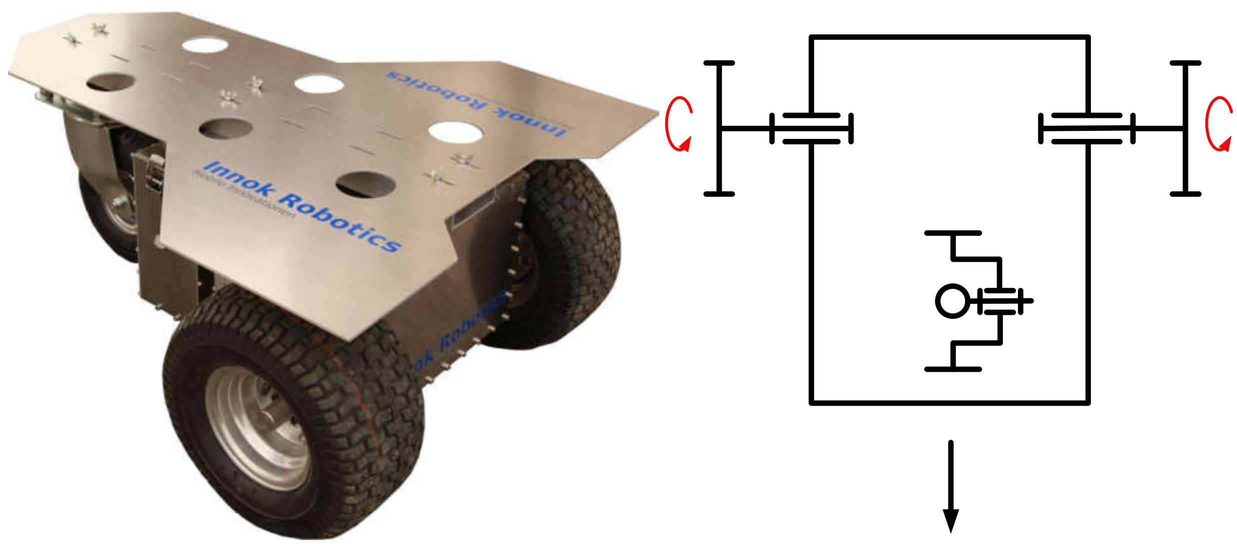


Fig 1.6 Innok Heros and it principle diagram [6]

Parameter	**Value**
Capacity	≈ 400 kg
Drive unit	2 wheels drive differential steering
Maximum speed	≈ 0.9 m/s
Maximum battery cycle	6 hrs

Table 1.6 Specifications of Innok Heros [6]

Advantages:

+ Perform well in outdoor environment and various terrain.

Disadvantages:

+ Short batteries cycle.

+ Small capacity.

1.3 Problem Statements

Design a line-following robot that use wheels to transverse. The robot will move along the lines drawn on the ground surface with these specifications:

- Maximum speed: 0.3 m/s (for indoor area, maximum speed allowed for AGVs must lower than avg walking speed of workers $\approx 1\ m/s \rightarrow 0.3\ m/s$ is acceptable [7].
- Minimum turning radius R_{min}: 500 mm
- Maximum error of tracking (on the whole line): ± 5 mm
- Robot can recognize color of load.

CHAPTER II: SELECTIONS

2.1 Mechanical selection:

Consider previous models that we have introduced in the first chapter, our group acknowledged that all principles ensure line tracking and weight carrying. However, we need to decide which principle is simplest to design, yet best suit our technical requirements.

2.1.1 Principle diagram:

In this chapter, we will review possible principle diagrams for our robot. Through that, we can analyze advantages and disadvantages of each solution and select mechanical properties base on these constraints:

- Minimum speed: 0.1 m/s

- Loaded cargo weight: 2kg

- Terrain surface type: Flat, no Ramp.

- Moving with a continuous line, one intersection.

- Robot has simple design and a reasonable price.

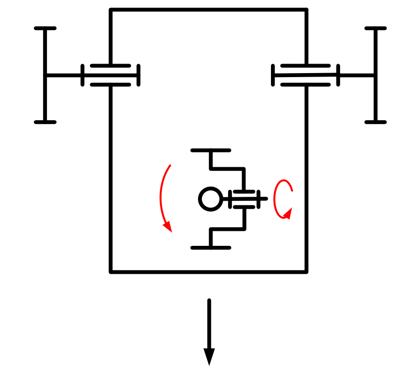
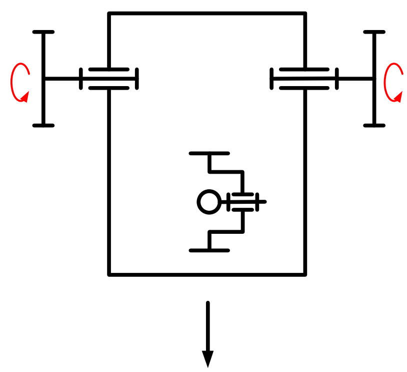
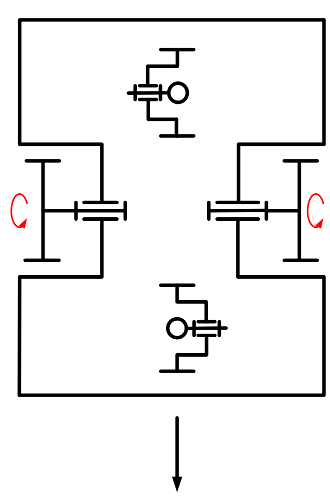
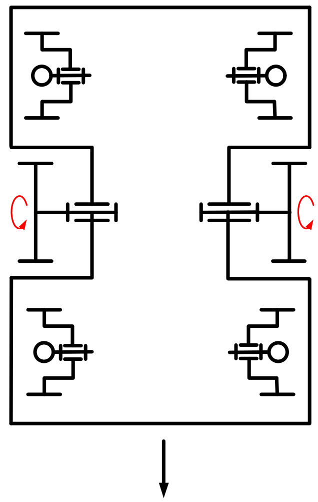
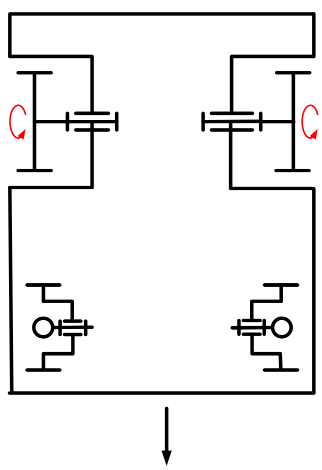
Configurations	Principle diagram	Advantages	Disadvantages
Three-wheel robot (1 driving and steering front wheel)	 ***Fig 2.1*** Principle diagram of config. 1	- Simple mechanism - Three wheels always coplanar - Better traction	- Higher moment of inertia requires motor to be more powerful - Expensive active wheel (both steering/driving)
Three-wheel robot (2 driving wheels behind)	 ***Fig 2.2*** Principle diagram of config. 2	- Simple mechanism - Three wheels always coplanar - Better traction	- Must synchronize the speed of two motor (or the robot will shake) - Chances of rolling over when steering in high speed
Four-wheel robot (2 driving wheels in the middle)		- Better weight distribution (4 wheels) - No extra torque on motor (weight is right above motors) - Flexible, can rotate around its center	- Must synchronize the speed of two motor (or the robot will shake) - Chances of slipping while steering in high speed - Four wheels may not coplanar

	Fig 2.3 Principle diagram of config. 3		
Six-wheel robot (2 driving wheels in the middle)	 ***Fig 2.4*** Principle diagram of config. 4	- High stability when steering - Can rotate around its center - Better weight distribution (6 wheels) help carrying bigger and heavier load - No extra torque on motor (weight is right above motors)	- Must synchronize the speed of two motor (or the robot will shake) - High cost - More wheels mean it is harder to ensure they are all coplanar
Four-wheel robot (2 driving rear wheels)	 ***Fig 2.5*** Principle diagram of config. 5	Better weight distribution (4 wheels)	- Must synchronize the speed of two motor (or the robot will shake) - Four wheels may not coplanar

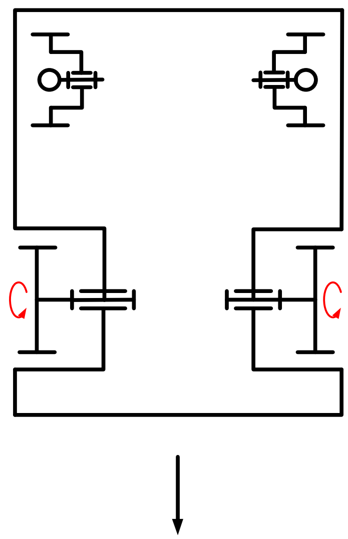
Four-wheel robot (2 driving front wheels)	 ***Fig 2.6*** Principle diagram of config. 6	Better weight distribution (4 wheels)	- Must synchronize the speed of two motor (or the robot will shake) - Four wheels may not coplanar

Table 2.1 Comparison between some principle diagrams

As we compared some frequently used principal diagrams for AGVs and mobile robots, we have chosen the configuration for our robot. Our robot needs a simple structure, easy to drive/steer, better traction and an affordable price.

→ The line following robot will be designed with **3 wheels**, where 2 active wheels are placed in the back and 1 driven wheel is placed in front of the robot. The loading position will also locate **in the center of robot**.

2.1.2 Wheels selection

a. Driven wheels

There are 2 types of driven wheels for line following mobile robot: castor wheels and ball casters.

Castor wheels	Ball caster wheels
- Capable of working in both indoor/outdoor environment. - Does not require much maintenance. - Lower cost. - Chances of slipping when steering in highspeed (shopping-cart phenomenon).	- Incapable of working in outdoor environment (due to the structure) - Require certain maintenance for working properly. - Higher cost. - Better steering due to having bearing inside.

Table 2.2 Comparison between 2 types of wheels

Some products in the market:

- Castor wheel V1 of Hshop:

Fig 2.7 Castor wheel V1 of Hshop [8]

Properties:

- Material: PP plastic, steel
- Zinc-plated

- Wheel diameter: 25 mm
- Height: 34 mm

- Castor wheel 161POA025P42 of Castor&Wheel VN:

Fig 2.8 Castor wheel 161POA025P42 of Castor&Wheel VN [9]

Properties:

- Material: PP plastic, steel
- Zinc-plated
- Wheel diameter: 25 mm
- Height: 35 mm
- Capacity: 10kg

- Pololu's Ball Caster with 3/8″ Plastic Ball

Fig 2.9 Pololu's Ball Caster with 3/8″ Plastic Ball [10]

Properties:

- Ball diameter: 0.375 in (≈10 mm)
- Height: 10 mm
- Material: ABS plastic
- Weight: 0.8 g

• Plastic ball caster of Hshop:
  - Ball diameter: 12 mm
  - Height: 15 mm
  - Ball material: steel
  - Weight: 30 g

Fig 2.10 Plastic ball caster of Hshop [11]

- Metal ball caster of Hshop:

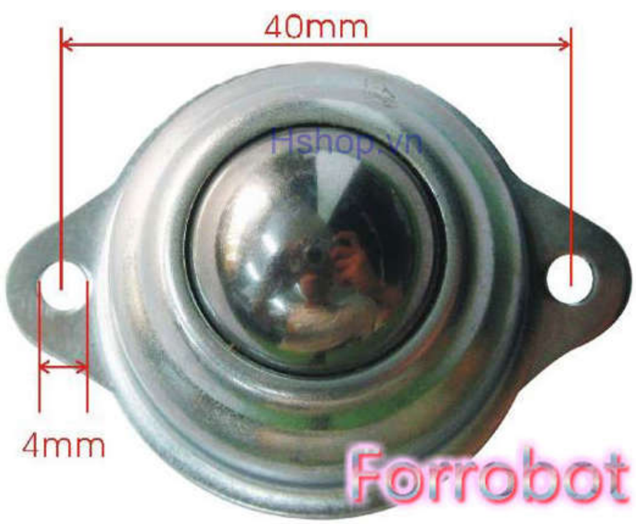


Fig 2.11 Metal ball caster of Hshop [12]

Properties:

- Ball diameter: 20 mm
- Material: steel
- Height: 20 mm

- Weight: 44 g

Our robot mainly work in indoor environment, therefore the ball castor wheel is suitable as our driven wheel.

→ Choose **Metal ball castor of Hshop** [12] as the driven wheel for our robot.

b. Driving wheels

This part represents which size and type of the wheel is chosen for the robot. Types of wheels on the market commonly used for line detectors:

Regular wheel series: V1,2,3,4,6,7,8… with these features:

+ V1, d = 65mm, b = 15mm [13]**:** sturdy, compact, highly aesthetic design, compatible with V1 DC geared motors, suitable for obstacle avoidance robot models, AGVs, self-balancing robot.

Material: Plastic core, rubber wheel.

+ V2, d = 65mm, b = 27mm [14]**:** most used in robot designs, can be attached to a variety of motor shafts. The inner layer of thick elastic foam is suitable for preventing the wheel from collapsing under load. Soft rubber tires ensure friction, optimized surface tread design for better traction.

Core: harden plastic.

Material: PP Plastic, foam, rubber.

+ V3, d = 80mm, b = 30mm [15]: the tire is designed with various spikes on the surface to increase traction with the road surface.

Material: plastic.

+ V4, d = 130mm, b = 60mm [16]: larger wheel size use for vehicles carrying bigger loads than V1, V2, V3.

Material: plastic.

+ V6, d = 85mm, b = 30mm [17]: able to withstand large loads.

Core material: high quality durable plastic.

Tires: high quality durable rubber.

+ V7, d = 96mm, b = 40mm [18]: often used in terrain climbing robots.

Material: high quality durable plastic.

+ V8, d = 115mm, b = 52mm [19]: Core: harden plastic.

Tires: high quality PP plastic, foam.

For driving wheels, there are two parameters need to be considered: diameter and width of the wheel. The bigger the diameter, the higher the maximum speed but the robot is also heavier. The larger the thickness, the higher the traction and this is good for our robot because it helps the robot accelerate faster.

From the list of wheels which can be found in the market, we can see that wheel V2 has the smallest radius so that our robot will have lower center of mass. It also has wider width in compares to wheels V1 to ensure better traction.

→ Choose **Normal wheels V2**: suitable for cars that require compact size, low internal load, high traction on the road, no slipping.

2.2 Electrical selection:

2.2.1 Line following sensor:

Features and requirements need to be taken into consideration when choosing line following sensor:

- Ability to quicky respond to color changes between black and white.

- Low interference.
- Compact in size for installation.

Proposed line following sensor options:

- CMU camera.
- Infrared sensor.
- Photoresistor sensor.

Sensor	**Camera**	**Infrared**	**Photoresistor**
Signal	Line image	Analog and digital	Digital
Complexity of control	Complex	Simple	Simple
Handling interference	Process by program	Can be handled by mechanical structure	Can be handled by mechanical structure
Pros	- Can easily recognize line. - Simple arrangement. - High accuracy.	- Small, easy to install. - Low price. - High accuracy, less affected by environmental light intensity.	- Small, easy to install. - Cheap. - Can recognize high-contrast line easily.

		- Can recognize high-contrast line easily.	
Cons	- Relatively high price. - Needs other sensors to operate.	- Only operate in short distance. - Very sensitive, easily lead to cross-over.	- Affected by environmental light intensity. - Interference caused by environment.

Table 2.3 *Comparisons of sensor types*

⟹ From the above comparisons, TCRT5000 Infrared Reflective sensor is chosen

2.2.2 Load and color sensor:

Features and requirements need to be taken into consideration when choosing color sensor:

- Ability to correctly recognize the color of the cargo.
- Ability to work at close range and changing light conditions.
- Compact in size for installation.

Features and requirements need to be taken into consideration when choosing cargo detecting sensor:

- Ability to detect if the cargo is loaded onto the robot.
- Ability to work at close range and changing light conditions.
- Compact in size for installation.

The color and cargo detecting sensor only work during the loading phase. Both has similar working requirements and is installed in the container of the robot. Therefore, it is optimized to use one sensor that can perform color recognition when the program is called at the loading line, then wait for a short period to confirm the load color.

⟹ From the above conclusion, TCS34725V2 color sensor is chosen.

2.2.3 Power source

Features and requirements need to be taken into consideration when choosing a power source:

- Two separate power supplies are required for the power and control circuits. The power source must has a voltage level suitable for circuit systems.
- The source has a compact size, so that it can be installed on the robot.
- The power source must be capable of providing enough power for the robot to operate for the desired period of time.

Proposed power source options:

- Use DC power converted from AC
- Use DC power from battery

Option	**Advantage**	**Disadvantage**
Switched-mode power supply (SMPS)	- Meet the requirements for current and voltage. - Can be used as 5V and 12V power source separately	- Too big and bulky to be installed directly on the robot. - Can be used stationarily and wired to the robot, but it will restrict the movement.

Lithium Ion battery (Li-ion)	- Higher capacity than LiPo battery. - Environmentally friendly.	- Highly flammable. - Low lifespan.
Lithium Polymer battery (LiPo)	- Light weight, can be easily installed on the robot. - High capacity, strong discharge current. - Capable of recharging for further usage.	- May be dangerous if not being charged properly. - Battery lifespan is greatly shortened after many cycles of charging and discharging.

Table 2.4 *Comparisons of power sources*

⟹ From the above comparisons, **Li-ion battery** is chosen.

2.3 Controller selection:

2.3.1 Control structure:

Centralized control:

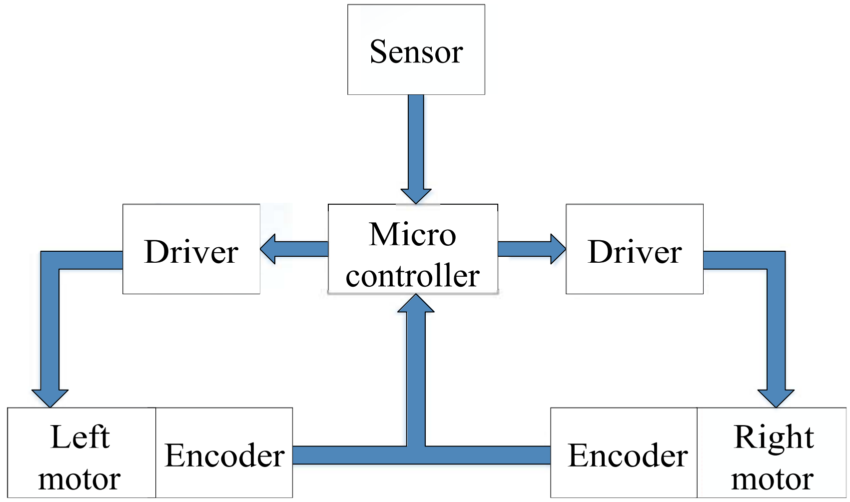


Fig 2.12 Structure of centralized control

The only microcontroller simultaneously receives and processes signals from sensors, receives and processes signals from 2 encoders, executes main program, calculates control values and transmits to two motors. Most of the actual Robots and race cars often use this control structure.

Decentralized control:

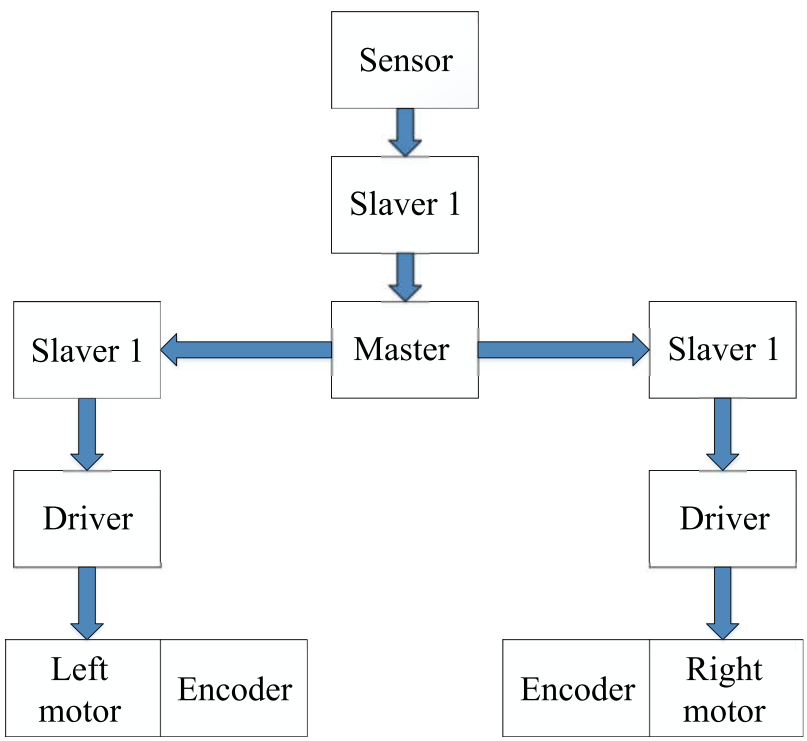


Fig 2.13 Structure of decentralized control

A microcontroller used as a computation master for the main controller program. The remaining slavers use other microcontrollers, performing separate tasks: receive and process signals from the sensor, calculate the relative position of the vehicle to the line and transmitting it to the master, receiving signal from the encoder, calculate the control algorithm for the motor, make sure the motor works according to the master requirements, etc. Signals are exchanged between microcontrollers have many different standards such as I2C, CAN... This structure reduces the amount of computation for the master and allow the robot to perform multiple functions at the same time.

	Centralized control	Decentralized control
Number of microcontrollers	One	Many
Processing speed	Slow	Fast
Advantages	Low space required Low weight Low cost	Easy to program and developed
Disadvantages	Difficult and complicates to code and debug	Require more space High weight High cost

Table 2.5 Comparison between Centralized control and Decentralized control

→ Choose decentralized control because we can test many components at the same time as a team.

2.3.2 Sensor algorithms:

Comparison Algorithm: the signal is based on the switching state of the sensor and then used to infer the position of the robot, the response speed is fast. Return signal is a sequence of binary numbers (0 and 1).

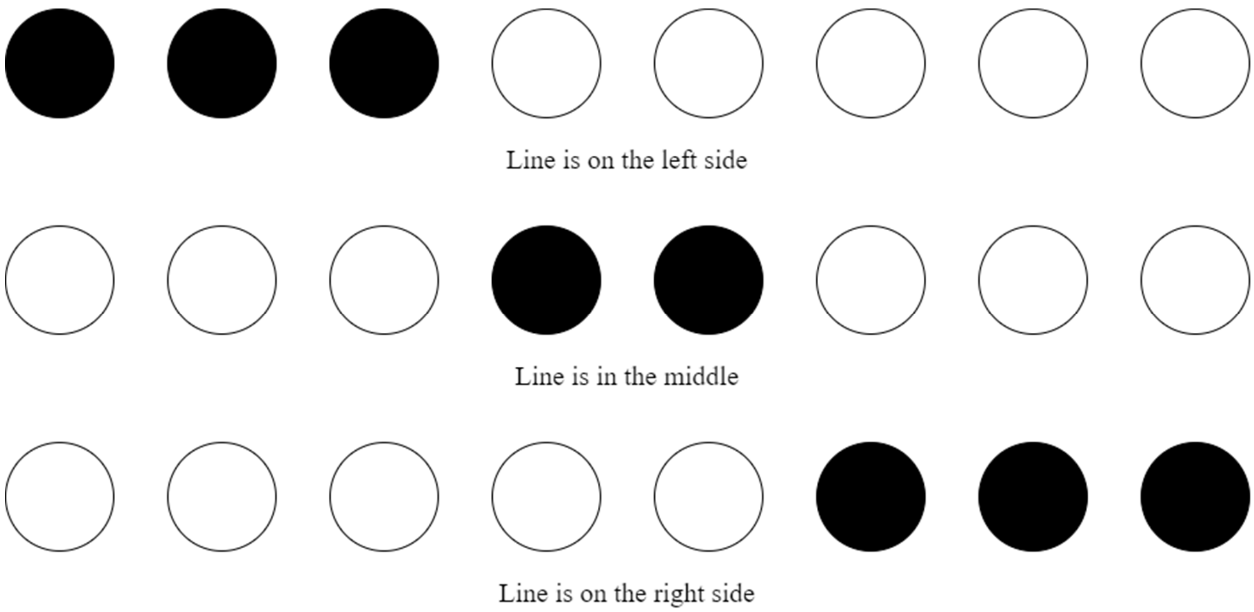


Fig 2.14 Sensor value read to digital

The simplest form of this algorithm only uses 2 central sensors (SL and SR) located on the sensor bar.

