March 2020

Design of DeLRo Autonomous Delivery Robot and AI Based Localization

Tolga Karakurt
University of South Florida

Design of DeLRo Autonomous Delivery Robot and AI Based Localization

by

Tolga Karakurt

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Electrical Engineering
Department of Electrical Engineering
College of Engineering
University of South Florida

Major Professor: Wilfrido A. Moreno, Ph.D.
Ismail Uysal, Ph.D.
Chung Seop Jeong, Ph.D.

Date of Approval:
February 26, 2020

Keywords: Mobile Robot, Skid-Steering,
Indoor Positioning, Convolutional Neural Networks (CNN), Q-Learning

For my mother

Acknowledgments

I would like first to express my deepest gratitude to my advisor, Dr. Wilfrido A. Moreno for his valuable support, constant guidance, and continuous mentorship over the last two years. He is someone with a great sense of humor and one of the smartest people I ever met. I would like to give special thanks to my committee members, Dr. Ismail Uysal and Dr. Chung Seop Jeong for their priceless thoughts and recommendations on this work. Additionally, I would like to thank to Dr. Kwang-Cheng Chen for his inspiring lectures, and guidance.

I would also like to thank my lab mates in USF DeLRo Team: Pallavi, Sebastian, Joshua, Anthony, and Ali for collaborating on the project. I am grateful towards for their helps on various parts of this project that have provided me more involvement in the lab. Many thanks to DeliverLogic Team: Joe (CEO), Matt and Susanna for sharing their expertise on project and keeping me on the scope. Special thanks to my dear friend, Tahir Topal, for countless insightful discussions and being there most of the times.

Lastly, I must express my very profound gratitude to my family for always supporting me, encouraging in all my decisions, and motivating me to pursue my career.

This work was supported by University of South Florida and the Ministry of National Education of the Republic of Turkey.

Table of Contents

List of Tables iii

List of Figures iv

Abstract vii

Chapter 1: Introduction 1
1.1 Concept of Description 1
1.2 Overview of Proposal 2
1.3 Contribution and Organization 2

Chapter 2: DeLRo Dynamic Modeling 3
2.1 System Requirements 4
2.2 Calculations and Capabilities 7
2.2.1 Four Wheeled Skid-Steering Drive System 7
2.2.2 Resistant Forces 8
2.2.3 Pivot Turning 12

Chapter 3: 4-Wheeled Robot Kinematic Model 17
3.1 DeLRo Kinematic Background 17
3.2 Model Analysis and Simulation 18
3.2.1 Skid-Steering Model 18
3.3 Lagrange Dynamical Equations 24

Chapter 4: DeLRo's Control System 29
4.1 State Space Modeling 29
4.1.1 Motor Control Modeling 30
4.1.2 PID Control 36
4.2 Block Diagram of Operation 39

Chapter 5: DeLRo Localization 43
5.1 Indoor Localization 45
5.1.1 Hardware Setup 45
5.1.2 Object Detection 48
5.1.3 Indoor Lab Environment 53
5.2 Mapping and Path Planner 53
5.2.1 Occupancy Grid and Q Learner Design 54

Chapter 6: Results 57

References.....60

List of Tables

Table 2.1 DeLRo design requirements 5

Table 2.2 DeLRo performance requirements.....6

Table 2.3 Estimated torque and power values at different climbing angles 11

Table 2.4 DeLRo's 2D spatial specifications 13

Table 2.5 Estimated spin rates at several wheel angular speeds.....15

Table 2.6 Brushless DC hub motor specifications.....16

Table 4.1 BLDC motor constant measurements 33

Table 4.2 Ziegler-Nichols (ZN) method of tuning.....37

Table 5.1 Markov decision process parameters.....55

List of Figures

Figure 1.1 System engineering “V” model 1

Figure 2.1 Comparison of operation spaces of a bicycle rider and DeLRo 3

Figure 2.2 DeLRo and pedestrian accommodation on sidewalk 4

Figure 2.3 Wheel rotational and linear speed 6

Figure 2.4 Wheel dynamics on ground level 7

Figure 2.5 Wheel dynamics at α degree slope 9

Figure 2.6 Estimated motor performance graph 12

Figure 2.7 DeLRo 2-dimensional dynamic scheme 13

Figure 2.8 DeLRo’s pivot turning diagram 14

Figure 2.9 Vendor’s motor performance curve 16

Figure 3.1 DeLRo’s simulated view on SolidWorks© 17

Figure 3.2 Force diagram 18

Figure 3.3 DeLRo kinematic model 19

Figure 3.4 Relationship between turning spin rate (Ω) and wheel angular speed (ω) 20