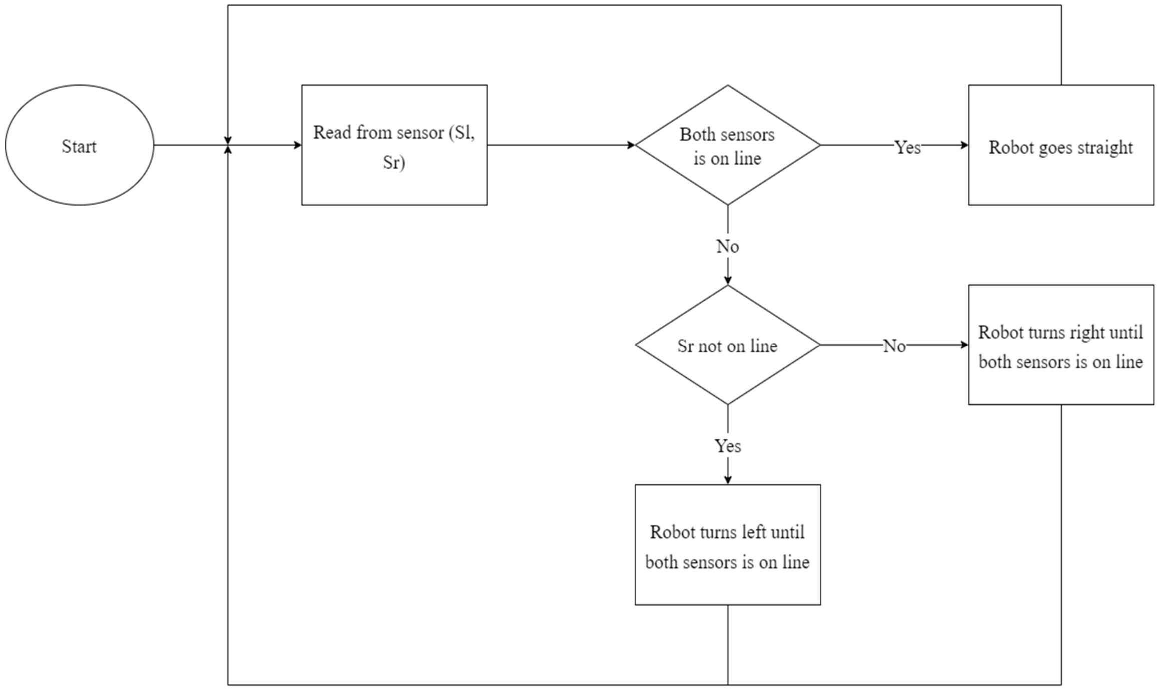


Fig 2.15 Comparison algorithm flowchart

Interpolation Algorithm: The readings are performed through approximation to find the position of the line, then find out the position of the robot with high accuracy. Return signal is a number, after being processed through the approximation formula (weighted, order 2).

- ***Second-order approximation algorithm:***

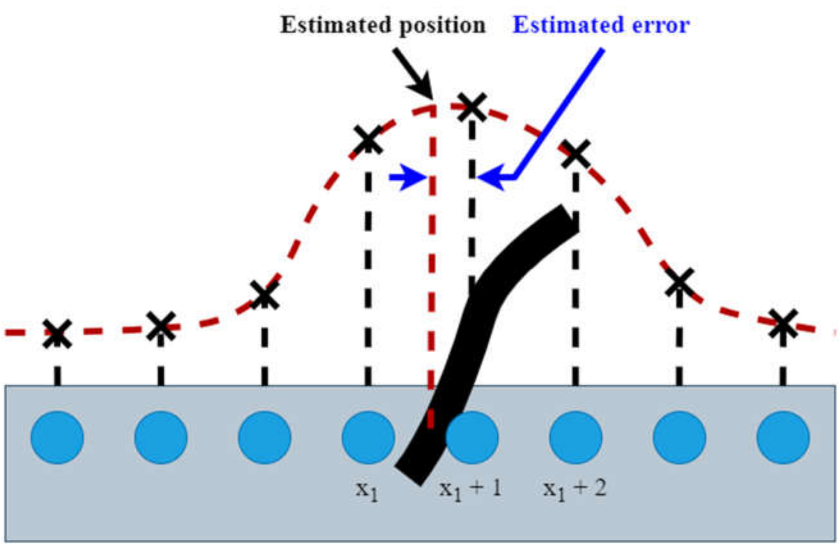


Fig 2.16 Second-order approximation algorithm

Suppose the coordinates of the leftmost sensor are -3 and the distance between the sensor is 1.

The value of the sensor circuit will be larger when the sensor receives the signal from the black line.

The coordinates of the three middle sensors are $x_1, x_1 + 1, x_1 + 2$ and the response profile of the sensor value in the range $[x_1, x_1 + 2]$ can be approximated to a second-order curve.

The relationship between the position coordinates of the sensors and the output value is shown as follows:

$$y_1 = ax_1^2 + bx_1 + c$$
$$y_2 = a(x_1 + 1)^2 + b(x_1 + 1) + c$$
$$y_3 = a(x_1 + 2)^2 + b(x_1 + 2) + c$$

The coordinates at the output value of the quadratic curve reach the maximum value and are considered as the true position of the line. Applying Langrange interpolation, we have:

$$x = \frac{-b}{a}, a = \frac{y_1 + y_3 - 2y_2}{2}, b = y_2 - y_1 - 2ax_1 - a$$

The coordinate at the center sensor position is 0 and 1. Therefore, the error e between the line and the center position of the robot is:

$$e = 0 \pm x = \pm x$$

- ***Weighted approximation algorithm:***

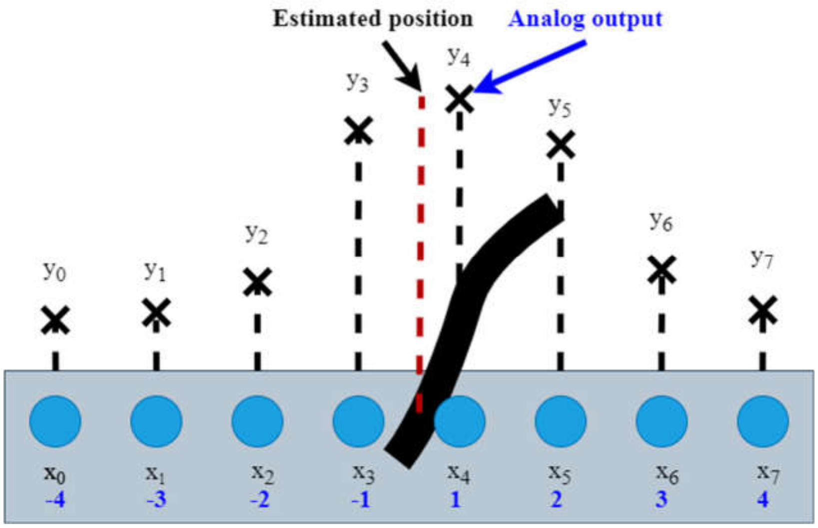


Fig 2.17 Weighted approximation algorithm

This technique is used in the defuzzification of fuzzy systems.

Suppose the coordinates of 8 sensors are respectively $x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7$ and the corresponding analog signal values are $y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7$. The estimated position of the line can be calculated using the centroid formula:

$$x = \frac{\sum_{i=0}^{7} x_i y_i}{\sum_{i=0}^{7} y_i} = \frac{4(y_7 - y_0) + 3(y_6 - y_1) + 2(y_5 - y_2) + (y_4 - y_3)}{\sum_{i=0}^{7} y_i}$$

→Choose weighted approximation algorithm, because it is proven to be better in according to the reference and the calculation is faster than second order interpolation. [20]

CHAPTER III: MECHANICAL DESIGN

3.1 Parameters of design

Technical specifications we need to calculate:

- Power and torque of motors.
- Dimensions of the robot, distance between driving wheels.
- Tolerance and motor mount.

Designing process:

- Motor calculation and choosing.
- Checking turning constraints.
- Designing chassis and motor mount.
- Calculating tolerance.
- Mechanical drawing.

3.2 Motor selection

a. Motor calculation

Input values:

- Robot with three wheels configuration: 2 driving wheels behind, one driven wheel in front.
- Friction coefficient (between wheels and the map): $\mu_t = 0.8$ [21]
- Safety factor: $s = 1.3$ [22]

Parameter	**Values**
Maximum speed	$v_{max} = 0.3\ m/s$
Approximate total mass	$M = 4\ kg$
Desire speed up time	0.25 s
Starting acceleration	$a = 1.2\ m/s^2$
Wheels radius	$r_w = 0.0325\ mm$
Wheel mass	0.02 kg
Friction coefficient (sliding)	$\mu_t = 0.8$
Rolling coefficient	$C_{roll} = 0.015$
Safety factor	$s = 1.3$

Table 3.1 Table of input values for choosing motor

Output values:

Velocity and angular velocity:

$$v_{max} = 0.3\ m/s$$

$$v_{max} = \omega_{max}\, r_w \rightarrow \omega_{max} = \frac{0.3}{0.0325} = 9.2308\ rad/s$$

$$\rightarrow n = \frac{60\omega_{max}}{2\pi} = 88.1474\ rpm$$

Power of motor:

We want the robot to achieve v_{max} in 0.25 seconds

$$\rightarrow a = \frac{0.3}{0.25} = 1.2\ m/s^2$$

Assume robot weight is 2kg, load is 2kg.

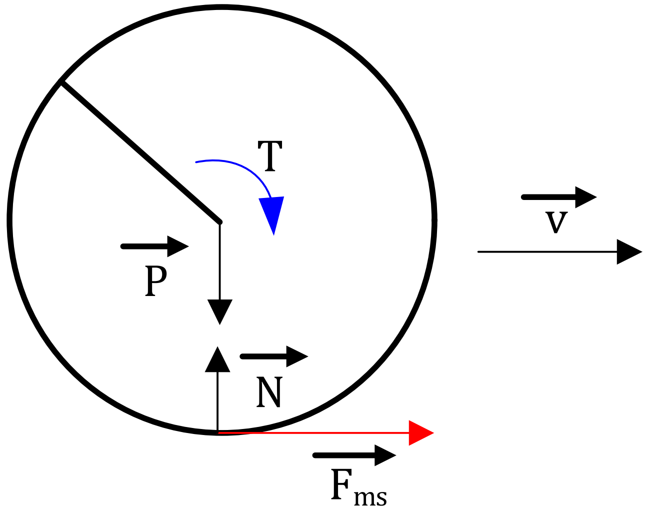


Fig 3.1 Force evaluation

Moment of inertia of the wheel (assume that it is a solid plate):

$$I = \frac{1}{2} m R^2 \ (4.1)$$

Assume the load is distributed evenly between 2 driving wheels, we can calculate the force acts on one wheel:

$$P = \left(m + \frac{M}{2}\right) g \ (4.2)$$

Apply the Second Law of Newton:

- Equation of torque apply on the wheel:

$$T - F_f \times R = I\gamma$$

with:

$$\gamma = \frac{a}{R}$$

Then:

$$T = F_f \times R + \frac{Ia}{R} \ (4.3)$$

- Apply the Third Law of Newton on the y-axis (and from equation 4.2):

$$N = P = \left(m + \frac{M}{2}\right) g \ (4.4)$$

- The force acts on x-axis:

$$F_f = C_{roll}\left(m + \frac{M}{2}\right) g \ (4.5)$$

- Subtitute equation (4.1), (4.4), (4.5) into equation (4.3), we get the torque apply on the wheel when running:

$$T = C_{roll}\left(m + \frac{M}{2}\right) gR \ + \ \frac{1}{2} mRa \ (4.6)$$

$$T = 0.015 \times \left(0{,}02 \ + \ \frac{4}{2}\right) \times 9.81 \times 0.0325 \ + \ \frac{1}{2} \times 0.02 \times 0.0325 \times 1.2$$
$$= \ 0.07917 \ Nm$$

- Condition for wheels not slipping on the map:

$$F_f \le F_{rf} = \mu_t N = \mu_t \left(m \ + \ \frac{M}{2}\right) g \ (4.7)$$

Substitute (4.7) into (4.3) we have:

$$T \le \mu_t \left(m \ + \ \frac{M}{2}\right) gR + \frac{1}{2} mRa$$

The largest torque can apply on the wheel to ensure traction:

$$T_s = \mu_t \left(m \ + \ \frac{M}{2}\right) gR + \frac{1}{2} mRa \ (4.8)$$

$$T_s = 0.8 \times \left(0.02 + \frac{4}{2}\right) \times 9.81 \times 0.0325 + \frac{1}{2} \times 0.02 \times 0.0325 \times 1.2 = 0.5156 \ Nm$$

- Angular velocity of the motor:

$$\omega_0 = \frac{v_{max}}{R} \ (4.9)$$

$$\omega_0 = 9.231\ rad/s$$

- Motor revolution numbers:

$$n_0 = \frac{60\omega}{2\pi} = 88.1496\ rpm$$

- Minimum power of motor:

$$P_{min} = \omega T \times 1.3 = 0.07917 \times 9.231 \times 1.3 = 0.95\ W$$

- Maximum power of motor:

$$P_{max} = \omega T \times 1.3 = 0.5156 \times 9.231 \times 1.3 = 6.2\ W$$

Safety factor for both speed and torque:	$s = 1.3$
Maximum speed of motors:	$\omega = 9.231\ rad/s$
Motor revolution numbers:	$n_0 = 88.1496\ rpm$
Working torque of the motor:	$T = 0.07917\ Nm$
Maximum torque that cause slip:	$T_s = 0.5156\ Nm$
Power of motor:	$0.95 \leq P \leq 6.2\ (W)$

From calculation, we choose DC Servo GM25-370 DC Geared Motor with following specifications:

Fig 3.2 DC Servo GM25-370 DC Geared Motor [23]

Properties	**Values**
Gear ratio	34:1
No-load current	150 mA
Loaded current	750 mA
No-load rotational speed	250 rpm
Rated rotational speed	140 rpm
Loaded torque	4.3 kg.cm
Maximum torque	5.2 kg.cm
Encoder	374 pulse/rev

Table 3.2 Properties of DC Servo GM25-370 DC Geared Motor

b. Choosing motor mount:

Motor mount can be designed specifically for the robot or can be chosen from various choices in the market. Due to the fact that we should lower the cost as much as possible, our group decide to use the motor mount that is sold in the market.

Fig 3.3 25mm DC Geared Motor Mounting Bracket of Hshop [24]

3.3 Calculation of robot dimension

Kinematic model of the robot is represented as shown:

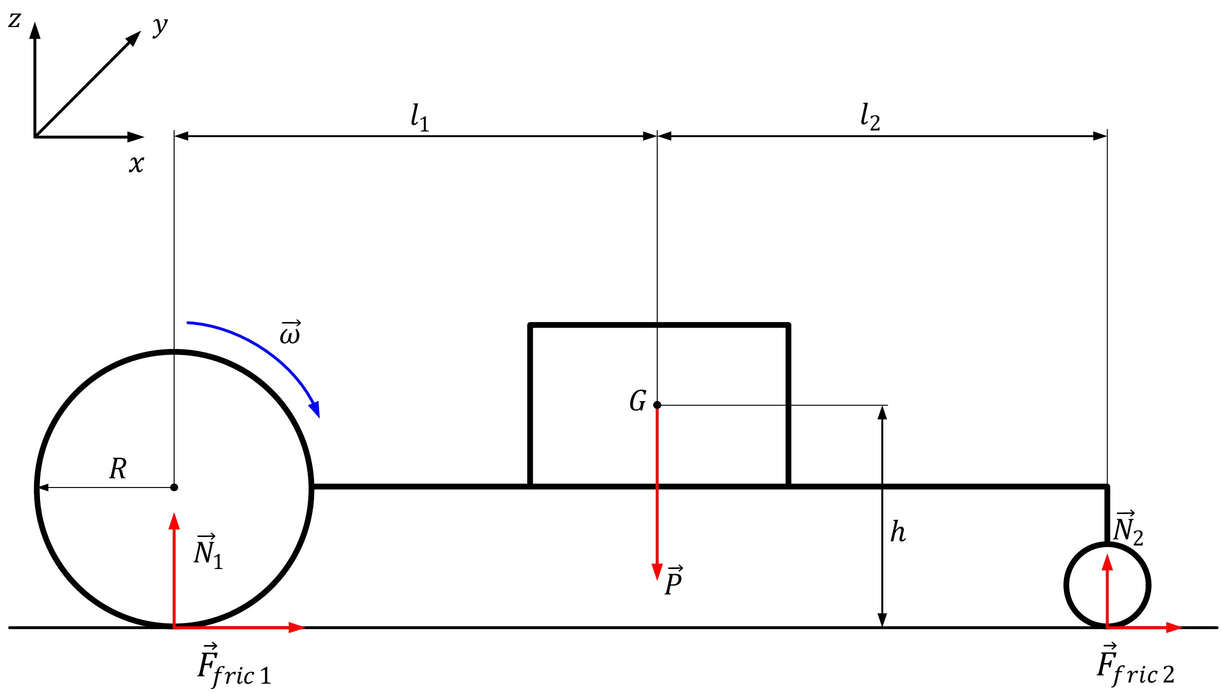


Fig 3.4 Kinematic model of the robot

Values in the model:

- G is the center of the robot.
- F_{fric1}, F_{fric2} are friction of rear wheels and front wheel, respectively.
- N_1, N_2 are normal forces applied on rear wheels and front wheel, respectively.
- l_1, l_2 are distances between the center of robot to rear wheels and front wheel, respectively.
- l is the wheel base (distance between the middle point of 2 rear wheels to front wheel)

Base on Newton's Second Law and equilibrium equations of the model, we have these set of dynamic equations:

$$\begin{cases} \sum F_x = ma \\ \sum F_z = 0 \\ \sum M_G = 0 \end{cases} \Leftrightarrow \begin{cases} F_{fric1} + F_{fric2} = ma & (1) \\ N_1 + N_2 = mg & (2) \\ N_1 l_1 - N_2 l_2 - F_{fric1} h - F_{fric2} h = 0 & (3) \end{cases}$$

We have:

$$(mg - N_2)l_1 - N_2\, l_2 - mah = 0 \Rightarrow mgl_1 - N_2 l = mah$$

$$\Rightarrow a = \frac{mgl_1 - N_2 l}{mh}$$

From that we have these relationships between acceleration, length and width of the robot:

- The smaller the value of h, the bigger the value of a.
- The robot accelerates faster as the length from center of robot G to 2 rear wheels l_1 is bigger.

We must ensure that the front wheel of the robot coincide with the platform in the whole accelerating process. Therefore, we have the condition: $N_2 \geq 0$.

From previous set of equations we have:

$$\begin{cases} (1),(3) \Leftrightarrow N_1 l_1 - N_2 l_2 = mah \\ \quad (2) \quad \Leftrightarrow \quad N_1 + N_2 = mg \end{cases} \Leftrightarrow \begin{cases} N_1 = m\dfrac{gl_2 + ah}{l_1 + l_2} \\ N_2 = m\dfrac{gl_1 - ah}{l_1 + l_2} \end{cases}$$

$$\rightarrow N_2 = m\frac{gl_1 - ah}{l_1 + l_2} \geq 0 \Rightarrow gl_1 - ah \geq 0 \Rightarrow \frac{l_1}{h} \geq \frac{a}{g}$$

Substitute:

Gravitational acceleration: $g = 9.81\ m/s^2$

Acceleration of the robot: $a = 1.2\ m/s^2$

$$\rightarrow \frac{l_1}{h} \geq \frac{1.2}{9.81} = 0.1223 \rightarrow l_1 \geq 0.1223 \times h$$

$$\rightarrow Choose\ l_1 = 80mm$$

l_2 is the driven parameter so it should be adaptive to the real design.

a. Non-sliding condition:

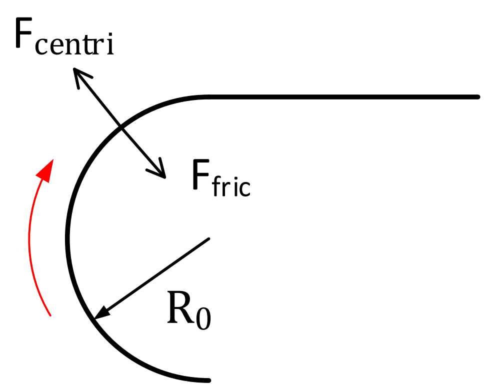


Fig 3.5 Smallest turning radius

$F_{fric} \geq F_{centri}$ (considering both driving wheels)

$$\rightarrow \mu_t\, m_{total}\, g \geq \frac{m_{total}\, v^2}{r} \rightarrow \mu_t\, g \geq \frac{v^2}{r}$$

$$\rightarrow v \leq \sqrt{\mu_t g R_0} = \sqrt{0.8 \times 9.81 \times 0.5} = 1.9809\, m/s$$

Our velocity satisfied the condition.

b. Condition for the robot to not overturn:

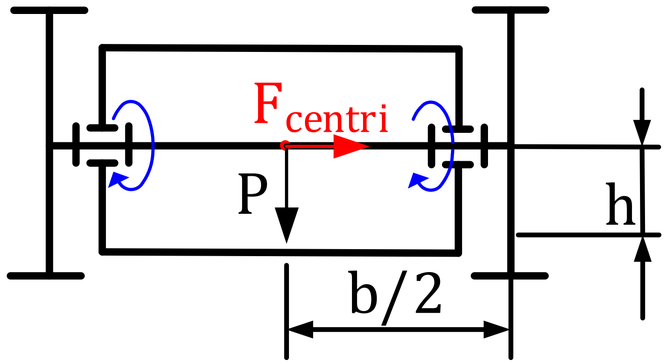


Fig 3.6 Dimension of the wheel base

To ensure the robot is not overturn, torque apply on the center of robot G caused by the gravity must larger than that of the centrifugal force. We have the condition:

$$P\frac{b}{2} \geq F_{centri} \times h$$

$$\rightarrow mg\frac{b}{2} \geq \frac{mv_{max}^2}{R} \times h$$

$$\rightarrow h \leq \frac{Rgb}{2v_{max}^2} = \frac{0.5 \times 9.81 \times b}{2 \times 0.3^2} = 27.25 \times b$$

This condition is easily applied as the coefficient of b is relatively large. Assume that the wheel track is $100\,mm = 0.1\,m$, equivalent to the width of the load, then $h \leq 27.25 \times 0.1 = 2.725\ (m)$. The highest possible point of the center of robot measured from the platform is far higher than the chosen value.

Choose $b = 0.16\,m \rightarrow h \leq 4.36\,m$

3.4 Structure simulation:

We choose acrylic plastic as the material of the robot as we want to reduce the cost.

Our main base and upper base are made from 5mm width acrylic sheet. The load is approximately 2kg, so there would be a force of 26N (20N with a safety factor of 1.3) distributes evenly on the surface of the upper base. For the main base, there will be a force of approximately 52N (40N with a safety factor of 1.3), equal the total mass of the whole robot.

Parameters of the simulation:

- Displacement less than 2mm (base on the working distance of the sensors are 8-10mm from the ground).
- Safety factor is 1.3

From the design, we tested the base and upper base using the simulation of Solidwork application:

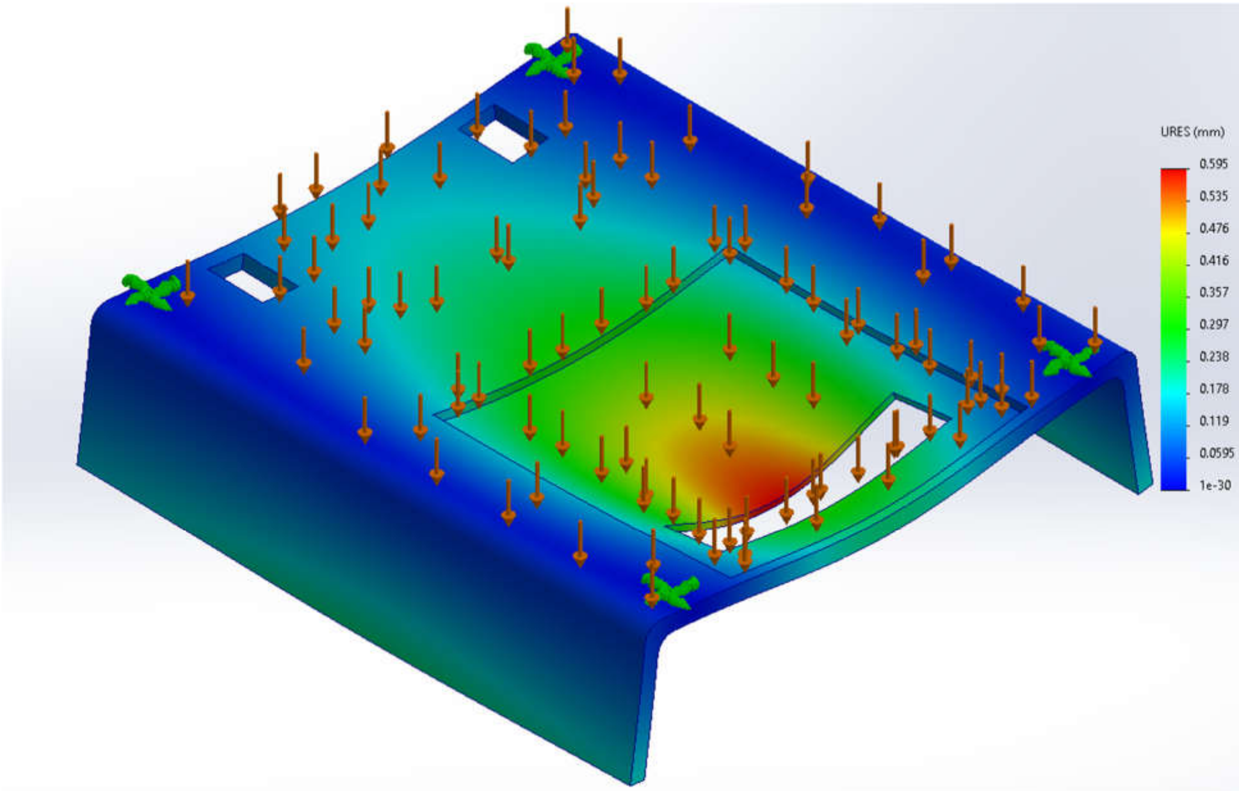


Fig 3.7 Simulation for displacement of the upper base

For the upper base, fixed positions are four standoffs.

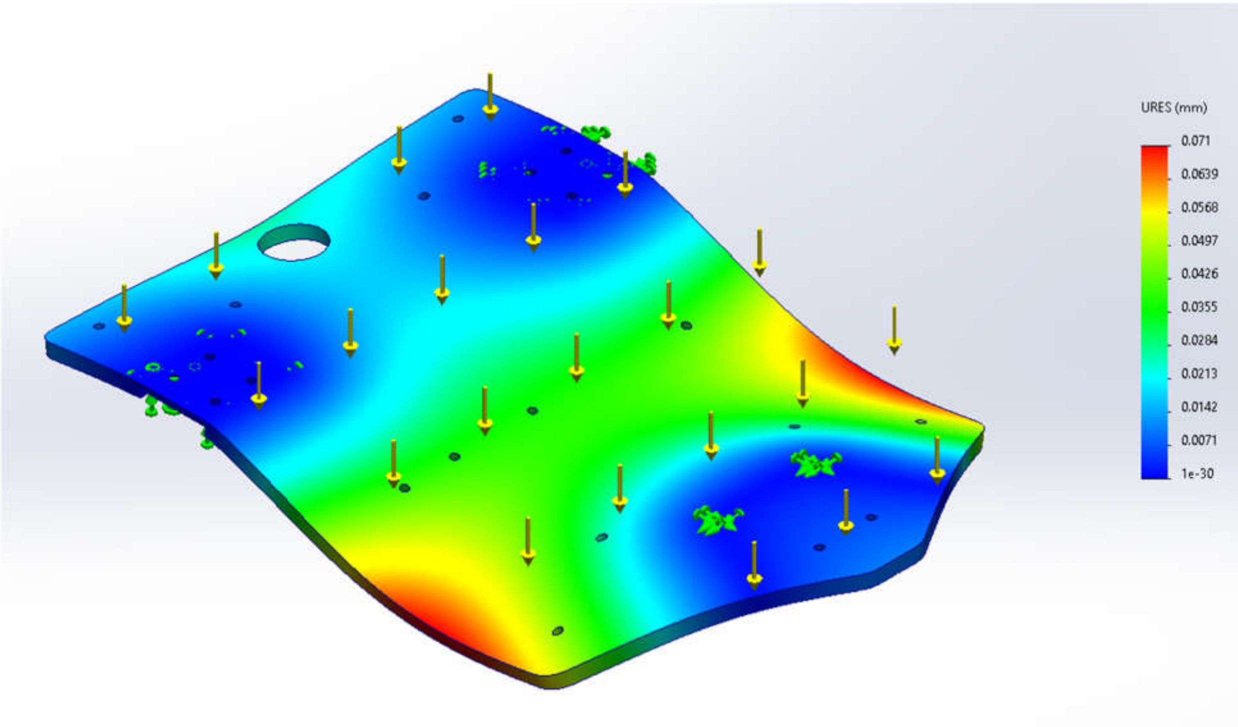


Fig 3.8 Stress simulation for the base

For the main base, fixed positions are two mounting brackets and the driven wheel.

From the simulation, we can observe that the maximum displacement in both bases are still less than 1mm and have small impacts on the robot structure. We conclude that 5mm acrylic sheet is satisfied with the test.

Parameters	***Value***
Length	270 mm
Width	230 mm
Height	105 mm
Height from sensor to the ground	≈11 mm
Distance between front wheel and rear wheels	96 mm

Table 3.3 Summary of robot dimension

CHAPTER IV: ELECTRICAL DESIGN

Designing process:

- Calculate sensors parameters
- Calculate power supply
- Calculate microcontroller peripherals.
- Select driver for motor
- Select color sensor

4.1 Sensor design

4.1.1 TCRT5000 specification

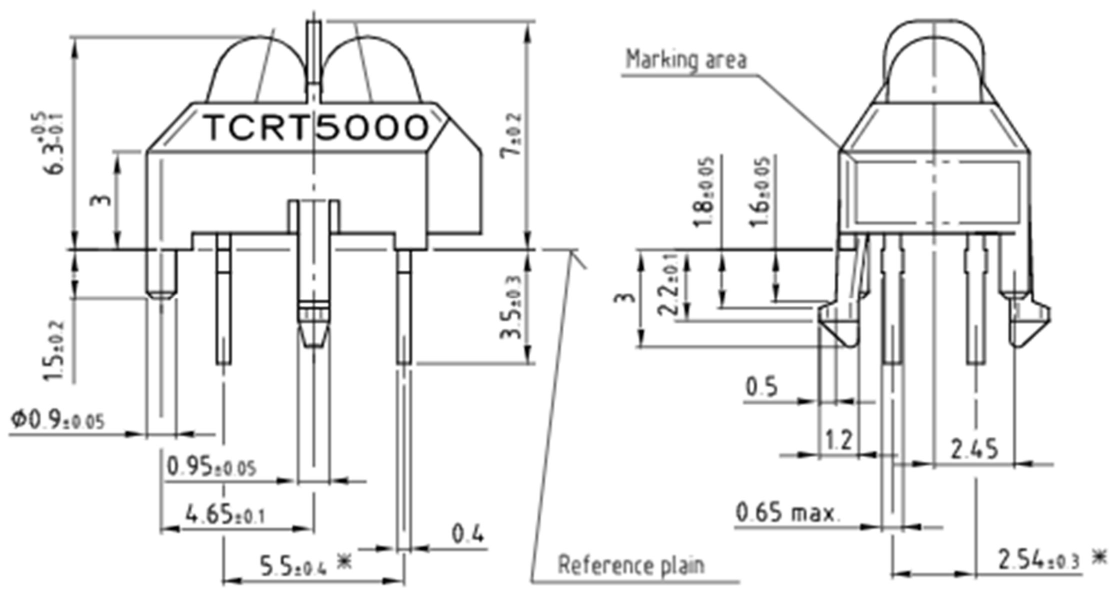


Fig 4.1 TCRT5000 dimension

Parameter	Value	Unit
Sensor dimensions	10.2 × 5.8 × 7	mm
Operating range	0.2 - 15	mm
Emitting wavelength	950	nm
Emitter angle	16	degree

Collector angle	30	degree
Collector current I_C	100	mA
Forward current I_F	60	mA

Table 4.1 TCRT5000 specification

4.1.2 Resistor calculation

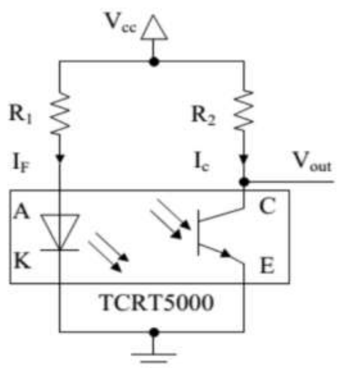


Fig 4.2 TCRT5000 electrical structure

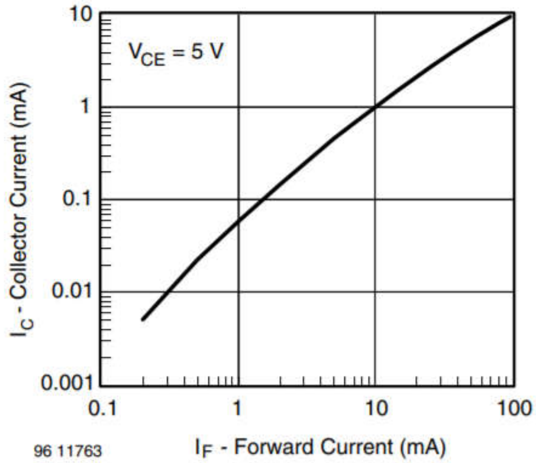
From the TCRT5000 datasheet, we have:

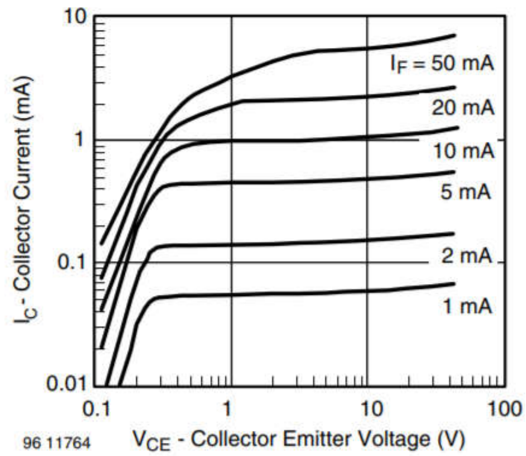
- $V_F = V_{AK} = 1{,}25\ (V)$ represents the voltage passing through diode.
- $I_F = 20\ (mA)$

Resistance value of resistor R_1 can be calculated as follow:

$$R_1 = \frac{V_{cc} - V_F}{I_F} = \frac{5 - 1.25}{0.02} = 187.5\ (\Omega)$$

Choose $R_F = 220\ (\Omega)$ => $I_F = \frac{5-1.25}{220} = 17\ (mA)$ ≤ 60 mA





a) Relationship between IF and IC *b) Relationship between IF, IC and VCE*

Fig 4.3 Graphs of relationship between current and voltage passing through LED

Using *Figure 4.2 a)* and $I_F = 17\ (mA)$, we can infer that $I_c = 1\ (mA)$. Then, using $I_c = 1\ (mA)$ and $I_F = 20\ (mA)$ in *Figure 4.2 b)*, we can infer the value $V_{CE} = 0.6\ (V)$. Then, resistance value of resistor R_2 can be calculated as follow:

$$R_2 = \frac{V_{CC} - V_{CE}}{I_C} = \frac{5 - 0{,}6}{1.10^{-3}} = 4400\Omega$$

Choose $R_2 = 4700\Omega$

4.1.3 Sensor placement

There are two ways to place the sensor: horizontal and vertical. When going from a white background to a black background, the sensor has to move a distance X_d before its analog value is finally determined.

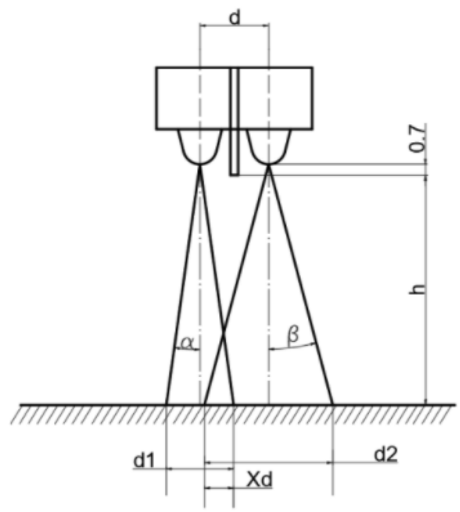


Fig 4.4 Transceiver area of sensor TCRT5000

We can choose the sensor layout so that the X_d value is small to ensure the respond time of the sensor. With X_d is the interference region of light between the collector and emitter.

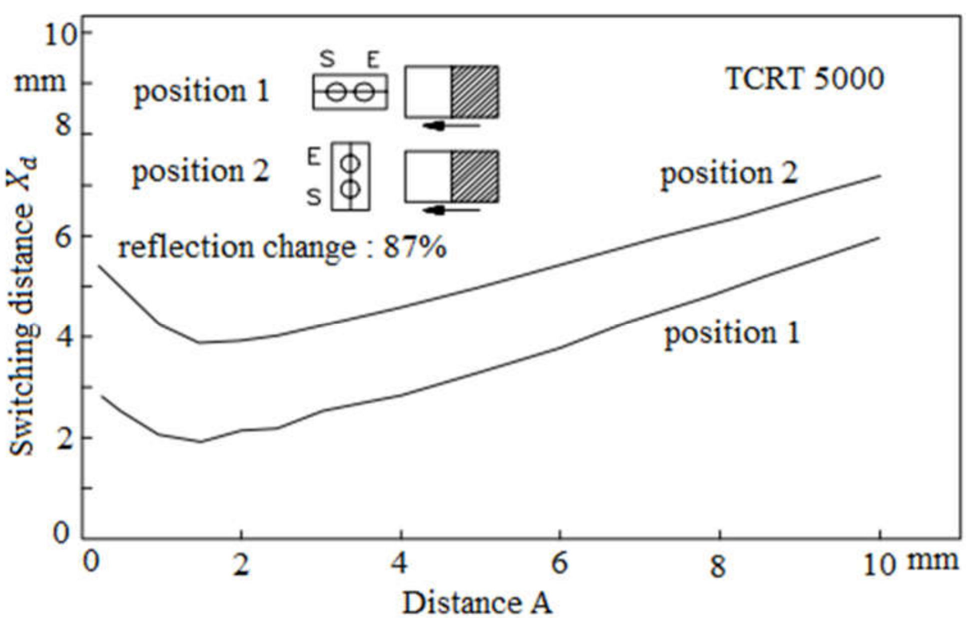


Fig 4.5 Graph of relationship between sensor placement and X_d

According to ***Figure 4.5***, we can easily see that in position 1 (horizontal layout), the switching distance X_d is smaller than that of position 2 (vertical layout) in the same distance A. The smaller the switching distance X_d, the more sensitive the sensor will operate since the transceiver width is smaller. However, the larger the switching distance X_d, the more accurate the result will be. If we choose the horizontal layout, further calibrations or adjustments in design is needed so that the center point of the transceiver zone aligns with the center of the sensor. In the case of vertical layout, no further calibration step is needed.

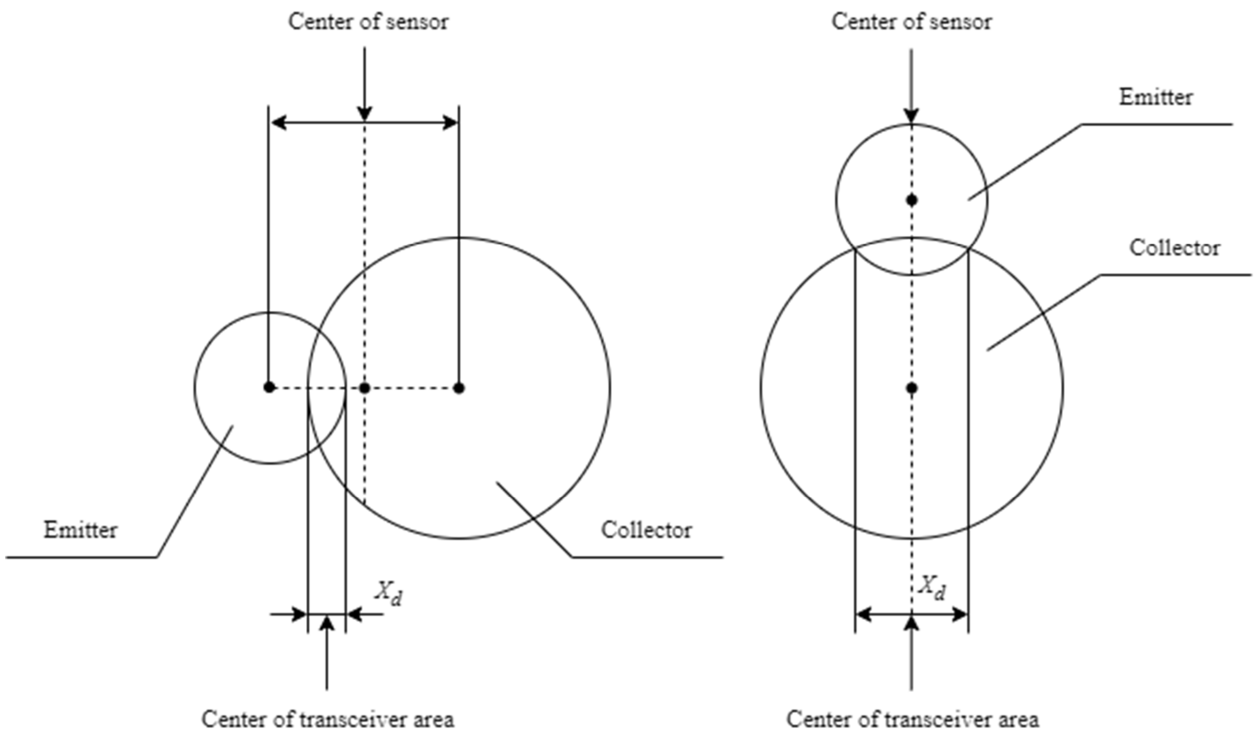


Fig 4.6 Comparison of distance X_d and center of transceiver area in horizontal and vertical layout of sensor

In conclusion, the vertical layout is chosen for accuracy and simplicity

4.1.4 Sensor height with respect to ground

To ensure that the phototransistor can receive the signal from the transmitter led when the robot follows the line, there must be an intersection between the transmitter and receiver areas.

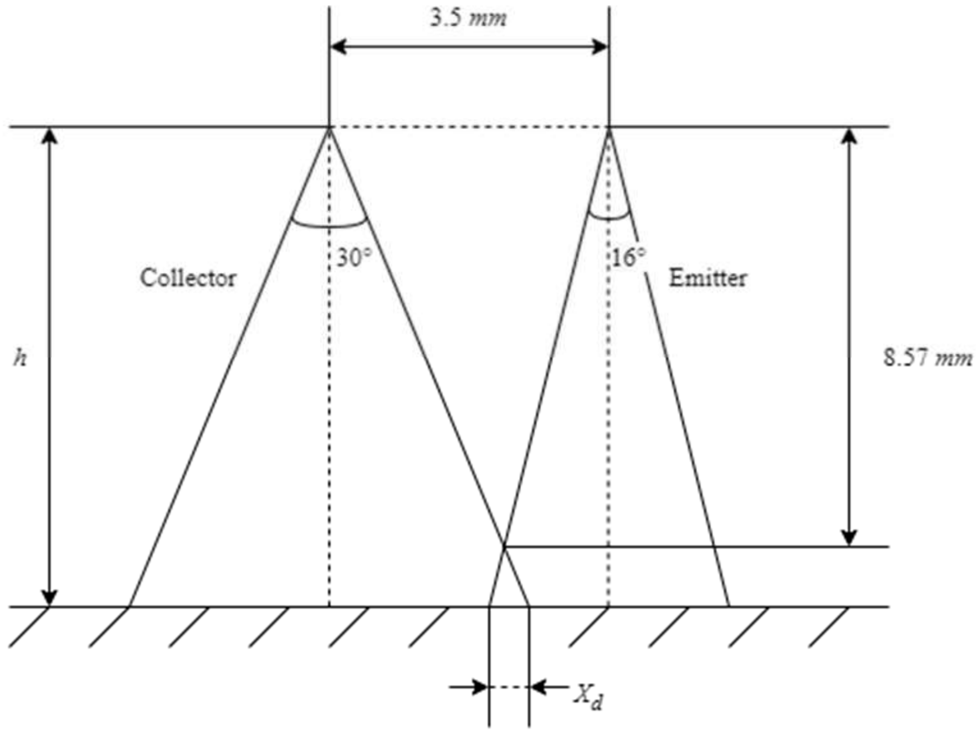


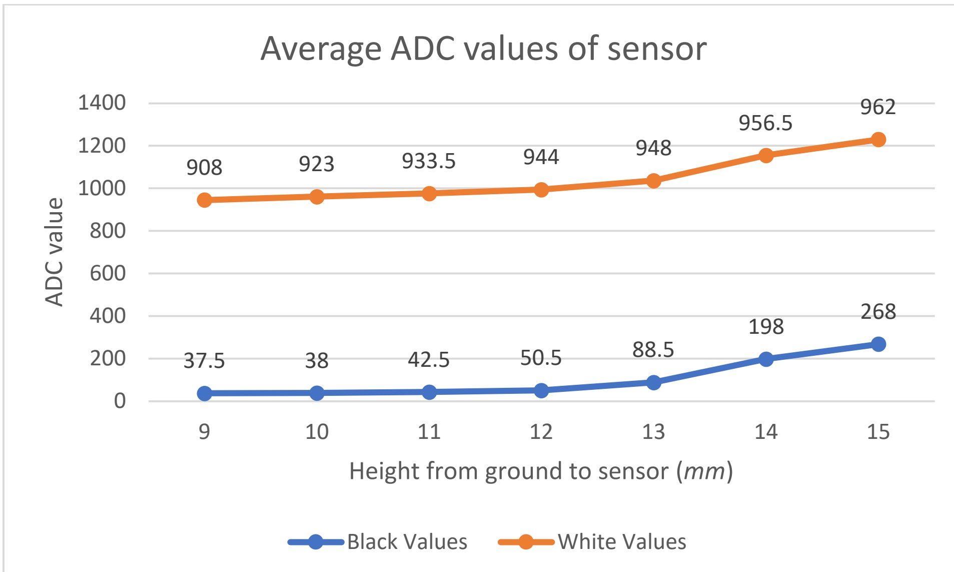
Fig 4.7 TCRT5000 geometric diagram for calculation

The operating range of TCRT5000 sensor is 0.2-15 mm. For the sensor to operate properly, X_d must have a positive value. Calculating from the datasheet diagram, the value of X_d in the height range from 0.2 to 8.57 mm is negative, thus will be omitted. Height from 8.57 to 15 mm will result in value X_d ranging from 0 to 9.16 mm.

From the above information, we will conduct an experiment to choose the appropriate height for sensor in the range 9-15mm that can maximize the difference between black and white background.