Figure 3.5 Motions of DeLRo 24

Figure 4.1 DeLRov0.5 interior layout (MDx: Motor driver) 29

Figure 4.2 Typical closed-loop control system 29

Figure 4.3 DC Motor electrical circuit 30

Figure 4.4 Simplified BLDC hub motor circuit 30

Figure 4.5 Simple closed-loop plant response 34

Figure 4.6 Underdamped case.....35
Figure 4.7 General PID controller block scheme.....36
Figure 4.8 Single phase plant response: K_u=0.2, T_u=0.49 [s].....37
Figure 4.9 Simulated motor speed graph (MATLAB-Simulink©) 38
Figure 4.10 Complete design of the control block and estimated system response 38
Figure 4.11 DeLRo's overall block diagram 39
Figure 4.12 Throttle D/A converter circuit 40
Figure 4.13 DAC response lag time.....40
Figure 4.14 Hall-effect buffer amplifier circuit 41
Figure 4.15 MATLAB-Simulink circle path simulation.....42
Figure 5.1 Projection of USF Engineering Building (ENB).....44
Figure 5.2 Image processor block components and system architecture.....45
Figure 5.3 MYNT EYE S1030 stereo depth camera and IMU coordinate axis 45
Figure 5.4 Leopard IMX185 CSI camera specifications 46
Figure 5.5 Geometrically distance projection between camera and object 46
Figure 5.6 DeLRo's Field of View (FOV) representation.....46
Figure 5.7 Screen shots of stereo and CSI cameras from DeLRo's perspective. 47
Figure 5.8 9x7x60mm chess board layout is used for calibration 47
Figure 5.9 Reprojection errors and parameters.....47
Figure 5.10 Custom indoor waypoints.....48
Figure 5.11 An illustration of essential 2-dimensional convolution on 3-channel input image. ...49
Figure 5.12 VGG-16 pretrained model architecture 50
Figure 5.13 Examples from our custom dataset, and instances of an augmented image.....50
Figure 5.14 Accuracy and loss curves 51
Figure 5.15 Improvement is remarkably visible, and variance is removed on the curves.....51

Figure 5.16 Total memory is approximately 27MB per image 52
Figure 5.17 Projection of USF Engineering Building (ENB).....53
Figure 5.18 Simplified grid maps 54
Figure 5.19 ENB 1st floor path planning.....55
Figure 6.1 DeLRo's overall view.....57
Figure 6.2 Test results of our trained dataset.....57
Figure 6.3 Q-Learner path generation results 58
Figure 6.4 Throttle percentage vs. motor speed curve.....59

Abstract

Autonomous robots are employed in numerous areas. In this thesis, it is proposed to design and build a self-controlled wheeled vehicle to deliver food.

As there are many applications of an autonomous agent for indoor and outdoor environments, this study is conducted on indoor settings whereas all the requirements and design processes are achieved for both operational boundaries.

The fundamental approach is to design and implement a Wheeled Mobile Robot (WMR), and to test skid-steering performance on proposed trajectories using a System Engineering approach. From this point of view, system requirements in mechanical, electrical, and software are evaluated, and overall system is divided into subblocks which are motor processor, image processor, and central processor.

One of main concerns is indoor and outdoor positioning. While outdoor tasks are widely solved in terms of the Global Positioning System (GPS) technology, indoor navigation appears with challenges. Hence, it is aimed to acquire a deeper understanding in mobile robot indoor localization through Deep Neural Networks (DNN) and learning algorithms.

Chapter 1: Introduction

1.1 Concept of Description

A four-wheeled robot, named “DeLRo”, is designed for delivery mission from point A to point B. To accomplish this primary task, it is examined through four different majors which are system dynamical behavior, kinematic capability, control hierarchy, and robot positioning. To that end, project management is handled with a System Engineering (SE) mindset following the “V” diagram, shown in Fig. 1.1, as a fundamental guidance.