First measurement				**Second measurement**			
h (mm)	***White value***	***Black value***	***Difference***	***h (mm)***	***White value***	***Black value***	***Difference***
9	38	902	864	9	37	914	877
10	39	918	879	10	37	928	928
11	43	936	932	11	42	931	889

12	51	943	892	12	50	945	895
13	88	949	861	13	89	947	838
14	197	956	759	14	201	957	756
15	269	961	692	15	267	963	696



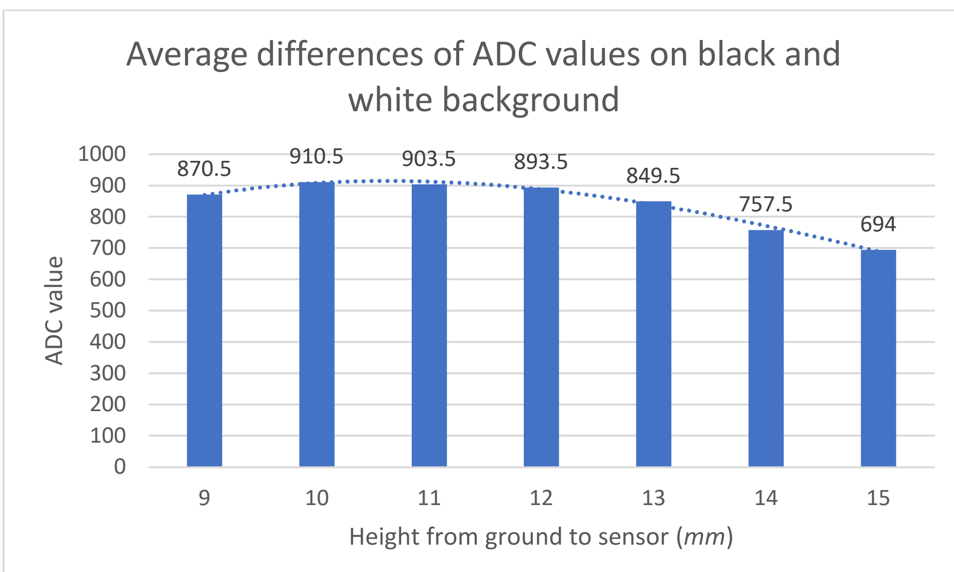


Fig 4.8 Measurement of sensors

From the above graphs, we can see that at $h = 10\ (mm)$, the analog signals at center of the black line and white ground have the largest difference of 910.5. Therefore,

choosing that height to place the sensor would result in the most accurate and noticeable feedback when the robot transverse from white to black ground and vice versa.
In conclusion, we choose the height from ground to sensor $h = 10\ (mm)$.

4.1.5 Distance between sensor

Using $h = 10\ (mm)$ and ***Figure 4.8***, we can calculate the minimum distance between two sensor that can meet the requirements of no interference as follow:

$$l_{min} = 2R = 2 \times (h + 0{,}7) \times tan15^0 \approx 5.73\ (mm)$$

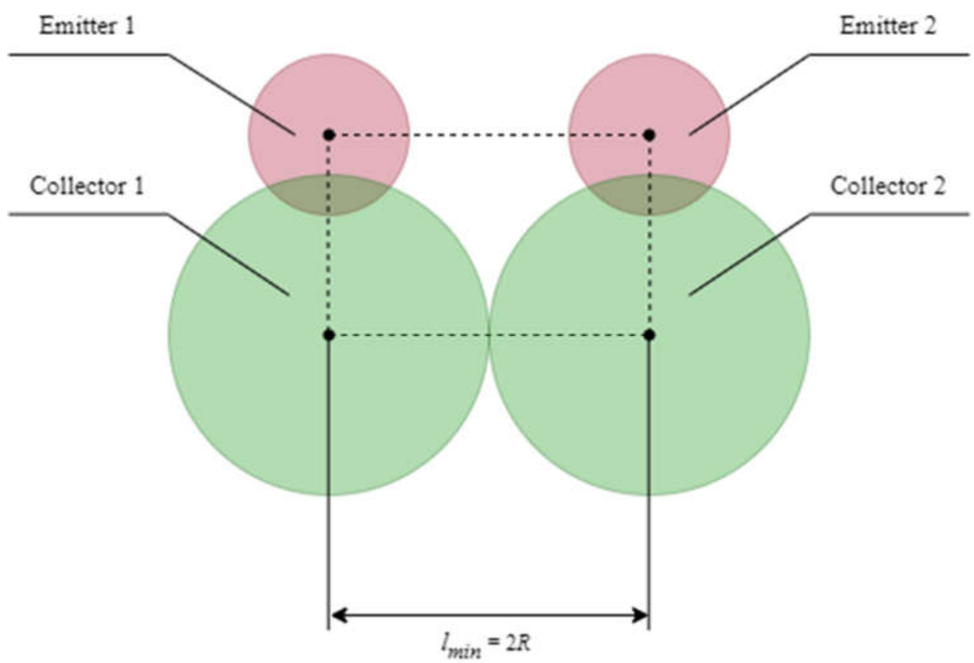


Fig 4.9 The minimum distance between two sensors

Theoretically, the minimum distance between two sensors can be as small as $5.73\ mm$. However, the width of TCRT5000 according to the datasheet is $d = 5.8\ mm > 5.73\ mm$. Therefore, the true minimum distance should be adjusted to $d_{min} = d = 5.8\ mm$

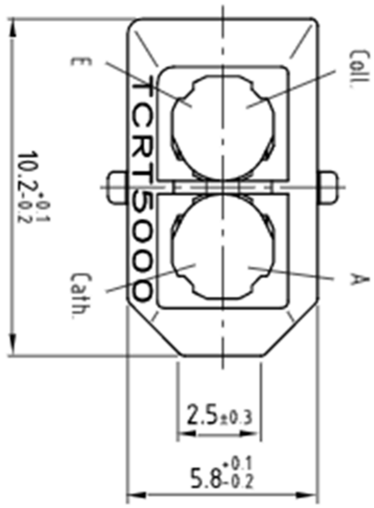


Fig 4.10 Dimension of a sensor

With the requirement $W_{line} = 26\ (mm)$, there are two cases to consider:

- Case 1: 2 sensors lie in the black zone of the line.
- Case 2: 3 sensors lie in the black zone of the line.

Considering Case 1:

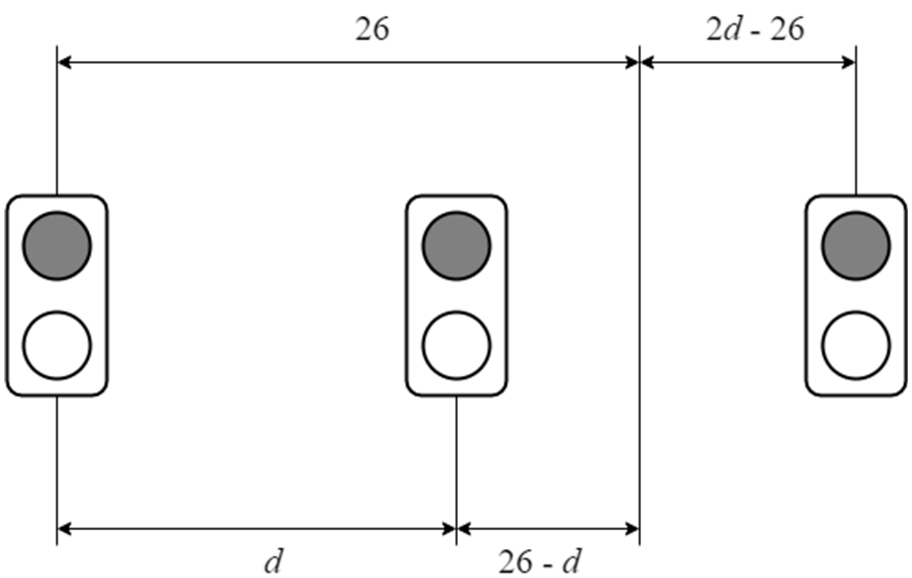


Fig 4.11 Case 1 illustration

When the robot moves right in range of $26 - d\ (mm)$, there will always be 2 sensors lie in the black zone of the line and their analog feedback will be the same. Alternatively, when the robot moves left in range of $2d - 26\ (mm)$, there will be only one sensor lies on the the line and one analog feedback, meaning ambiguity. Thus, to prevent this problem, we must choose distance d between two sensors such that $f_1 = 26 - d$ and $f_2 = 2d - 26$ simultaneously reach their minimum.

Since f_1 is a decreasing linear function and f_2 is an increasing linear function, the only solution for the above problem is $f_1 = f_2$. From that, we can calculate $d \approx 17\ (mm)$.

Considering Case 2:

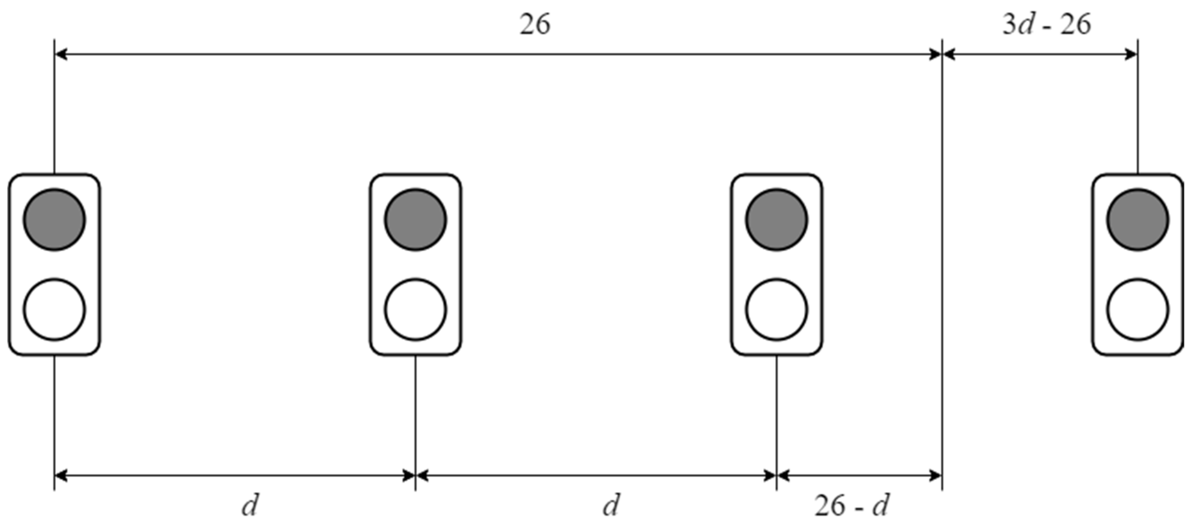


Fig 4.12 Case 2 illustration

When the robot moves right in range of $26 - 2d\ (mm)$, there will always be 3 sensors lie in the black zone of the line and their analog feedback will be the same. Alternatively, when the robot moves left in range of $3d - 26\ (mm)$, there will be only two sensors lie on the the line and one analog feedback, meaning ambiguity. Thus, to prevent this problem, we must choose distance d between two sensors such that $f_1 = 26 - 2d$ and $f_2 = 3d - 26$ simultaneously reach their minimum.

Since f_1 is a decreasing linear function and f_2 is an increasing linear function, the only solution for the above problem is $f_1 = f_2$. From that, we can calculate $d \approx 10.4\ (mm)$.

Therefore, after considering the probability of ambiguity, we choose the distance between two sensors $d = 17\ mm$.

4.1.6 Number of sensors

In order to implement the weighted approximation algorithm, we need at least 3 sensors to be able to determine the center of the line. Assuming that the line center coincides with the center of interpolation from the sensor, we also need to add 2 sensors at the left and right ends to be able to determine whether the vehicle is deviating to the left or right. Thus, with 5 sensors, we can completely determine the center of the line and the direction of deviation of the vehicle.

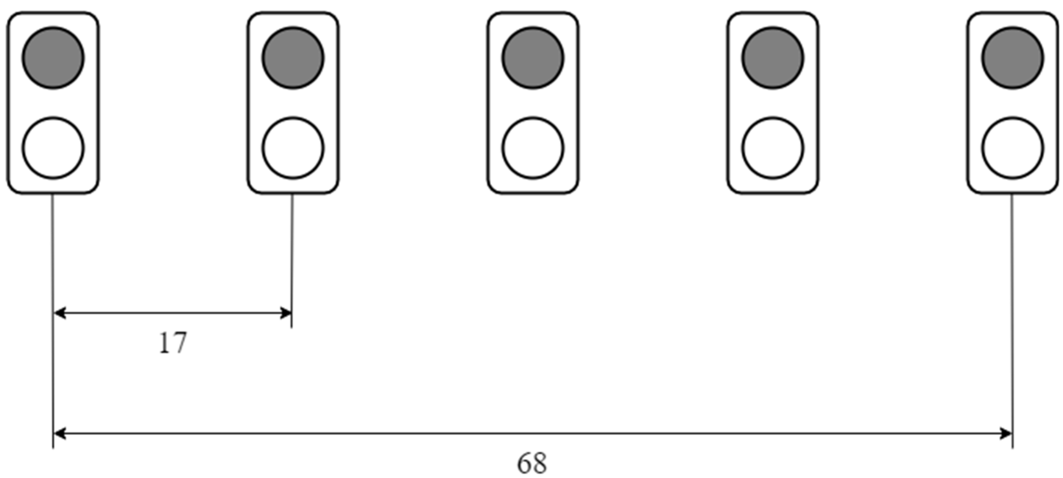


Fig 4.13 Illustration of calculating number of sensors

However, with 5 sensors, when the robot enters a corner, it may deviate from the line because of insufficient feedback signals. Therefore, we also include another pair of sensors: one at each end of the sensor circuit to ensure the robot operates seamlessly on its the entire journey.

In conclusion, a total of 7 sensors are used for the robot sensor circuit.

4.1.7 PCB design of TCRT5000 sensors board

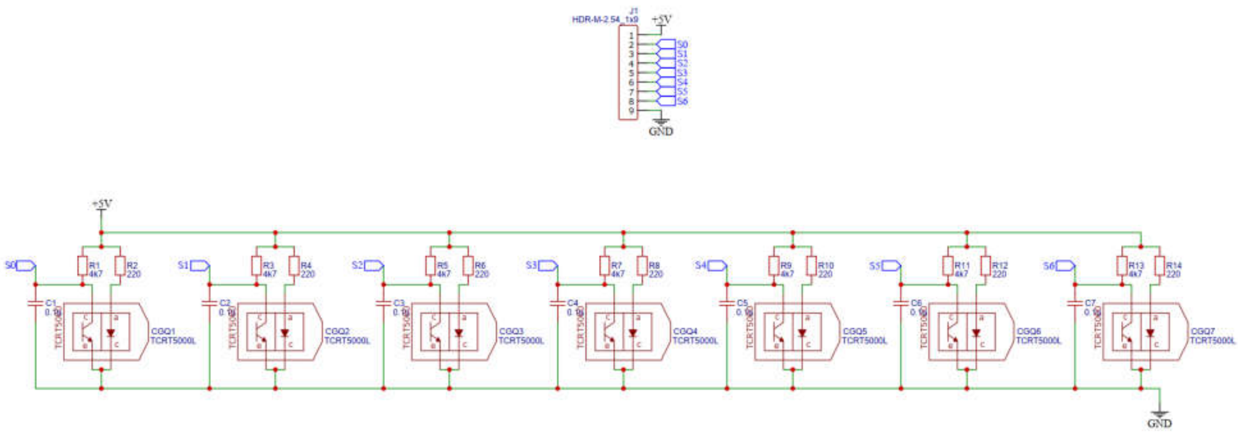


Fig 4.14 PCB design of TCRT5000 sensors board

4.1.8 Calibration of sensor value

We designed a 7 sensors circuit placed vertically with a distance between the sensors of 17 mm. Each line detector sensor will return a different analog signal under the same conditions. Therefore, calibrating the sensor is necessary. Select the calibration method by software with the following formula:

$$y_{calib\ i} = y_{min} + \frac{y_{max} - y_{min}}{x_{max,i} - x_{min,i}}(x_i - x_{min,i})$$

Where:

- $x_{max,i}$ và $x_{min,i}$: Maximum and minimum analog value of sensor i.
- y_{max} và y_{min}: Maximum and minimum expected value of sensor.
- x_i: Analog value of sensor i.
- $y_{calib\ i}$: Calibrated value of sensor i.

We assembled 7 sensors distances 17 mm among themselves, on our printed PCB and onto our mobile robot chassis.

After measuring at $h = 10\ (mm)$, 7 sensor value at white and black color are different among themselves:

Sensor	Value at white	Value at black
1	554	26
2	427	26
3	416	26
4	453	26
5	470	26
6	349	26
7	473	26
Average	448.8571	26

Table 4.2 Sensor value at black and white color

Select $y_{max} = 449$ và $y_{min} = 26$, we have the calibrated value:

Sensors	y_{calib}
1	$y_{calib\ 1} = 26 + \frac{423}{528}(x_1 - 26)$
2	$y_{calib\ 2} = 26 + \frac{423}{401}(x_2 - 26)$
3	$y_{calib\ 3} = 26 + \frac{423}{390}(x_3 - 26)$
4	$y_{calib\ 4} = 26 + \frac{423}{427}(x_4 - 26)$
5	$y_{calib\ 5} = 26 + \frac{423}{444}(x_5 - 26)$
6	$y_{calib\ 6} = 26 + \frac{423}{323}(x_6 - 26)$
7	$y_{calib\ 7} = 26 + \frac{423}{447}(x_7 - 26)$

Table 4.3 Sensor value after calibration

4.1.9 TCRT5000 weighted approximation algorithm

We use 7 IR sensors, let the value of the weight of them are x_1, x_2, x_3, x_4, x_5, x_6, x_7 and the analog value are y_1, y_2, y_3, y_4, y_5, y_6, y_7. Then the position of the center of line will be calculated by formula:

$$x = L_{CB}\frac{\sum_{i=1}^{7} x_i y_i}{\sum_{i=1}^{7} y_i}$$

$$= 17 \times \frac{3(y_{calib\ 1} - y_{calib\ 7}) + 2(y_{calib\ 2} - y_{calib\ 6}) + (y_{calib\ 3} - y_{calib\ 5})}{y_{calib\ 1} + y_{calib\ 2} + y_{calib\ 3} + y_{calib\ 4} + y_{calib\ 5} + y_{calib\ 6} + y_{calib\ 7}}\,(mm)$$

We experiment at height h = 10 (mm), and let the edge of line from when the robot is oriented right side of line -30 (mm) to when the robot is oriented left side of line 30 (mm). We obtain the table:

Position	**S0**	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**
-30	35	30	33	38	181	295	41
-25	34	29	34	37	218	265	40
-20	34	33	35	38	295	112	40
-15	34	33	35	38	382	38	41
-10	33	33	35	80	380	36	42
-5	34	33	33	272	130	35	39
0	36	39	49	397	42	38	41
5	39	43	338	177	36	34	37
10	39	43	357	46	40	37	40
15	40	45	416	44	39	38	39
20	41	259	279	40	38	37	40
25	40	359	42	39	39	38	40
30	42	387	38	40	40	39	39

Table 4.4 Feedback value of sensors in experiment

From above table we apply the calibration and calculated the weighted average algorithm value. Then we fit a line to estimate the relationship between real distance and weighted average value.

Real distance from center of line (mm)	**Weighted approximate distance from center of line (mm)**	**Error (mm)**
-30	-20.04	9.96
-25	-19.22	5.78
-20	-13.54	6.46
-15	-10.71	4.29
-10	-10.02	0.02
-5	-3.52	1.48
0	-0.34	0.34
5	8.02	3.02
10	9.29	0.71
15	10.18	4.82
20	16	4
25	18.49	6.51
30	19.2	10.8

Table 4.5 Distance from center of line compared to weighted approximation algorithm

Based on the data we have, we use MATLab to fit a line into data

$$y = 0{,}7198x + 0{,}2915$$

Where:

- x is distance from center of line
- y is weighted approximation algorithm value

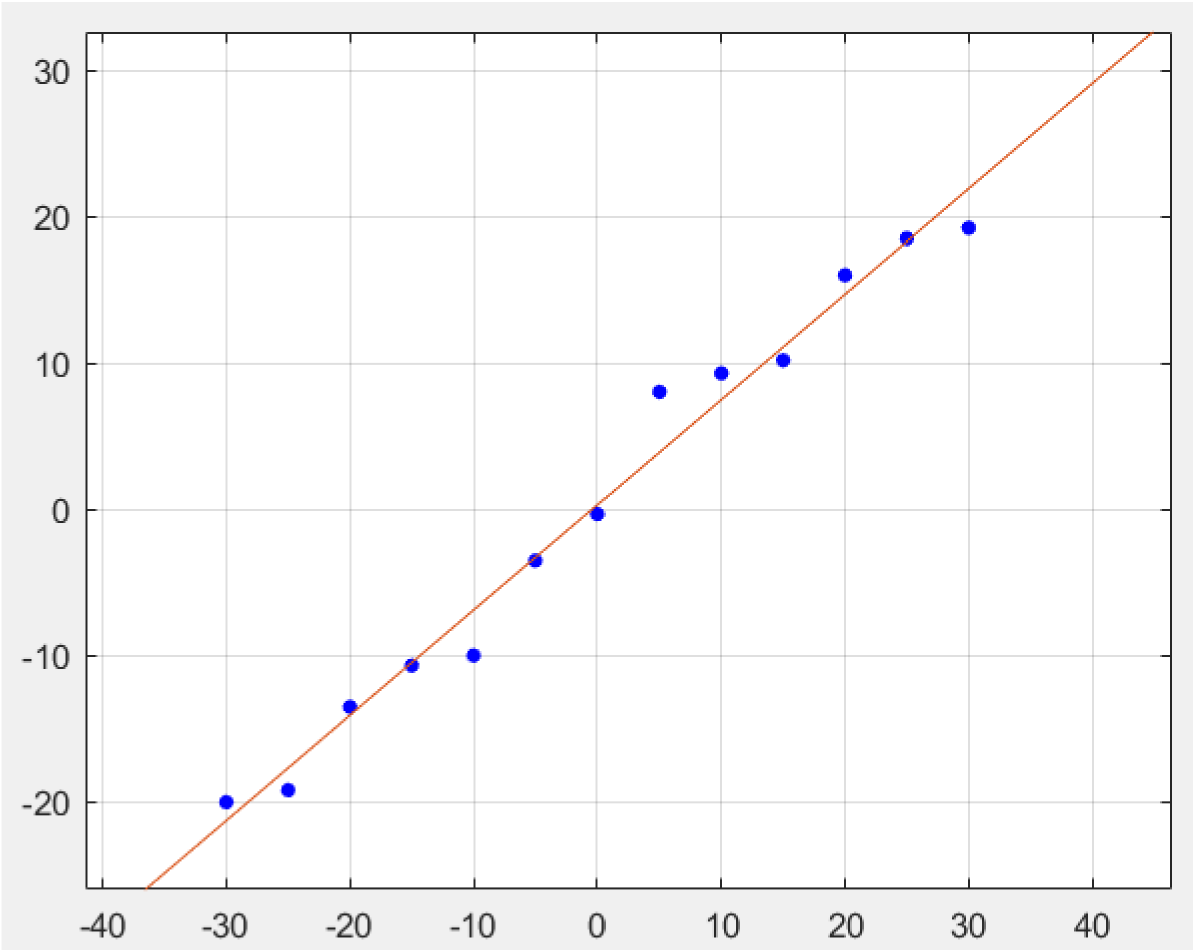


Fig 4.15 Linear fit of weighted approximation algorithm value

Then our error can be calculated as

$$e = x = \frac{y - 0{,}2915}{0{,}7198}$$

4.2 PCB design

4.2.1 Power supply

The dynamic circuit consists of the two actuators, coresponding to the two DC motors with encoder and their respective drivers. The power consumption of all necessary devices are listed in the following tables.

Device	Quantity	Current (*)	Voltage	Power
Driver TB6612	2	$12\ mA$	$5\ V$	0.12
DC Motor with encoder	2	$300\ mA$	12 V	12

Table 4.6 Dynamic circuit power

Device	Quantity	Current	Voltage	Power
TCRT5000 sensor module	1	$120\ mA$	$5\ V$	$0.6\ W$
Atmega328	2	$0.2\ mA$	$5\ V$	$0.002\ W$
Atmega32	1	$0.2\ mA$	$5\ V$	$0.001\ W$
Encoder	2	$100\ mA$	$3.3\ V$	0.66
TCS34725	1	$0.65\ mA$	$3.3\ V$	$0.0021\ W$

Table 4.7 Control circuit power

Track length can be approximated using the following formula:

$$L = 2000 + 500 + 500 \times 3.14 + \frac{800 \times 3.14 \times 2}{4} + \left(1500 - 800 \times \frac{3.14}{8}\right)$$

$$L = 6512\ (mm) = 6.512\ (m)$$

With a speed of $v = 0.3\ (m/s)$, the robot will complete the track in:

$$t = \frac{6.512}{0.3} = 21.7067$$

Assuming that 100 test run is necessary before finalization of the robot, within one – hour time, the robot will have completed $3600/21.7067 \approx 166$ tracks. The energy required for such an operation is:

$$E = P_{total} \times h = 11.5656\ (Wh)$$

Furthermore, the power supply must be able to satisfy power requirements of all components of the robot, which must yield a minimum voltage of $12\ (V)$. In terms of current supply, the source has to give $1603.6\ (mA)$ for an hour, i.e has a capacity of $C \geq 1603.6\ (mAh)$

For these reasons, a pack of four Lishen 18650 batteries is chosen as the power supply for the kinematics board on the robot, giving a total voltage of $14.8\ (V)$ and with capacity of $C = 2000\ (mAh)$ and a discharge rate of $20C$, it can supply a current of $2(A)$ for a one – hour period and give a maximum discharge current of $40A$.

Similarly, a pack of two Lishen 18650 batteries can be used to supply a voltage of $7.4\ (V)$ to all three microcontrollers, the sensor module and the color sensor.

The power from the batteries are sufficient, however voltage regulators and/or buck – boost converters are required to regulate the output voltage of the battery pack to the required voltages of the specific components.

The LM2596 buck converter IC is used as it converts voltage with relatively high efficiency (above 73%) and is capable of driving a $3A$ load with excellent line and load regulation. [20] This device is available in adjustable output version and it is internally compensated to minimize the number of external components to simplify the power supply design.

4.2.2 Buck converter

The LM2596 offers fixed voltages of 3.3 (V), 5 (V), and 12 (V); there is also an adjustable version of the chip, however, this requires additional resistors to set the desire output voltage and would make the board design marginally harder.

An example circuit of the fixed voltage version of LM2596 is given in the datasheet:

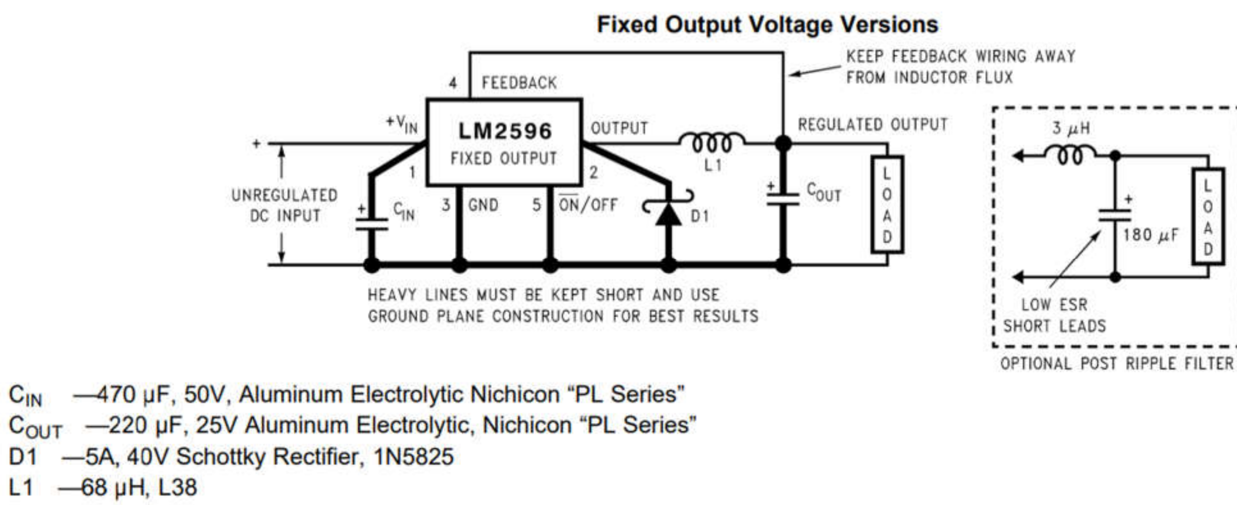
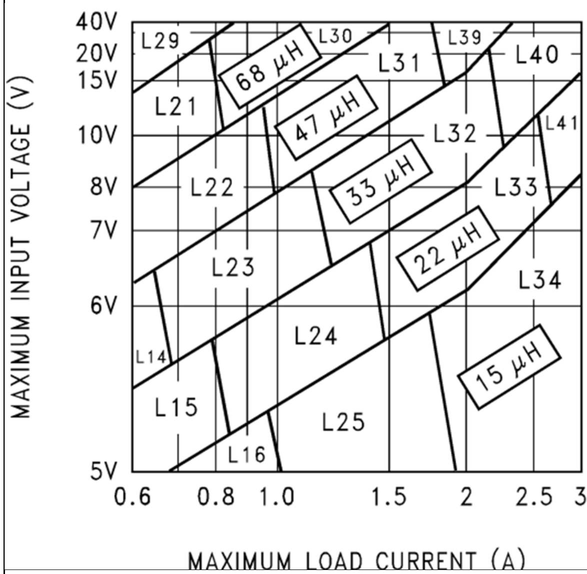


Fig 4.16 Example circuit for fixed – voltage LM2596

The datasheet also presented the buck regulator design procedure for both fixed output and adjustable output cases. Here, the fixed output procedure is carried out with input voltage from the battery pack.

PROCEDURE (Fixed Output Voltage Version LM2596 – 12)
Given: $$V_{out} = 12\ (V)$$ $$V_{in} = 14.8\ (V)$$ $$I_{Load(max)} = 750\ (mA)$$
1. Inductor Selection (L_1)



From the inductor value selection guide and table 3 of **[1]**, identify the inductance to be $47\ (mH)$

Fig 4.17 Inductor value selection guide LM2596 – 12

2. Output Capacitor Selection (C_{out}) From the quick design component selection table 1 in **[1]**, select a capacitor of $330\ (\mu F)$. The advised voltage rating of the capacitor is $25\ (V)$, however, the capacitor voltage rating for electrolytic capacitors should be at least 1.5 times greater than the output voltage, and often much higher voltage ratings are needed to satisfy the low ESR requirements for low output ripple voltage. Therefore, select an electrolytic capacitor with $\mu F/V = 330\ /50$ Also, for the selected output capacitor to work well with the inductor, let us switch the the chosen inductor in the previous step with one that has an inductance of $33\mu H$
3. Catch Diode Selection (D_1) The catch diode current rating must be at least 1.3 times greater than the maximum load current. The reverse voltage rating of the diode should be at least 1.25 times the maximum input voltage. This diode must be fast (short reverse recovery time) and must

be located close to the LM2596 using short leads and short printed circuit traces. Because of their fast-switching speed and low forward voltage drop, Schottky diodes provide the best performance and efficiency, and should be the first choice, especially in low output voltage applications. From table 6 of **[1]**, Schottky 1N5820 would suffice, however, due to commercial availability, a 1N5822 is selected.
4. Input Capacitor (C_{in}) The important parameters for the input capacitor are the input voltage rating and the RMS current rating (not discussed here). With a nominal input voltage of $14.8\ (V)$, an aluminum electrolytic capacitor with a voltage rating greater than $14.8 \times 1.5 = 22.2\ (V)$ would be needed. The next higher capacitor voltage rating is $25\ (V)$. Select a capacitor with $\mu F/V = 680\ /25$

Table 4.8 Information on PROCEDURE for

Fixed Output Voltage Version LM2596 – 12)

A similar procedure can be carried out for the other two LM2596s, the results is presented in the following table.

Fixed Output Voltage Version **LM2596 – 5**	**Fixed Output Voltage Version** **LM2596 – 3.3**
Given: $V_{out} = 5\ (V)$	**Given:** $V_{out} = 3.3\ (V)$

$V_{in} = 14.8\ (V)$ $I_{Load(max)} = 900\ (mA)$	$V_{in} = 14.8\ (V)$ $I_{Load(max)} = 600\ (mA)^{(*)}$
1. Inductor Selection L_2 Select $L_2 = 68\mu H$	**1. Inductor Selection L_3** Select $L_3 = 68\mu H$
2. Output Capacitor Selection (C_{out}) Select capacitor with $\mu F/V = 330\ /50$, switch to $L_2 = 33\ (\mu H)$	**2. Output Capacitor Selection (C_{out})** Select capacitor with $\mu F/V = 330\ /50$, switch to $L_2 = 33\ (\mu H)$
3. Catch Diode Selection (D_1) Select Schottky 1N5822	**3. Catch Diode Selection (D_1)** Select Schottky 1N5822
4. Input Capacitor (C_{in}) Select a capacitor with $\mu F/V = 680\ /25$	**4. Input Capacitor (C_{in})** Select a capacitor with $\mu F/V = 680\ /25$

Table 4.9 Electrical calculation for Fixed Output Voltage Version LM2596s)

In addition to the components listed above, supplementary components such as reverse current protection diode, additional filtering stages are also added to the block.

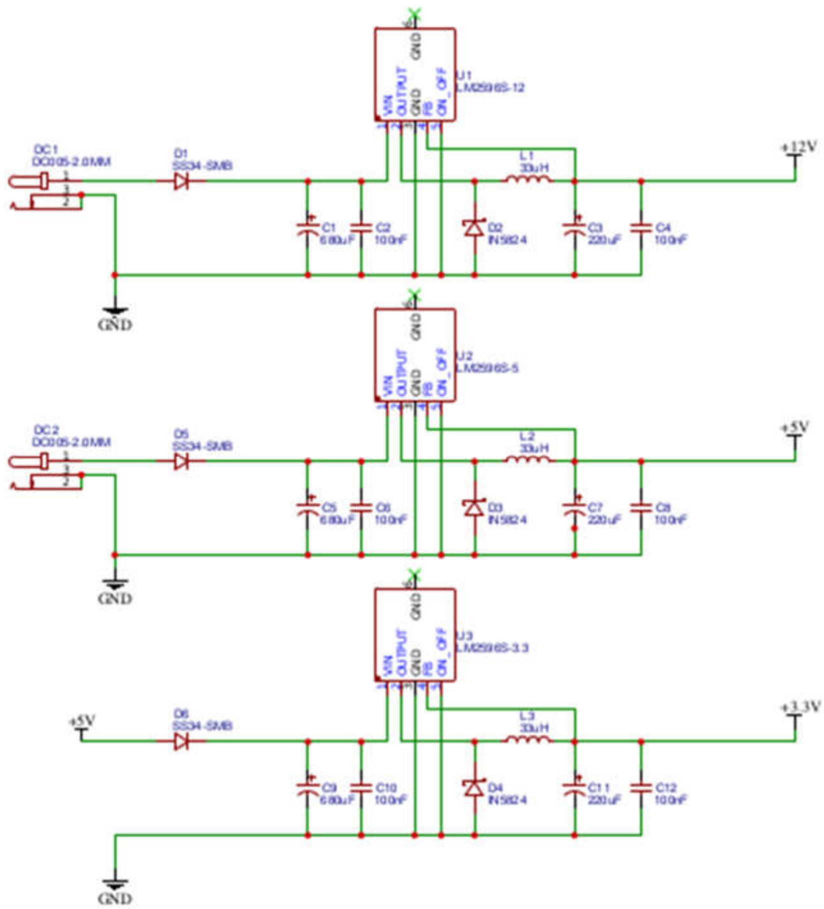


Fig 4.18 Buck converter block

4.2.3 Microcontroller block

As proposed, one Atmega32 and two Atmega328 are chosen as master and slaves respectively. The three microcontrollers communicate via I2C protocol. This protocol is chosen for many reasons:

- It requires only 2 wires for data transmission.
- At $(100\ Kbaud)$, the maximum bus length is about $1\ (m)$.
- Allows up to 128 devices on the bus.
- The sensor TCS34725 being used also communicates via I2C.

- It is an open-drain/open-collector communication standard which implies integrated circuits (IC's) with different voltage supply rails can be connected for communication.

Pullup resistors need to be connected from the I2C lines to the supply to enable communication. The pullup resistors pull the line high when it is not driven low by the open-drain interface. The value of the pullup resistor is an important design consideration for I2C systems as an incorrect value can lead to signal loss.

According to the reference [25], the minimum and maximum pull – up resistance is:

$$R_{p(min)} = \frac{V_{CC} - V_{OL}(max)}{I_{OL}}$$

$$R_{p(max)} = \frac{t_r}{0.8473 \times C_b}$$

Where:

- t_r: rise time of both SDA and SCL signals
- C_b: capacitive load for each bus line
- V_{OL}: low – level output voltage

There parameters are given in **[2]** in the following table

Parameter	Standard Mode (Max)	Fast Mode (Max)	Fast Mode Plus (Max)	Unit

t_r		1000	300	120	(ns)
C_b		400	400	550	(pF)
V_{OL}	$3\ (mA)$ current sink, $V_{CC} > 2\ (V)$	0.4	0.4	0.4	(V)
	$3\ (mA)$ current sink, $V_{CC} \leq 2\ (V)$	-	$0.2 \times V_{CC}$	$0.2 \times V_{CC}$	(V)

Table 4.10 Parametric from I2C specifications

For standard I2C communication with $V_{CC} = 3.3\ (V)$, the pull – up resistors values are:

$$R_{p(min)} = \frac{V_{CC} - V_{OL}(max)}{I_{OL}} = \frac{3.3 - 0.4}{3 \times 10^{-3}} = 966.667\ (\Omega)$$

$$R_{p(max)} = \frac{t_r}{0.8473 \times C_b} = \frac{1000 \times 10^{-9}}{0.8473 \times 400 \times 10^{-12}} = 2950.5\ (\Omega)$$

Two 1 $(k\Omega)$ pull – up resistors are chosen and placed on SDA and SCL lines.

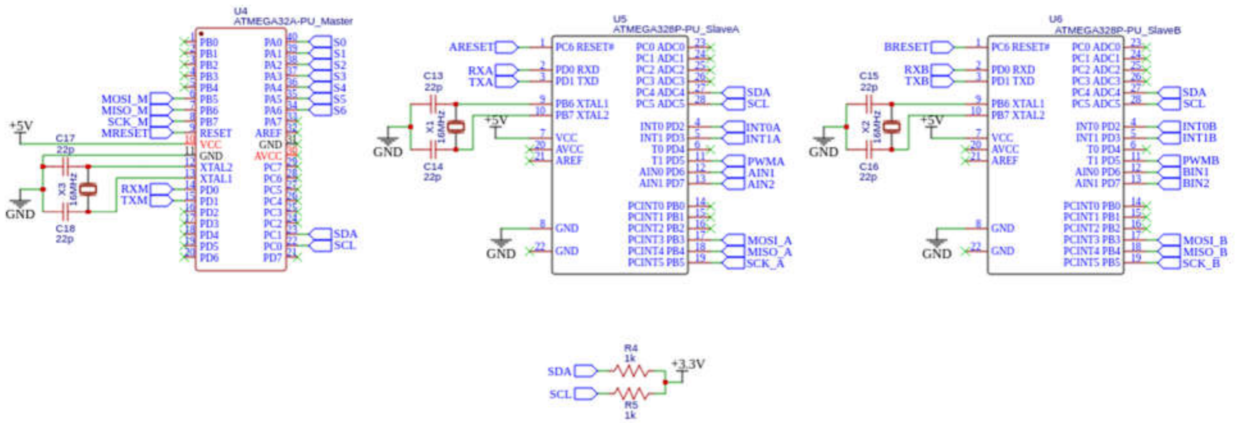


Fig 4.19 Microcontroller block

4.2.4 Motor and driver block

The driver chosen is the TB6612 due to its portable size, ability to linearize the motor output speed with respect to input PWM. Each Slave drives their coresponding DC motor.

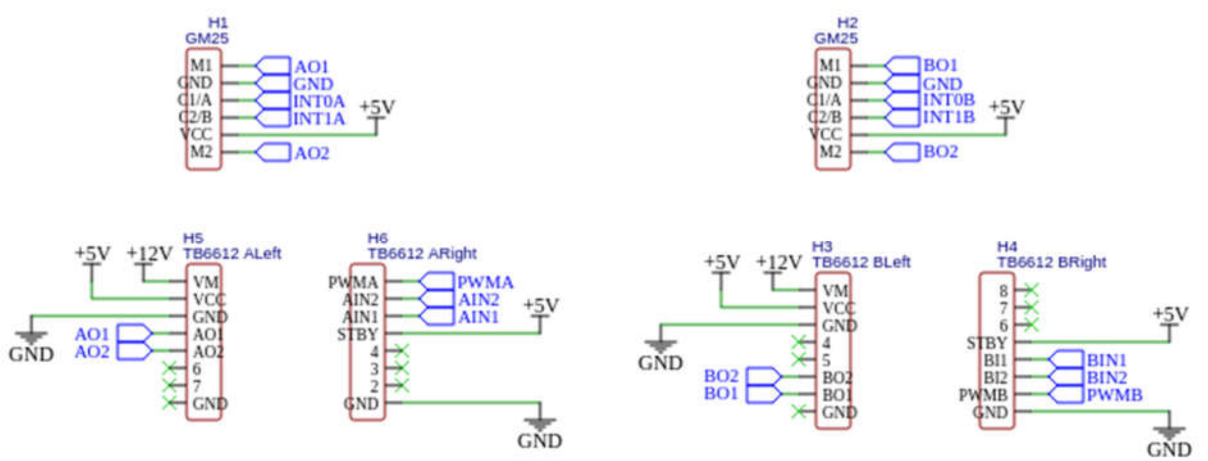


Fig 4.20 Driver and Motor block

4.2.5 Color sensor block

The TCS34725 is capable of RGB detection. It is useful since the primary objective of this sensor is differentiating between two available color of the loaded cargo. Two LEDs are also embedded for better accuracy, and they are active by default.

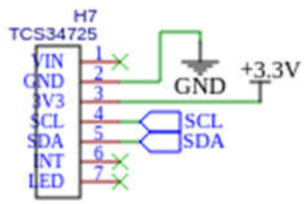


Fig 4.21 Color sensor

CHAPTER V: MATHEMATICAL MODELLING OF SYSTEM AND CONTROLLER DESIGN

Designing process:

- Determine motor controller
- Mathematical modelling mobile robot
- Determine robot controller

5.1 Motor control

5.1.1 Check linearity

We examine the linear relation between the input ($\%PWM$) and the output revolution of the motor's shaft (rpm). To do that, we PWM signal is fed into the motor, duty cycle from $5 - 100\%$. After that, we read the rpm signal from the encoder.

%PMW	rpm
5%	0
10%	0
15%	28.133
20%	50.7385
25%	69.346
30%	85.8935
35%	104.143
40%	118.375
45%	129.678
50%	141.245
55%	151.046

60%	161.361
65%	168.31
70%	176.172
75%	181.71
80%	187.617
85%	192.982
90%	198.063
95%	202.461
100%	219.046

Table 5.1 Table of rpm data of motor

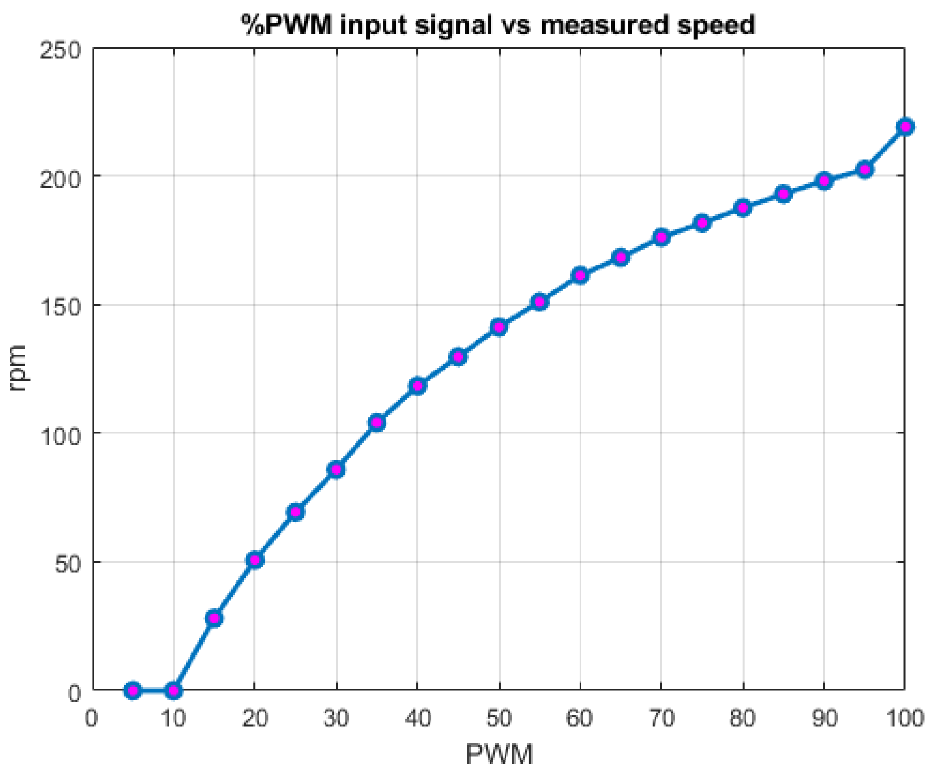


Fig 5.1 %PWM vs measured velocity (rpm) graph

Test with TB6612 motor driver

Follow the same procedure as above:

%PMW	rpm
5%	0
10%	21.4281
15%	33.0784
20%	45.6978
25%	59.2714
30%	71.612
35%	85.1655
40%	98.9197
45%	111.3331
50%	123.494
55%	136.1053
60%	149.1838
65%	162.2477
70%	176.4101
75%	187.6647
80%	198.401
85%	214.4792
90%	229.4622
95%	241.7738

100%	254.8411

Table 5.2 Table of rpm data of motor

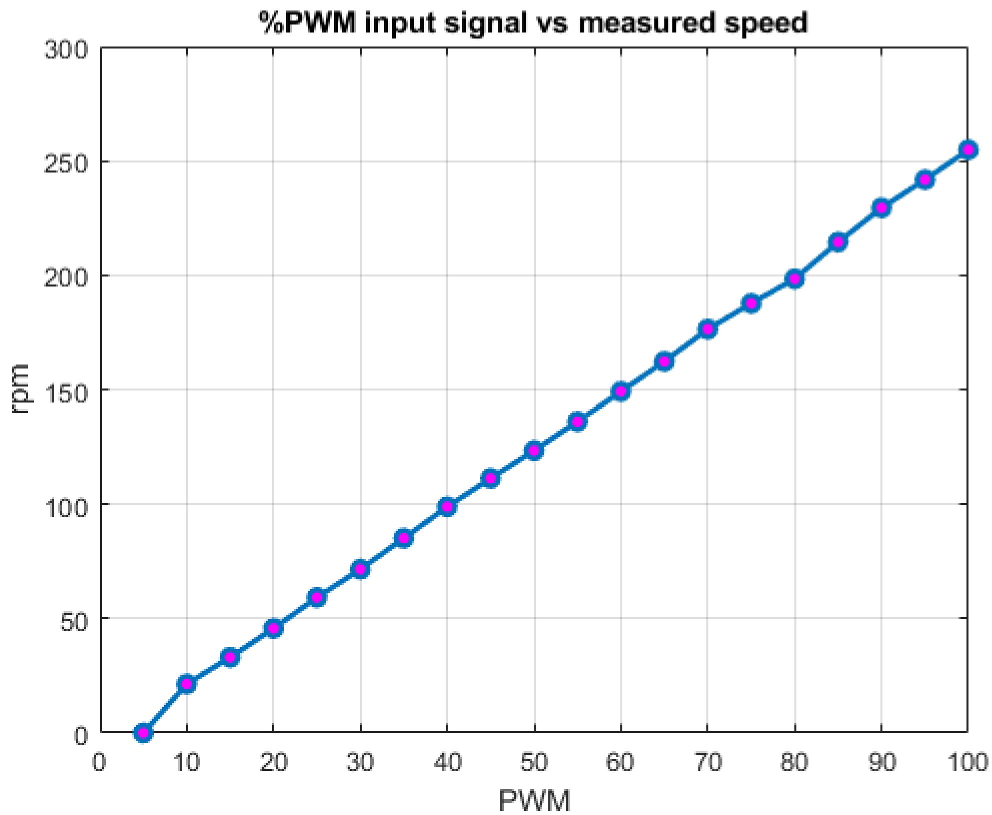


Fig 5.2 %PWM vs measured velocity (rpm) graph

We see that the TB6612 helps making the relationship between PWM and speed linear better than L298N.

Conclusion: Choose TB6612 as motor drivers.