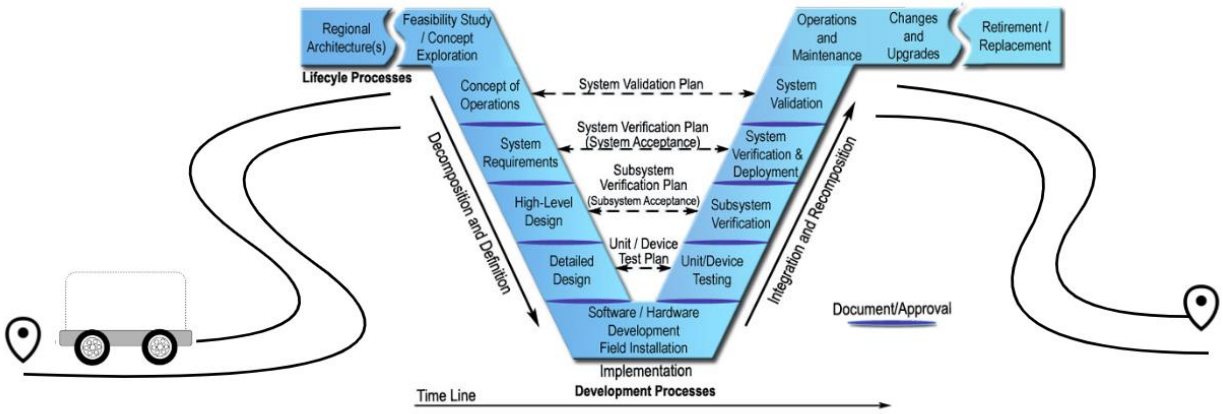


Figure 1.1 System engineering “V” model [1]

1.2 Overview of Proposal

This thesis aims to design, develop, and implement a wheeled autonomous robot for delivering tasks using a System Engineering approach for the design, test, verification, and validation processes.

1.3 Contribution and Organization

Kinematic modeling bridges the geometric properties and the capabilities of the platform. Hence, the mechanical design is bounded by the control input of the system. The dynamic design includes the measurement of the system responses under different physical scenarios, while the kinematic approach analyzes to estimate DeLRo's behavior with respect to the environmental settings.

Ultimately, this study presents a guide on how to analyze and to design a complete system from a holistic perspective. As it is crucial to use systems engineering approach in designing of such a complex system, it aids to stay on track and to tackle those detouring instances by acknowledging the five pillars of SE which are concept of operation, requirements, high level design, verification, and validation processes.

This thesis document is composed in five additional chapters. The requirements of project and calculation of robot dynamics are introduced in Chapter 2. Chapter 3 studies the mathematical modelling of system kinematics. The four-wheeled architecture and the control principles are presented in Chapter 4 including the motor control hierarchy of the design using a Proportional Integral Derivative (PID) control algorithm. Chapter 5 describes the robot localization methodology utilizing the Reinforcement Learning and Deep Neural Networks (DNN) for a pre-planned task.

Chapter 2: DeLRo Dynamic Modeling

As it is fundamental to develop and execute the mathematical model of a dynamic system, in this section we aim to derive the mathematical equilibrium to estimate and track robot's behavior.

Dynamic design of the robot has been theoretically completed and validated within the campus environment. In this regard, the first observation of the bot maneuverability is carried out in the USF Tampa campus. According to the regulations, minimum width of sidewalks is to be 5 feet [2], and DeLRo 's proposed dimension is 30 inches by 30 inches which is considered to be remained under the half of the minimum sidewalk width (1ft = 12in).