5.1.2 Determine motor transfer function

Prepare data for System identification toolbox:

Let a PWM input changes in the function:

$$f(t) = 127.5\sin(t) + 127.5$$

$$\rightarrow T = 6.28(s)$$

Then:

$$N \times \delta t = m \times T$$

δt is from the Nyquist criterion and our microcontroller speed.

$$f_{Sampling} \geq 2f_{signal} \rightarrow \delta t \leq \frac{1}{2}T$$

For Atmega328PU, our minimum sampling time $\delta t = 0.022\ s$. Our T satisfy the condition.

$$\rightarrow N \times 0.022 = m \times 6.2832$$

We choose $N = 1428;\ m = 5$ (N, m must be integer)

For the motor, the output signal is the revolution of the output shaft (rpm), input is PWM in a sinusoidal wave. Therefore, in order to obtain the motor transfer function, we will provide the motor with $\%PWM$, then measure the revolution of the output shaft.

Motor 1

- Plotting the input and output data. We obtain the following graph:

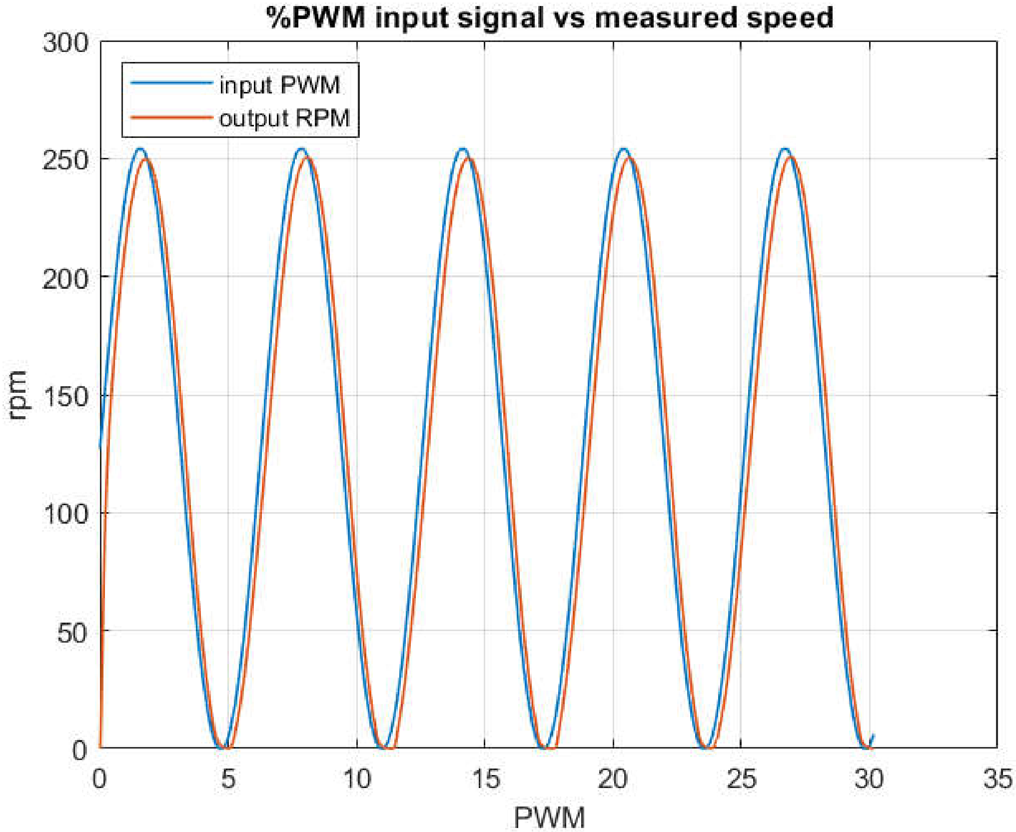


Fig 5.3 *Measured speed of the motor*

- Using MATLAB System Identification Toolbox (SIT), we yield the following transfer function of the motor-driver plant 1

$$G(s) = \frac{1.008}{0.18162 \times s + 1} = \frac{5.55}{s + 5.51}$$

	Current value
Settling time	$T_s = 0.18162 \times 4 = 0.72648\ (s)$
Overshoot	1%
Steady-state error for step input	0.01

Table 5.3 Value to calculate transfer function of the motor-driver plant 1

Validation of transfer function

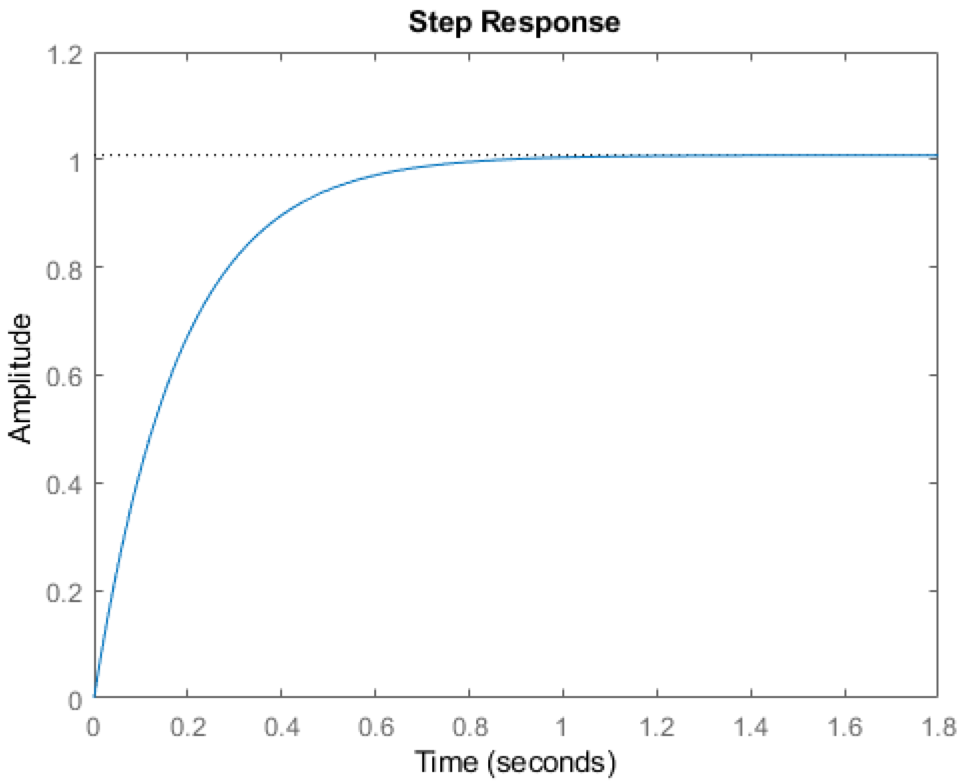


Fig 5.4 Step response graph of the transfer function

a) PID Controller Design

When we add a controller, the closed loop transfer function will change, so we have to test to make sure it is accurate in real life. Assume negative unity feedback.

Validation of transfer function

Check Steady state error for step input:

$$\lim_{s\to 0}\frac{1}{1+\frac{5.55}{s+5.51}} \approx 0.5$$

Test by real data plot:

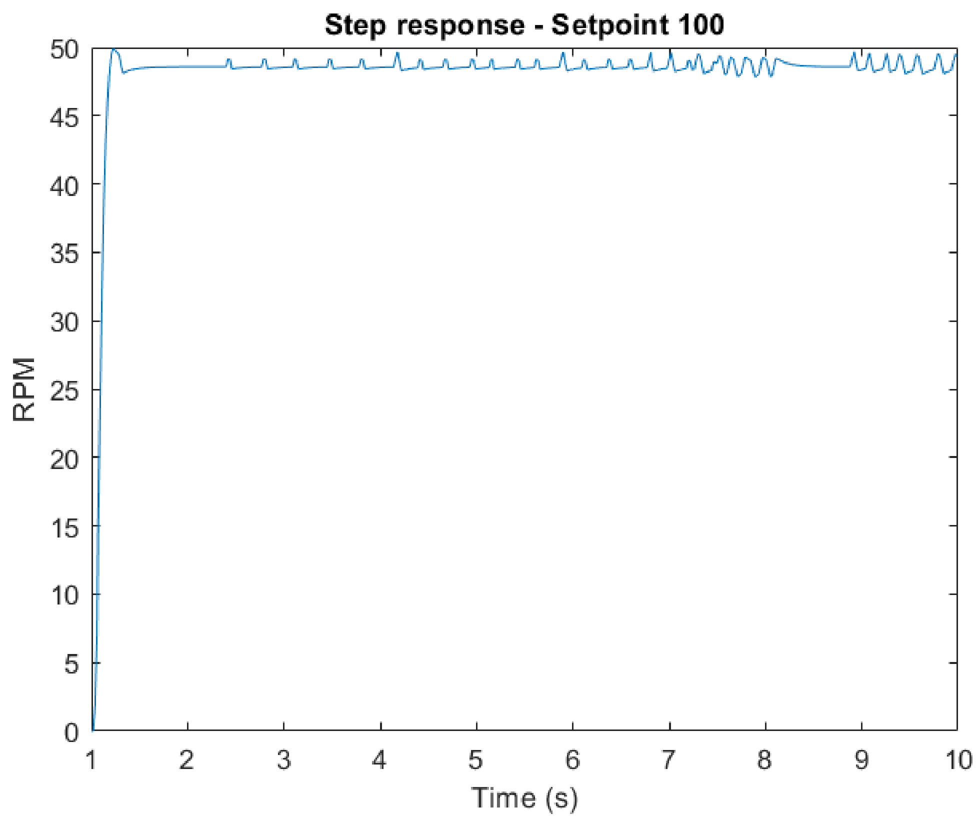


Fig 5.5 Step response graph of the transfer function at setpoint 100

We want to make the motor-driver settling time smaller than the sampling time of the microcontroller so that it can work fine. $T_s < \delta t$

Select $T_s = 0.015\ s \rightarrow \frac{0.72648}{0.015} = 48$ times faster than original

Keep the overshoot at this moment. We also want to remove steady-state error for step input

With the above requirements, we implement the PID controller for this motor-driver plant. Assuming negative unity feedback, the forward transfer function will be

$$K\frac{(s+a_1)(s+a_2)}{s}\times\frac{5.55}{s+5.51}$$

$$\zeta=-\frac{\ln(\%OS)}{\sqrt{\ln(\%OS)^2+\pi^2}}=-\frac{\ln(0.01)}{\sqrt{\ln(0.01)^2+\pi^2}}=0.8261$$

$$T_s=0.015=\frac{-\ln\left(0.02\sqrt{1-\zeta^2}\right)}{\zeta\omega_n}\rightarrow\omega_n=\frac{-\ln\left(0.02\sqrt{1-0.8261^2}\right)}{0.015\times0.8261}=361.991$$

First, we try PI controller to make the system get zero steady state error for step input:

b) PI

Desired characteristics equation:

$$1+\frac{K(s+a_2)}{s}\frac{5.55}{s+5.51}=0\rightarrow s^2+5.51s+5.55\times Ks+5.55\times Ka_2=0$$

Desired characteristics equation form: $s^2+2\zeta\omega_n s+\omega_n^2=0$

$$\rightarrow K=106.7696;a_2=221.1336$$

```
% Proportional Integral controller
Kpi = 106.7696;
a2 = 221.1336;
Ki = Kpi*a2
sys_PIcl = feedback(Kpi*(s+a2)*OL/s,1);
step(sys_PIcl)
grid
title('Step Response with PI Control')
```

```
% Proportional Derivative controller
Kpd = 0.3528;
sys_PDcl = feedback(Kpd*(s+132.24)*OL,1);
step(sys_PDcl)
grid
title('Step Response with PD Control')
```

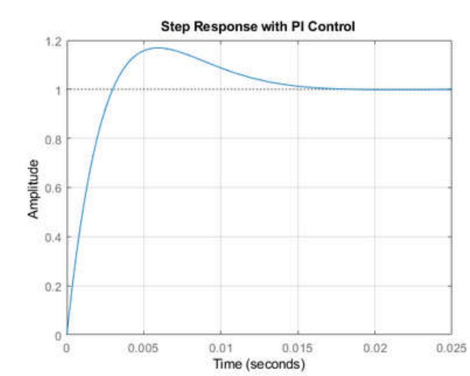


Fig 5.6 Response of PI controller

c) PD

Desired characteristics equation:

$$1 + K(s + a_1)\frac{5.55}{s + 5.51} = 0 \rightarrow s + 5.51 + 5.55 \times Ks + 5.55 \times Ka_1 = 0$$

$$\rightarrow s = -\frac{5.51 + 5.55Ka_1}{5.55K + 1} = 48 \times (-5.51) = -264.48 = s_0$$

Select $a_1 = 132.24$

$$K = \frac{1}{|T(s_0)|} = 0.3528$$

$$0.3528(s + 132.24) \times \frac{5.55}{s + 5.51}$$

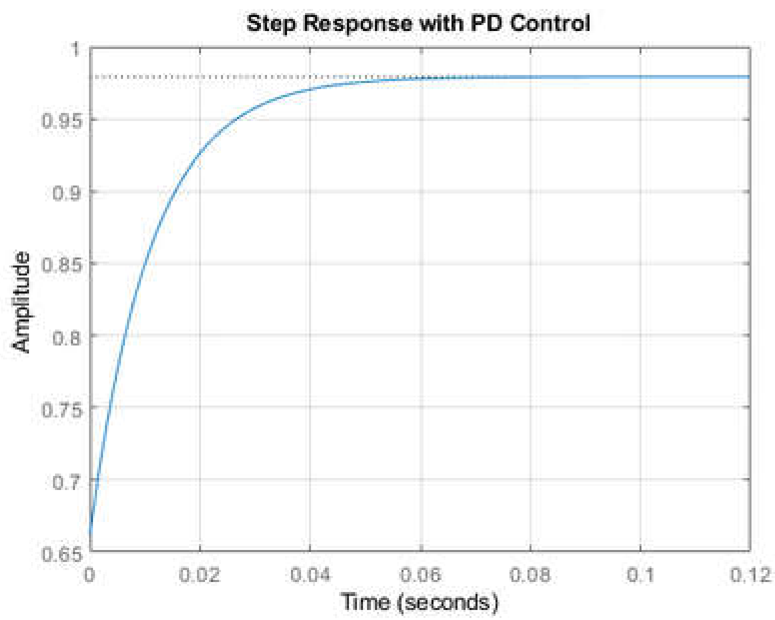


Fig 5.7 Response of PD controller

d) PID controller:

$$\frac{K(s + a_1)(s + a_2)}{s}$$

Select $a_2 = 0.001$

Desired characteristics equation:

$$1 + \frac{K(s + a_1)(s + a_2)}{s}\frac{5.55}{s + 5.51} = 0$$

$$\rightarrow s^2 + 5.51s + 5.55 \times [Ks^2 + K(a_1 + a_2)s + Ka_1a_2] = 0$$

$$\rightarrow (5.55K+1)s^2 + [5.51 + 5.55K(a_1+a_2)]s + 5.55Ka_1a_2 = 0$$

Desired characteristics equation form: $s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$

$$\rightarrow \begin{cases} \omega_n = \sqrt{\dfrac{5.55Ka_1a_2}{5.55K+1}} = 361.991 \rightarrow \sqrt{\dfrac{0.00555Ka_1}{5.55K+1}} = 361.991 \\ \zeta = \dfrac{5.51 + 5.55K(a_1+a_2)}{2\omega_n(5.55K+1)} = \dfrac{5.51 + 5.55Ka_1 + 0.00555K}{2\times 361.991 \times (5.55K+1)} = 0.8261 \end{cases}$$

$$\rightarrow \begin{cases} K = \dfrac{1}{5.55} \\ a_1 = 592.5738 \end{cases}$$

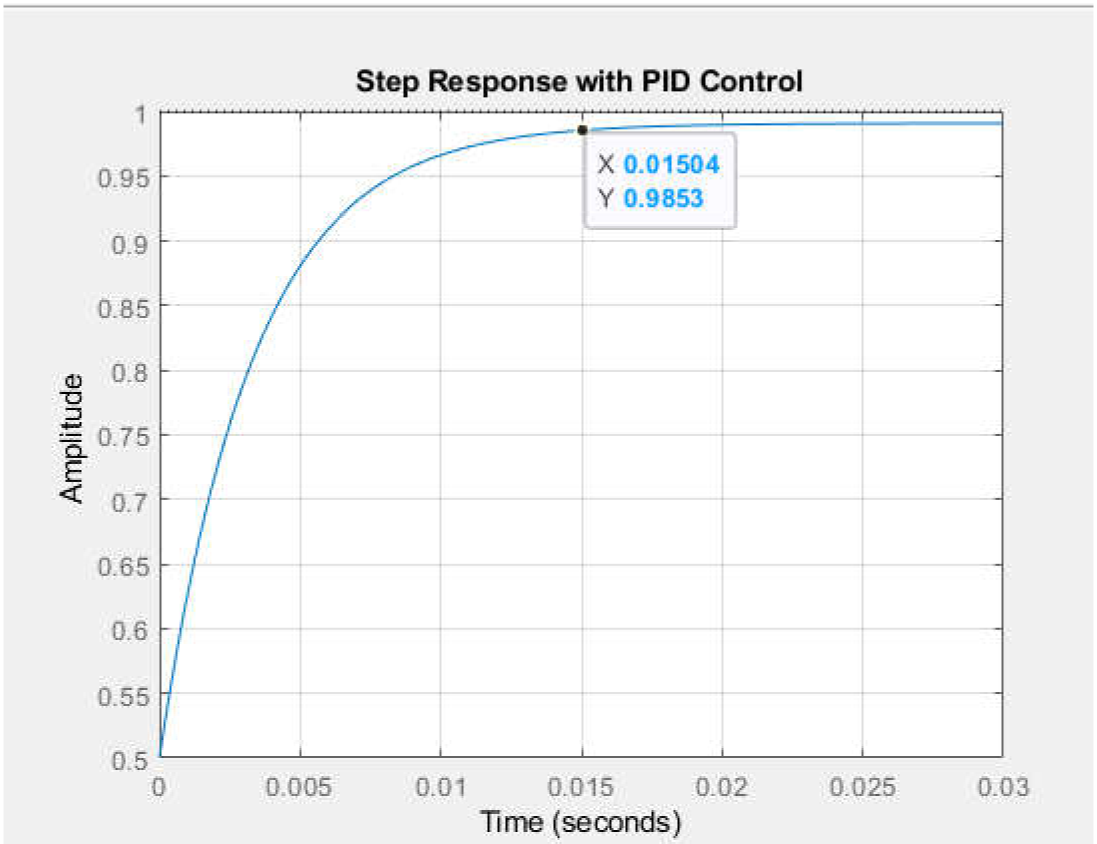


Fig 5.8 Response of PID controller

Result of 3 controllers PI,PD,PID parameters K_p, K_i, K_d in reality is not usable, the motor vibrates and shakes strongly.

We apply method tuning with Ziegler-Nichols method for unknown system [26]:

We found the maximum gain $K_p = 11$ where the response to step input is steady oscillation, then we have the period of oscillation P_{cr}:

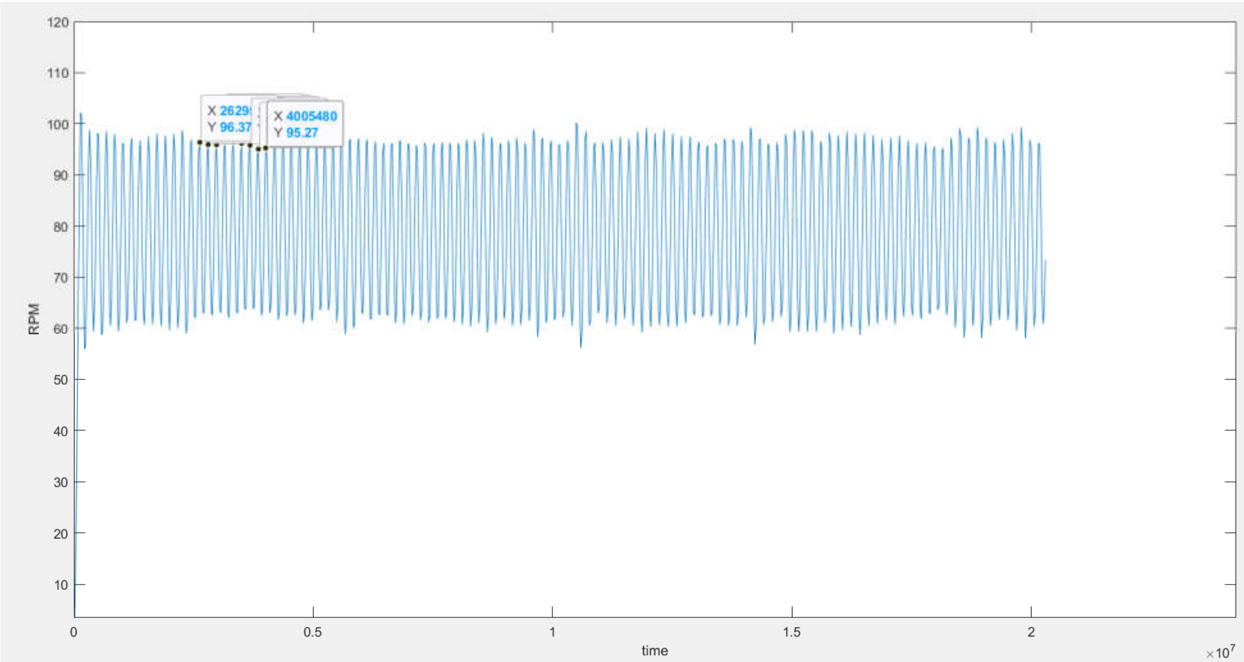


Fig 5.9 Response of motor 1 at setpoint 88 with P controller, $K_p = 11$

Based on our data $K_{cr} = 11, P_{cr} = 0.17472s$

Then we use Ziegler-Nichols formula for calculate the PID gain:

$$K_p = 0.6K_{cr} = 6.6$$

$$K_i = \frac{K_p}{0.5P_{cr}} = \frac{6.6}{0.5 \times 0.17472} = 75.55$$

$$K_d = K_p \times 0.125 \times P_{cr} = 6.6 \times 0.125 \times 0.17472 = 0.144$$

Motor 1 response after applying PID gain:

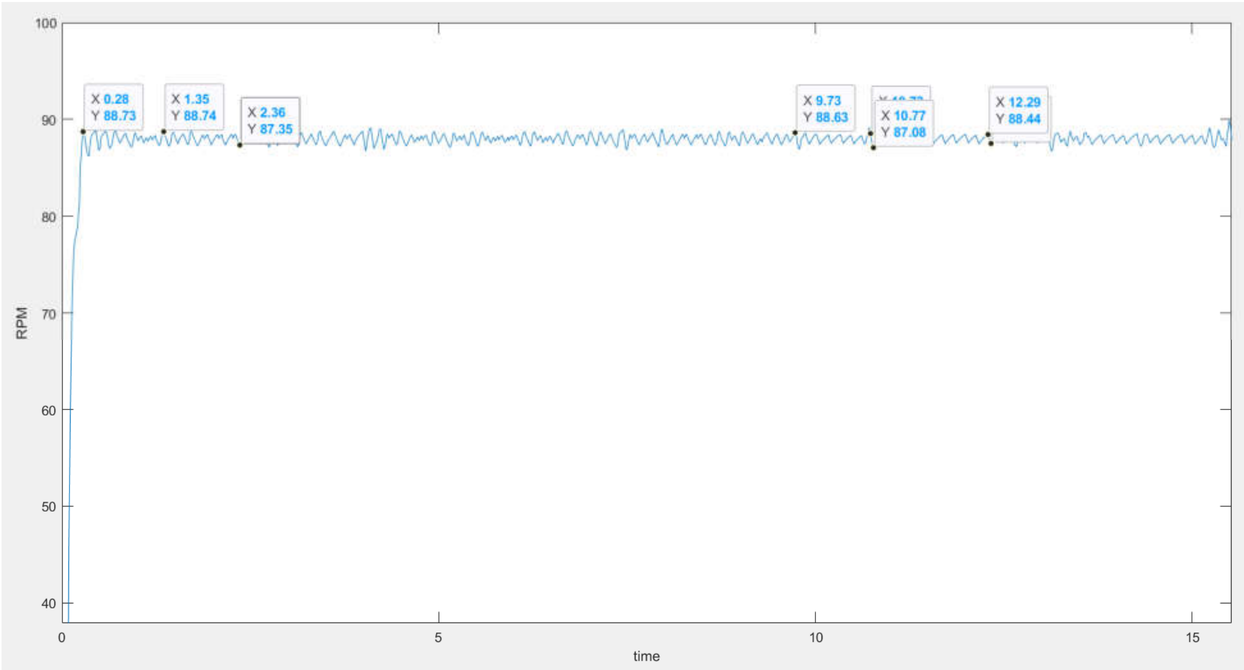


Fig 5.10 ZN PID controller on Motor 1 at setpoint 88 with adjusted K_p, K_i, K_d

System response with above $\mathrm{K_p, K_i, K_d}$

Settling time: $0.26s$

Overshoot $\%OS = 0.8\%$

Steady state error < 1

We will check our motor-driver plant 2 to see if they are identical in terms of system transfer function. Then if they are, calibration made on plant 2 will also be applied the same way plant 1. After that, the real run test will adjust the parameters for each plant.