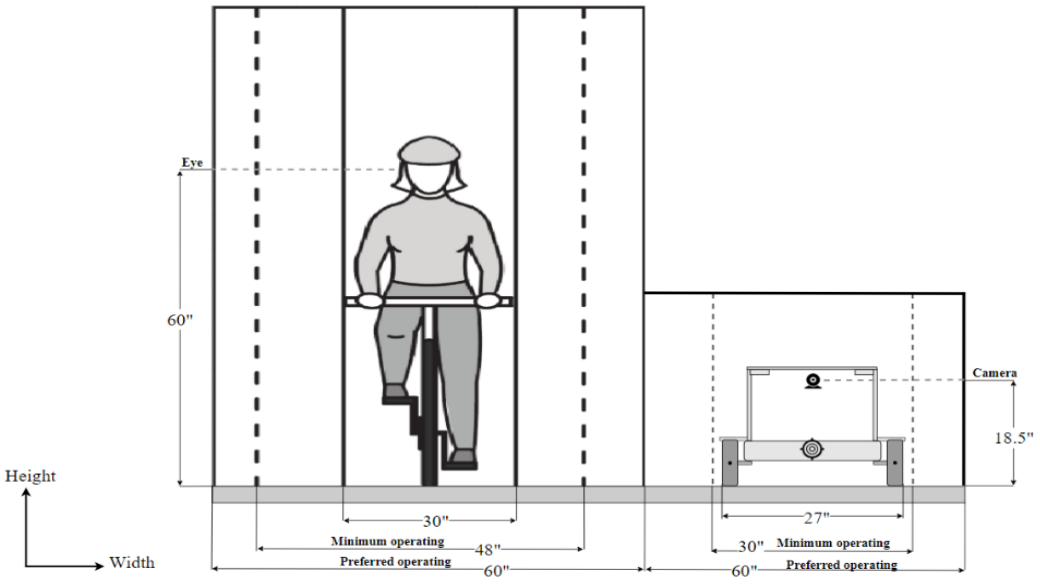


Figure 2.1 Comparison of operating spaces of a bicycle rider and DeLRo.

2.1 System Requirements

The minimum preferred operating area is calculated as 30 in same as the bike rider's spatial dimensions [3]. In this case, the number of scenarios are illustrated in Fig.2.2 which is facilitated by pedestrians and DeLRo on sidewalks. One significant approach is evaluation of the location and orientation of active objects on the runway. Four different scenarios are considered in scaling of the bot sizes to avoid confliction on the pathway.