Motor 2

- Plotting the input and output data. We obtain the following graph:

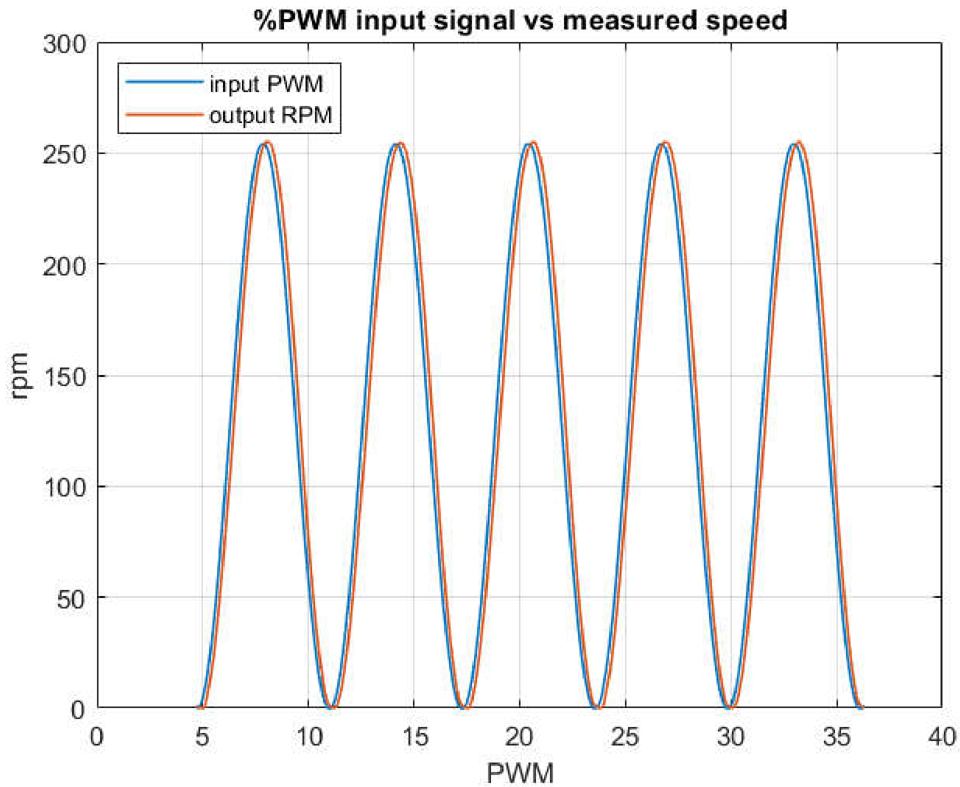


Fig 5.11 Measured speed of the motor

- Using MATLAB System Identification Toolbox (SIT), we yield the following transfer function of the motor-driver plant 2

$$G(s) = \frac{1.0074}{0.17213 \times s + 1} = \frac{5.852}{s + 5.809}$$

	Current value
Settling time	$T_s = 0.17213 \times 4 = 0.6885\ (s)$
Overshoot	1%
Steady-state error for step input	0.01

Table 5.4 Value to calculate transfer function of the motor-driver plant 2

Validation of transfer function

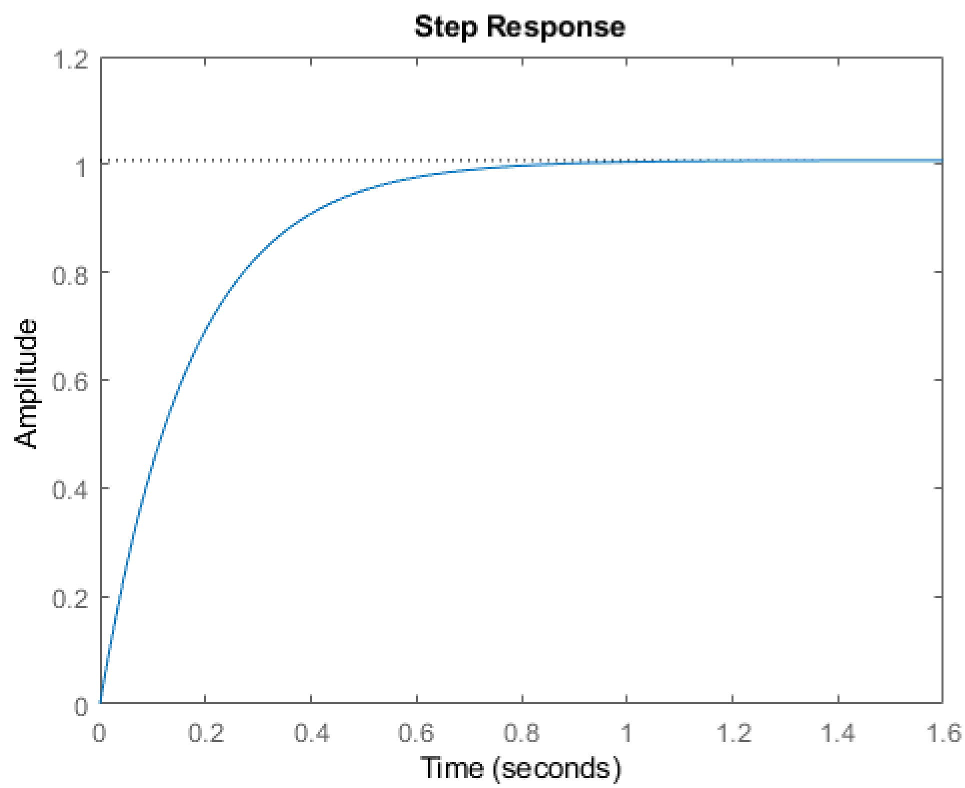


Fig 5.12 Step response graph of the transfer function

PID Controller Design

When we add a controller, the closed loop transfer function will change, so we have to test to make sure it is accurate in real life. Assume negative unity feedback.

Validation of transfer function

Check Steady state error for step input:

$$\lim_{s \to 0} \frac{1}{1 + \frac{5.852}{s + 5.809}} \approx 0.5$$

Test by real data plot:

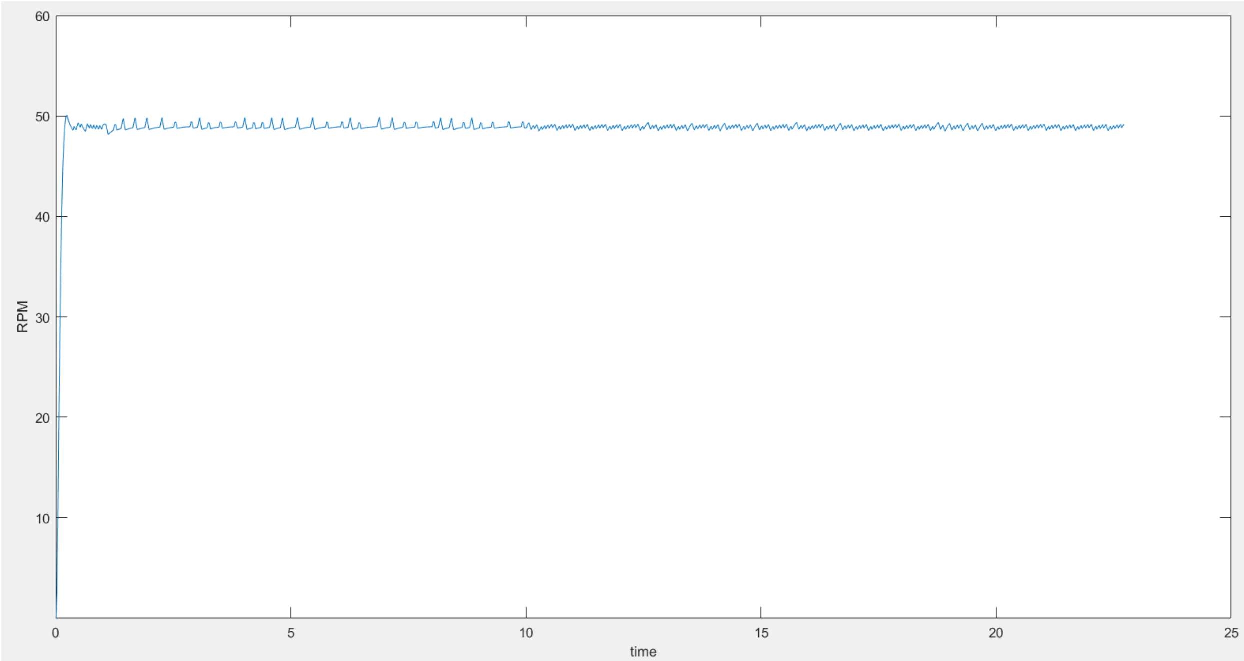


Fig 5.13 Measured speed of motor 2 at setpoint 100

→Plant 1 is nearly identical to plant 2

→We will apply the same controller as motor 1 then tune it later.

5.2 System control

5.2.1 Kinematic modelling of mobile robot

Based on the overview, we can see that our robot only performs on plain surface, on top of that, we can determine that the external forces (such as gravity, friction force, …) are invariant with time. In addition, our goal, as stated above, is to maintain the velocity when travelling, so we choose to analyze the kinematic of the robot for system modeling, which is solving the problem about its position and orientation.

For simplicity, we will focus on 2 driving wheels from now on.

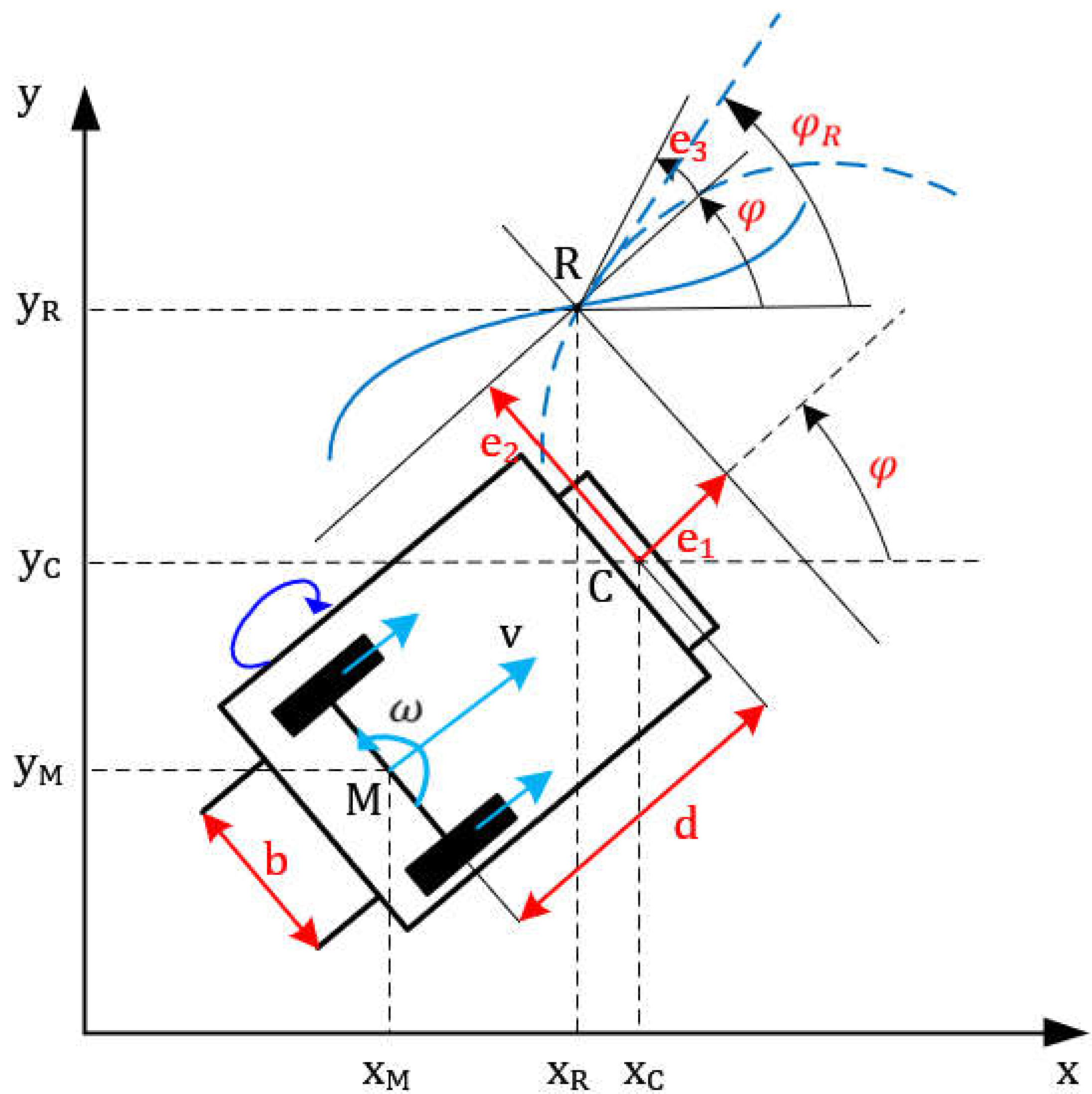


Fig 5.14 Coordinates of the mobile robot in the reference frame

The mobile robot model moves with velocity v and has angular velocity ω with the angular velocities of the left and right wheels being ω_L, ω_R respectively. Assuming the wheels roll without slipping, the translational velocities of the two wheels are expressed as:

$$V_L = \omega_L r \quad V_R = \omega_R \tag{5.0}$$

V_L: left wheel translational velocity (m/s)

V_R: right wheel translational velocity (m/s)

ω_L: left wheel angular velocity (rad/s)

ω_R: right wheel angular velocity (rad/s)

r: wheel radius (m)

According to the figure, we have a relationship between the tangential speed of the wheel and the velocity v of the car:

$$V_L = v - \frac{b}{2}\omega$$

$$V_R = v + \frac{b}{2}\omega$$

$$\Rightarrow \begin{cases} \omega = \frac{(V_R - V_L)}{b} = \frac{r}{b}(\omega_R - \omega_L) \\ v = \frac{1}{2}(V_R + V_L) = \frac{r}{2}(\omega_R + \omega_L) \end{cases} \quad (5.1)$$

We also have

$$\begin{cases} V_x = v\cos\varphi \\ V_y = v\sin\varphi \end{cases}$$

Then:

$$\begin{bmatrix} V_x \\ V_y \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\cos\theta & \frac{r}{2}\cos\theta \\ \frac{r}{2}\sin\theta & \frac{r}{2}\sin\theta \\ \frac{r}{b} & -\frac{r}{b} \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} \quad (5.2)$$

The kinematic equations at M

$$\begin{bmatrix} \dot{x_M} \\ \dot{y_M} \\ \dot{\varphi_M} \end{bmatrix} = \begin{bmatrix} cos\varphi & 0 \\ sin\varphi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (5.3)$$

While: v, ω are velocity and angular velocity.

The kinematic equations at C:

$$\begin{cases} x_C = x_M + d\,cos\,\varphi \\ y_C = y_M + d\,sin\,\varphi \\ \varphi_C = \varphi \end{cases} \quad (5.4)$$

Set d = MC

Differentiate both side:

$$\begin{cases} \dot{x_M} = \dot{x_M} - d\,\dot{\varphi} sin\,\varphi \\ \dot{y_M} = \dot{y_M} + d\dot{\varphi}\,cos\,\varphi \\ \dot{\varphi_M} = \dot{\varphi} \end{cases} \quad (5.5)$$

The kinematic equations at R:

$$\begin{cases} \dot{x_R} = v_R \cos\varphi \\ \dot{y_R} = v_R \sin\varphi \\ \dot{\varphi_R} = \omega_R \end{cases} \tag{5.6}$$

5.2.2 Error modelling of mobile robot

The controller is designed for midpoint of motor axis M to track the line, but we will first model the point C to investigate the error relative to reference point R with the desired velocity, because it is the midpoint of our sensor board. To do that, first we have the error of the robot relative to the reference point determined as follows:

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_R - x_C \\ y_R - y_C \\ \varphi_R - \varphi_C \end{bmatrix} \tag{5.7}$$

Differentiate both sides, error model of the robot is obtained:

$$\begin{bmatrix} \dot{e_1} \\ \dot{e_2} \\ \dot{e_3} \end{bmatrix} = \begin{bmatrix} v_R \cos e_3 \\ v_R \sin e_3 \\ \omega_R \end{bmatrix} + \begin{bmatrix} -1 & e_2 \\ 0 & -d \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{5.8}$$

An equilibrium point of the mobile robot system is the reference point, where $[e_1\ e_2\ e_3]^T = [0\ 0\ 0]^T$

Linearization equation around equilibrium point $[e_1\ e_2\ e_3]^T = [0\ 0\ 0]^T$ (5.8), we have:

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} 0 & \omega & 0 \\ 0 & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ 0 & -d \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{5.9}$$

In reality, calculating e_1 is unnecessary because we suppose robot moves with $\mathrm{v} = \mathrm{v_r}$. So $e_1 \approx 0$. Shorten (5.6) and let $\mathrm{u} = \omega$ we have:

$$\begin{bmatrix} \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} 0 & v_r \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e_2 \\ e_3 \end{bmatrix} + \begin{bmatrix} -d \\ -1 \end{bmatrix} u \tag{5.10}$$

Differentiate both sides of first equation of (5.7), we have:

$$\ddot{e_2} = -du - v_r u$$

Let $x_1 = e_2$

Let $x_2 = \dot{x_1} - \beta_1 u$, with $\beta_1 = -d$

We have:

$$\dot{x_2} = \ddot{x_1} - \beta_1 \dot{u} = -d\dot{u} - v_r u + d\dot{u} = -v_r u$$

In conclusion, we have state-space equation:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}-d\\ -v_r\end{bmatrix}u \tag{5.11}$$

Equation (5.8) is state-space equation $\dot{x} = Ax + Bu$

First, we check the controllability

$$M = [B \quad AB] = \begin{bmatrix}-d & -v_r\\ -v_r & 0\end{bmatrix} \rightarrow \text{rank}\,(M) = 2 \neq 0$$

So the system is controllable

Define $\mathrm{K} = [\mathrm{k}_1\ \mathrm{k}_2]$

Desired characteristic equation of the system:

$$|\mathrm{sI} - \mathrm{A} + \mathrm{BK}| = 0$$

$$s^2 - (k_1 d + k_2 v_r)s - k_1 v_r = 0$$

Desired characteristic equation:

$$s^2 + 2\zeta w_n s + {w_n}^2 = 0$$

Equating both sides

$$\begin{cases}-(k_1 d + k_2 v_r) = 2\zeta w_n\\ \quad -k_1 v_r = w_n^2\end{cases} \tag{5.12}$$

From $\mathrm{u} = -\mathrm{Kx} = -k_1 x_1 - k_2 x_2$ and $x_2 = \dot{x_1} + du$, we have:

$$u = -\frac{k_1}{1+k_2 d}x_1 - \frac{k_2}{1+k_2 d}\dot{x}_1 = -\frac{k_1}{1+k_2 d}e - \frac{k_2}{1+k_2 d}\dot{e} \tag{5.13}$$

Let $K_{Pr} = -\frac{k_1}{1+k_2 d}$ and $K_{Dr} = -\frac{k_2}{1+k_2 d}$

$$\mathrm{u} = \mathrm{K_{Pr}}\mathrm{e} + \mathrm{K_{Dr}}\dot{\mathrm{e}} \tag{5.14}$$

Equation (5.10) show that u is PD controller. Let

$$K_{Pr} > 0, K_{Dr} > 0$$

$$(5.12) \rightarrow \begin{cases} 1 + k_2 d > 0 \\ k_2 < 0 \end{cases} \rightarrow -5 < k_2 < 0 \ (d = 0.2m) \tag{5.15}$$

With $v_r = 0.3\ m/s$ we have:

$$(5.9) \rightarrow k_1 = -\frac{\omega_n^2}{v_r}$$

$$(5.12) \rightarrow \mathrm{k}_2 = \frac{d\omega_n^2 - 2\zeta v_r \omega_n}{v_r^2} = \frac{0.2\omega_n^2 - 0.6\zeta\omega_n}{0.09} \tag{5.15}$$

We want our error to have zero overshoot. Then:

$$\zeta = 1$$

$$(5.11), (5.12) \rightarrow -5 < \frac{0.2\omega_n^2 - 0.6\omega_n}{0.09} < 0$$

$$\rightarrow -5 < \frac{20}{9}\omega_n^2 - \frac{20}{3}\omega_n < 0$$

$$\rightarrow 0 < \omega_n < 3$$

Select $\omega_n = 2.8$

The settling time of system:

$$T_s = \frac{4}{\zeta\omega_n} \rightarrow T_s = \frac{4}{2.8} = 1.4286\ (s)$$

From (5.15) we calculate k_1, k_2

$$k_1 = -\frac{\omega_n^2}{v_r} = -\frac{2.8^2}{0.3} = -26.1333$$

$$k_2 = \frac{0.2\omega_n^2 - 0.6\omega_n}{0.09} = \frac{0.2 \times 2.8^2 - 0.6 \times 2.8}{0.09} = -0.112$$

Then

$$K_{Pr} = -\frac{k_1}{1 + k_2 d} = -\frac{-26.1333}{1 - 0.112 \times 0.2} = 233.3333$$

$$K_{Dr} = -\frac{k_2}{1 + k_2 d} = -\frac{-0.112}{1 - 0.112 \times 0.2} = 0.1146$$

The control law to track line:

$$\mathrm{u} = \omega = 233.3333e + 0.1146\dot{e}$$

To control the robot, from (5.10)

$$u = \omega = \frac{r}{b}(\omega_R - \omega_L) \rightarrow \omega_R - \omega_L = \frac{bu}{r} \quad (5.16)$$

Our Atmega328PU microcontroller control PWM by 8-bit value (0-255).

Because the graph of PWM and angular velocity of motor are linear, we have calculated that under no load:

$$RPM = 1.02 * PWM\ value$$

In the algorithm of microcontroller:

$$PWM\ value_{right} = PWM\ base\ value - q$$

$$PWM\ value_{left} = PWM\ base\ value + q$$

$$(5.16) \rightarrow RPM_{right} - RPM_{left} = -\frac{2\pi}{60} * \frac{bu}{r} = -\frac{\pi bu}{30r}$$

$$\rightarrow PWM\ value_{right} - PWM\ value_{left} = -0.5587 \times \frac{bu}{r} = 2q$$

$$\rightarrow q = -0.2794 \times \frac{bu}{r} = -1.8054u$$

We have to recalculate when under load.

This formula assumes two motors have the same speed at the same PWM input. In reality, we will test to make the approriate calibration to formula.

5.2.3 Simulation of mobile robot system

Because our motor settling time is $0.26s$, we should set system sampling time $T_{sampling} = 0.26s$.

The result for above controller is not suitable for system real sampling time $T_{sampling} = 0.26s$ because it can't follow the line according to simulation.

We set it to $0.003s$ to test the simulation, then we will tune the PID parameter manually to reduce shaking.

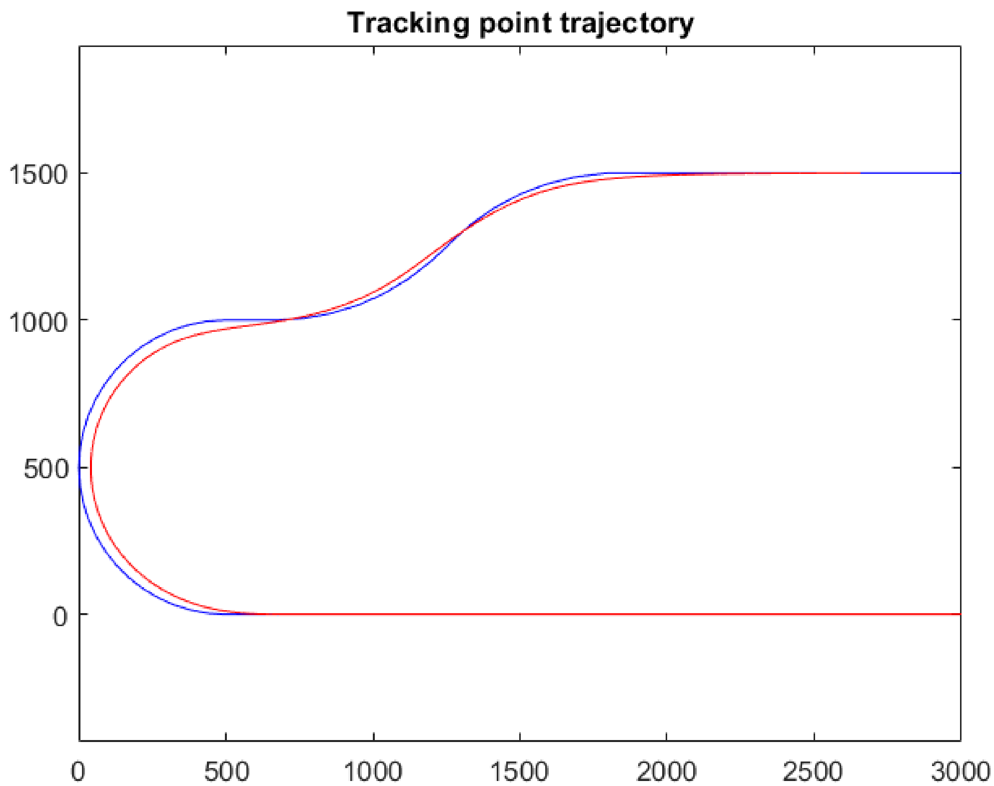


Fig 5.15 Trajectory of the tracking point

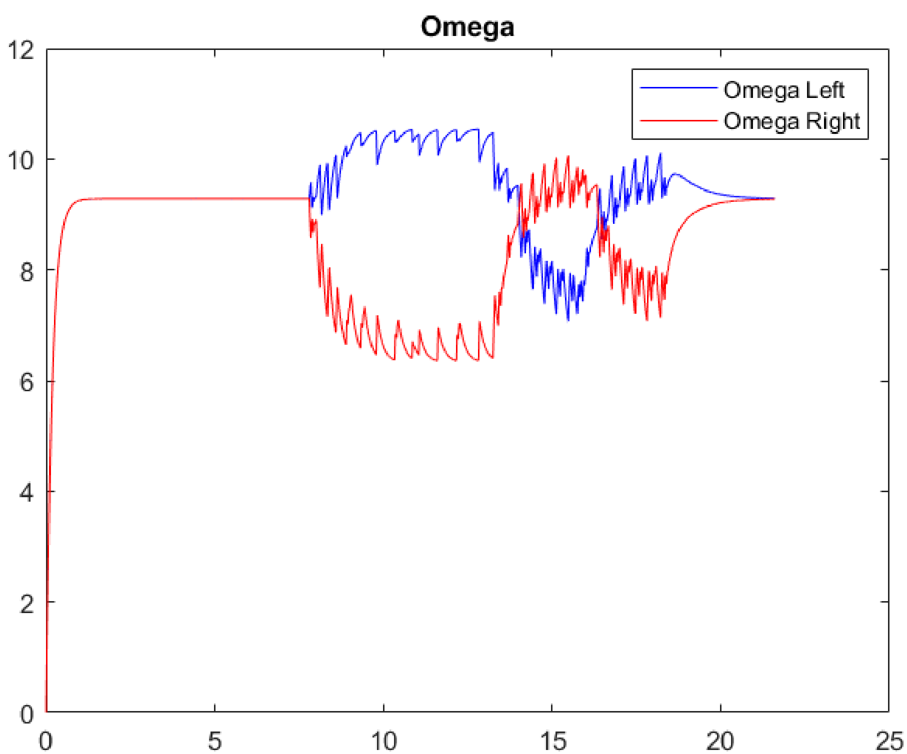


Fig 5.16 Angular velocity of 2 motors by time

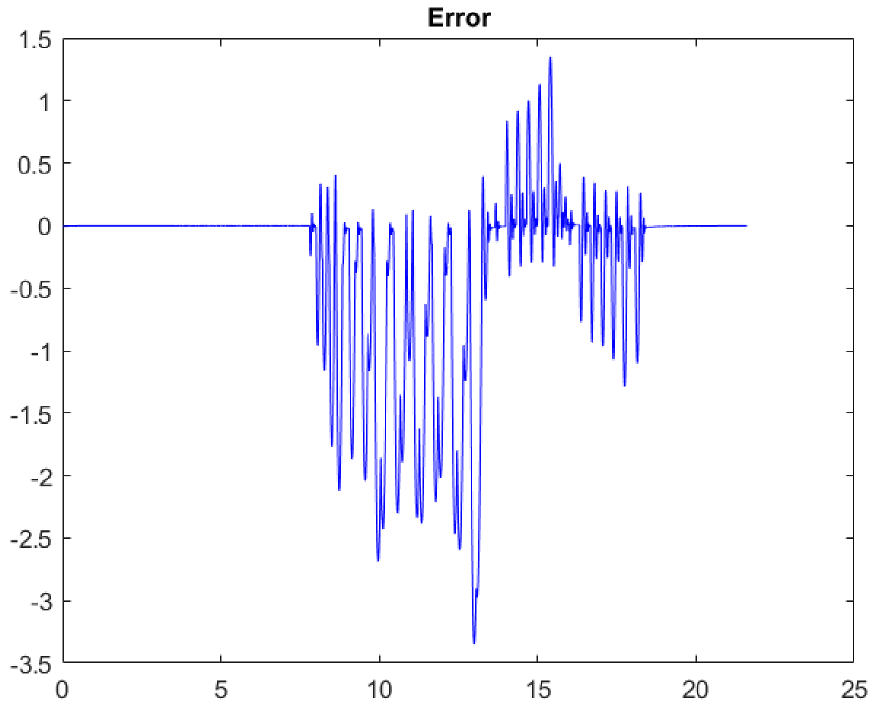


Fig 5.17 Error plot by time

Trajectory of tracking point plot: Our robot is able to follow the line.

Angular velocity plot: the velocity oscillates around our setpoint 9.2 rad/s

Error plot: Our simulation has not included the linewidth, so any value under half the wide $(13mm)$ is consider no error. For other value, the real error is equal $value - 13(mm)$

→ Our calculated controller achieved $e_{max} = 0\ mm$.

We begin tuning the PID for our system by setting $T_{sampling} = 0.26s$, then we apply the Ziegler-Nichols method for tuning [26]. Our goal is make the robot velocity's reach steady state, and it can finish the line, error can converges to 0.

First, we find the K_p gain the make the angular velocity oscillates steadily in a range. We found at $K_p = 0.29$, the robot meets above requirement:

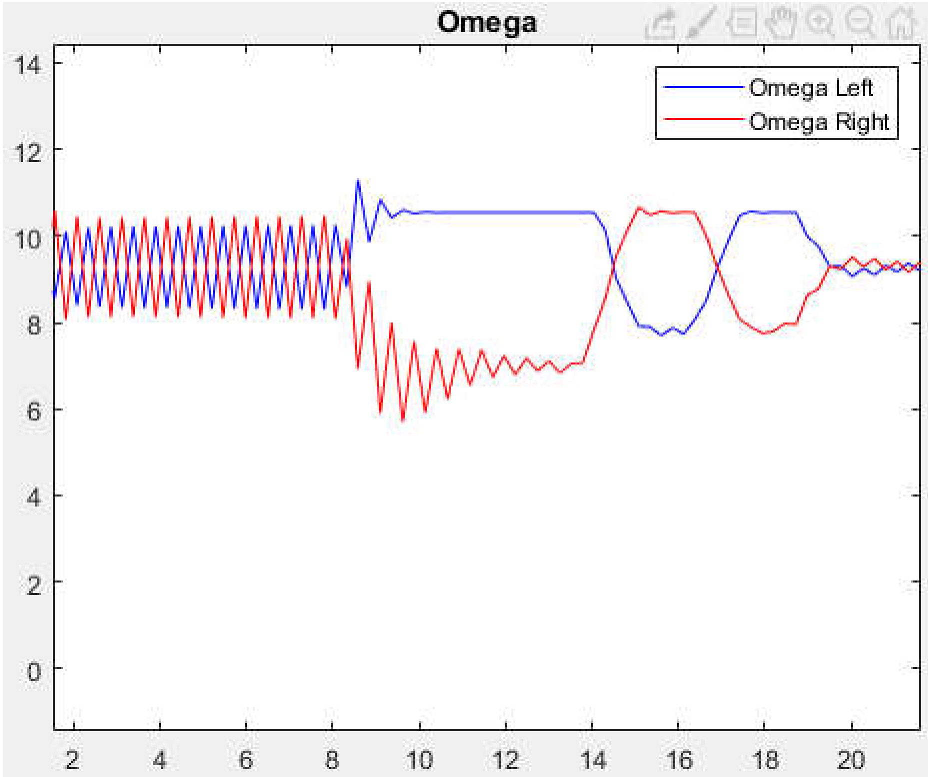


Fig 5.18 Angular velocity of 2 motors by time (Ziegler-Nichols test)

Now we apply Ziegler-Nichols rule for PID gain of the robot controller, with $P_{cr} = 0.52$:

$$K_p = 0.33K_{cr} = 0.33 \times 0.29 = 0.0967$$

$$K_i = \frac{K_p}{0.5P_{cr}} = \frac{0.09667}{0.5 \times 0.52} = 0.3718$$

$$K_d = K_p \times 0.33 \times P_{cr} = 0.0967 \times 0.33 \times 0.52 = 0.0168$$

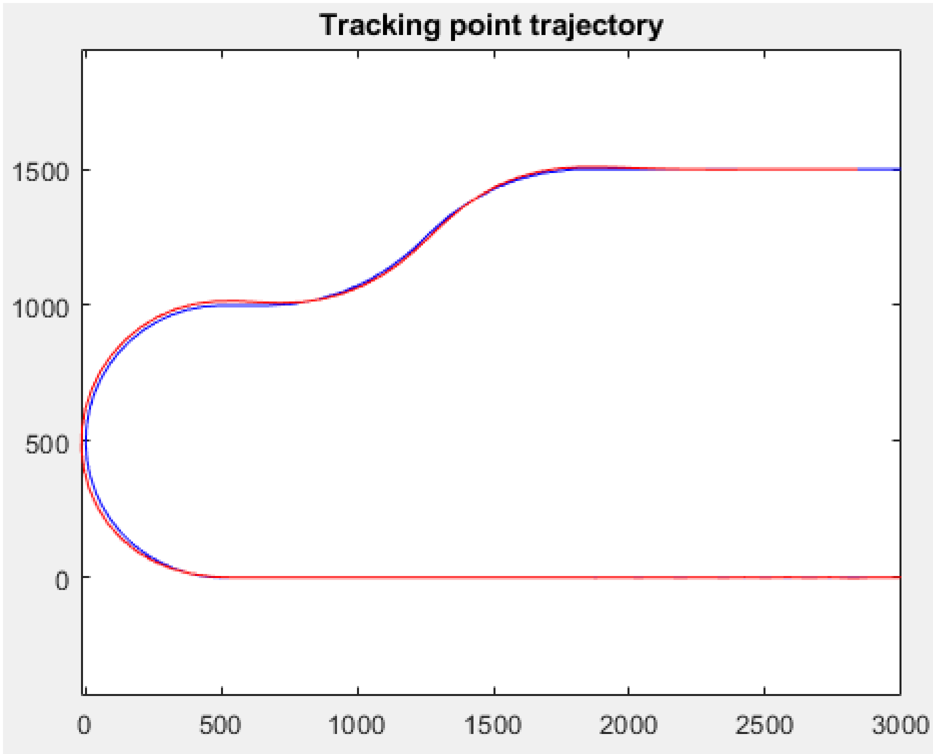


Fig 5.19 Trajectory of the tracking point after tuning

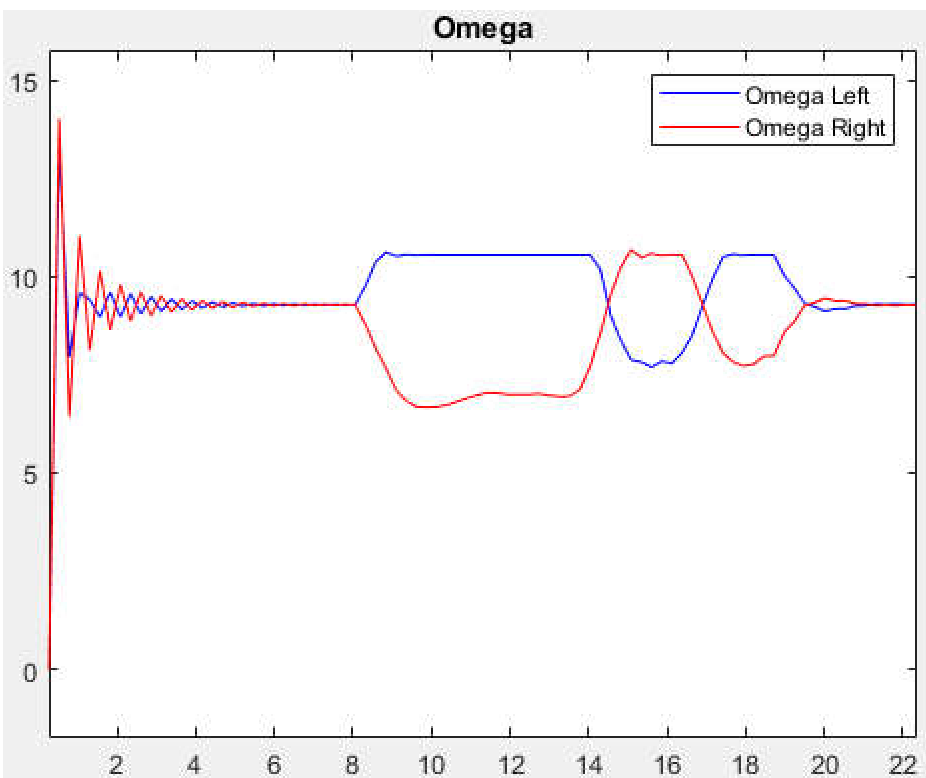


Fig 5.20 Angular velocity by time after tuning

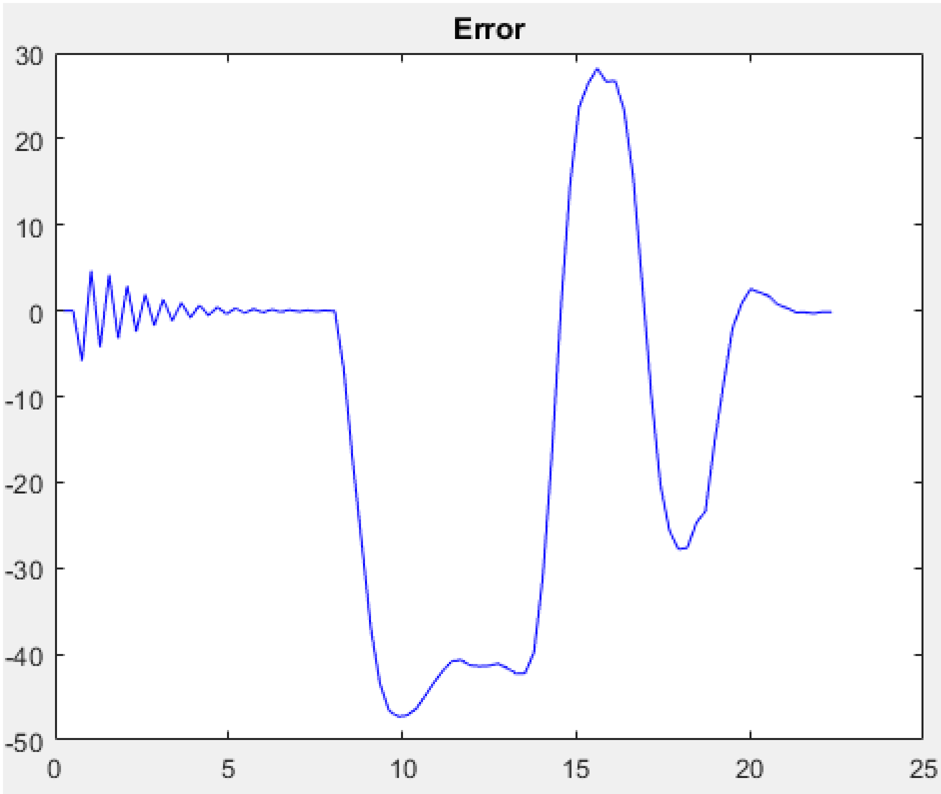


Fig 5.21 Error by time after tuning

→The robot velocity's reach steady state, and it can finish the line, error can converges to 0 on straight road. The maximum error is about $45 - 13 = 32$ (mm) at the curve

CHAPTER VI: RESULT OF TEST RUN

In real system, our robot works well with the calculated controller, not the tuned in simulation. We can also set the $T_{sampling} = 0.003\ (s)$ for the master controller, without any visible effects on the system.

After tuning, our config for PID gain of mobile robot are $K_p = 233,\ K_i = 11,\ K_d = 0.8$ for the line before loading. After loading, $K_p = 233,\ K_i = 15,\ K_d = 1$

Real error reading from the sensors:

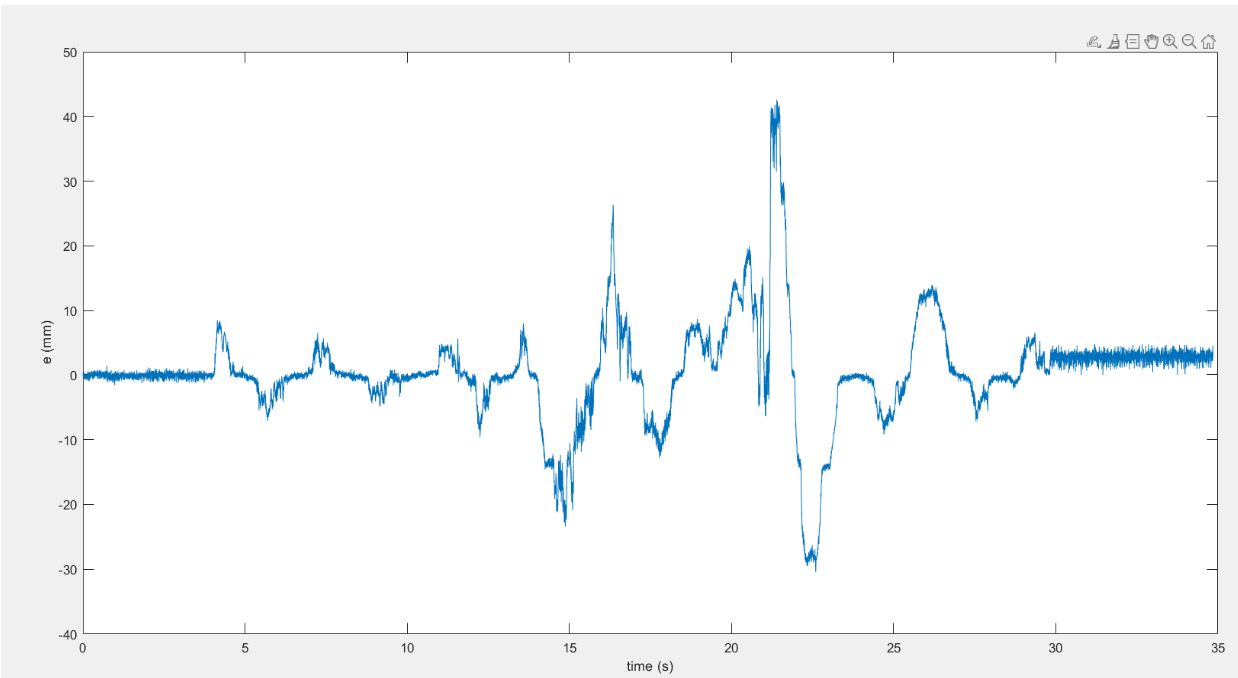


Fig 6.1 Error by time in real run

The robot velocity's reach steady state, and it can finish the line, error can converges to 0 on straight road. The maximum error is about $42 - 13 = 29\ (mm)$ at the $500mm$ curve. A video will be included to observe the error at the midpoint between 2 wheels.

Conclusion: Our robot follow the line well and it can detect the load color, following the respective line for each color. Although there maybe a little shaking during the course.

Improvement proposal: Continue to tune PID controller for both motor and system.

REFERENCES

[1] "MODEL DC-10S - Savant Automation," [Online]. Available: https://agvsystems.com/wp-content/uploads/2019/09/I_1-AGV-AGC_Model_DC-10S.pdf.

[2] "Rocla AGV Brochure," [Online]. Available: https://www.scribd.com/document/495322434/Rocla-AGV-Brochure.

[3] "The Amazon Robotics Family: Kiva, Pegasus, Xanthus, and more…," [Online]. Available: https://www.allaboutlean.com/amazon-robotics-family/.

[4] "AGC Tow Cart 500kgs," [Online]. Available: www.toyotamaterialhandling.com.au.

[5] "CarryBee(AGV)," [Online]. Available: https://www.aiki-tcs.co.jp/english/products/index.html.

[6] "Innok Hero," [Online]. Available: https://www.wevolver.com/specs/innok.

[7] G. Ullrich, in *Automated Guided Vehicle Systems: A Primer with Practical Applications*, p. 162.

[8] "Banh da huong V1," [Online]. Available: https://hshop.vn/products/banh-da-huong-v1.

[9] "161poa025p42," [Online]. Available: https://castorwheel.vn/banh-xe-noi-that/nhua-161poa025p42.html.

[10] "Pololu Ball Caster with 3/8″ Metal Ball," [Online]. Available: https://www.pololu.com/product/951.

[11] "Plastic ball caster of Hshop," [Online]. Available: https://hshop.vn/products/banh-da-huong-mat-trou-nhua-nho.

[12] "Banh mat trau kim loai," [Online]. Available: https://hshop.vn/products/banh-da-huong-mat-trou-kimloai-lon.

[13] "V1," [Online]. Available: https://hshop.vn/products/banh-xe-dong-co-dc-giam-toc-v1-plastic-geared-tt-motor-65mm.

[14] "V2," [Online]. Available: https://hshop.vn/products/banh-xe-v2-65mmkhop-luc-gioc-12mm.

[15] "V3," [Online]. Available: https://hshop.vn/products/banh-xe-v3-80mmkhop-luc-gioc-12mm.

[16] "V4," [Online]. Available: https://hshop.vn/products/banh-xe-v4-130mmkhop-luc-gioc-12mm.

[17] "V6," [Online]. Available: https://hshop.vn/products/banh-xe-v6-85mmkhop-luc-gioc-12mm.

[18] "V7," [Online]. Available: https://hshop.vn/products/banh-xe-v7-96mmkhop-luc-gioc-12mm.

[19] "V8," [Online]. Available: https://hshop.vn/products/banh-xe-v8-115mm-khop-luc-giac-12mm.

[20] "LM2596 Datasheet," [Online]. Available: https://www.alldatasheet.com/datasheet-pdf/pdf/524001/TI/LM2596.html.

[21] "Friction Coefficients," [Online]. Available: https://www.engineeringtoolbox.com/friction-coefficients-d_778.html.

[22] "Safety factors," [Online]. Available: https://www.engineeringtoolbox.com/factors-safety-fos-d_1624.html.

[23] "GM25," [Online]. Available: https://hshop.vn/products/dong-co-dc-servo-giamtoc-ga25-v1.

[24] "Ga dong co," [Online]. Available: https://hshop.vn/products/go-dong-co-duong-kenh-25mm.

[25] "I2C Bus Pullup Resistor Calculation," [Online]. Available: https://sites.ifi.unicamp.br/soares/files/2017/03/Pull-up-slva689.pdf.

[26] A. S. M. a. K. R. Godfrey, "Control Systems Technology, IEEE Transactions on February 1998," *Rule-Based Autotuning Based on Frequency,* vol. 6, no. IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, 1998.