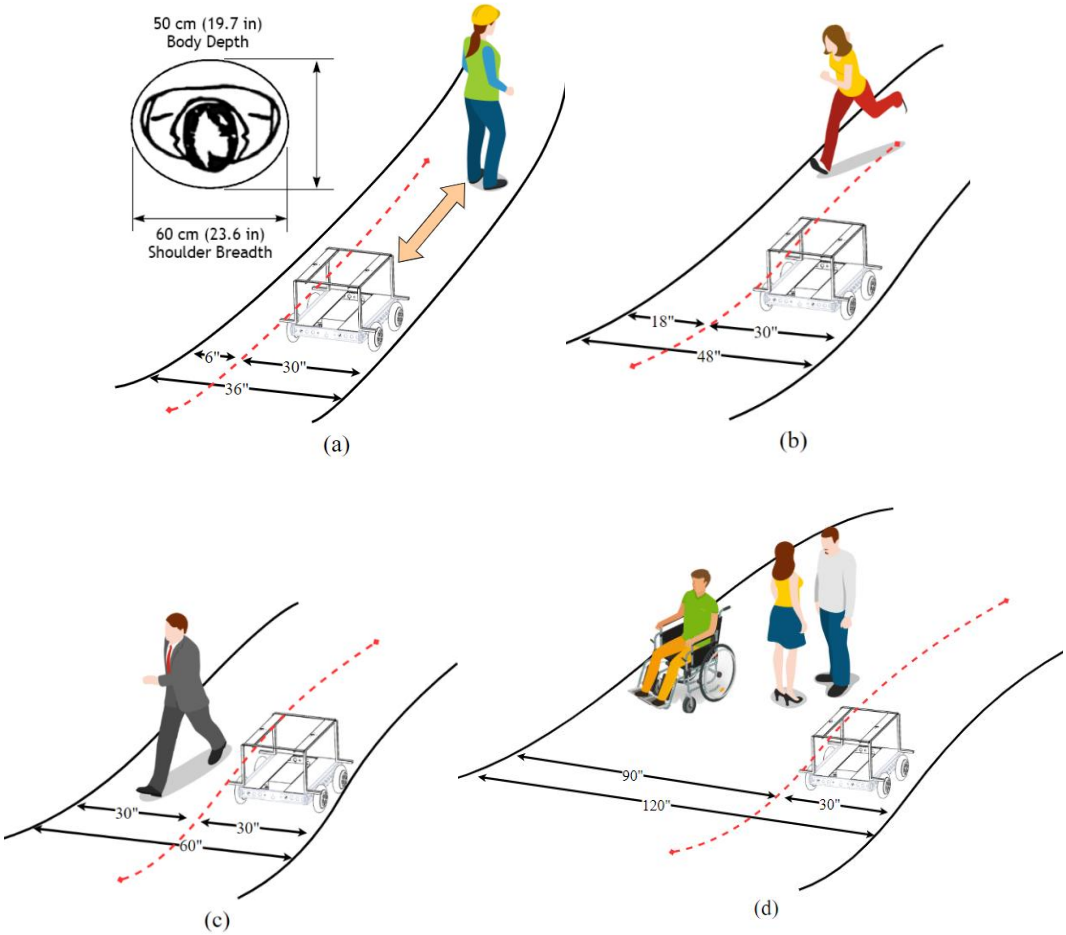


Figure 2.2 DeLRo and pedestrian accommodation on sidewalk. (The icons are obtained from https://www.iconfinder.com/)

In Fig. 2.2(a) the robot tracks the person in the same direction at 1.5 mph which is under average human walking speed. The distance between the robot and the person is the preferred

operating zone, 60 in. Fig. 2.2(b) represents that DeLRo stops and waits for the pedestrian who approaches to the bot in the case of sidewalk being under the minimum dimensions. Fig. 2.2(c) shows a typical robot-human pass scenario and the robot keeps the right pass behavior.

The average occupancy of two persons passing by is 56 inches [4]. In this case, the bot fits approximately equivalent space to an individual. For the scenarios which remains under 30 inches width, it is assumed that robot convoys a person by giving priority pass. Based on the observations and calculations it is resolved to design a wheeled agent which is 30 inches width, 30 inches length and 21inches height.

Physical capability of a system leads many fundamental questions to research on and to start building: How many wheels would be appropriate to run the robot? How much payload is allowed? What are the speed requirements? How much power is required for minimum? As environmental conditions limit the design, all these factors are cross-functional in architecture of the platform. A key point is to estimate torque parameter which brings a great significance for wheel dynamics that connect chassis to ground. In order to achieve this, it is required to create various scenarios based on these factors which aid to narrow the scope and reach the final parameter. After trade-off processes, the best and efficient parameters are listed in the following table.

Table 2.1 DeLRo design requirements

Parameter		**Value**	
W_G	Gross Weight	80	$[lb]$
r	Wheel Radius	$\frac{4}{12}$ (4 [in])	$[ft]$
v	Speed	2.2(~1.5[mph])	$\left[\frac{ft}{s}\right]$
α	Maximum Slope	40	$degree$
	Number of Wheels	4	

After a certain period of researching on motors, it is chosen to employ a brushless dc hub motor (BLDC) for the reasons which are being compact in structure and presenting high efficiency, power and torque capabilities.

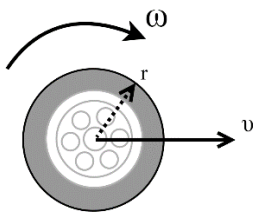


Figure 2.3 Wheel rotational and linear speed

- Angular velocity:

$$\omega = 12\frac{v\ 60}{\pi\ 2\ r}\left[\frac{rev}{min}\right];\quad 1\ [feet] = 12\ [in] \tag{2.1}$$

- Acceleration:

$$\mathrm{a} = \frac{\Delta_v}{\Delta_t} = \frac{v_{final} - v_{initial}}{t_{final} - t_{initial}}\left[\frac{ft}{s^2}\right] \tag{2.2}$$

While there are many factors such as power consumption, sidewalk occupancy, weight, and torque, velocity is essentially predicted depending on the acceleration capability of DeLRo.

Table 2.2 DeLRo performance requirements

Parameter		**Value**	
ω	Angular velocity	63	$\left[\frac{rev}{min}\right]$
a	Acceleration	1.47	$\left[\frac{ft}{s^2}\right]$
t_α	Acceleration time	1.49	$[s]$
t_o	Operation Time	2	$[h]$
η	Efficiency	85	[%]

2.2 Calculations and Capabilities

2.2.1 Four Wheeled Skid-Steering Drive System

Skid-steer system is commonly used in numerous robotics applications to facilitate a advanced load capacity, agile maneuverability, and high traction force by adjusting the number of wheels [5]. In addition, simplicity of the mechanical design is another great advantage. Therefore, it removes steering complexity and slippery problem in majority of the cases. On the other hand, controllability of the system still has challenges on odometry inconsistency and wheel synchronization.

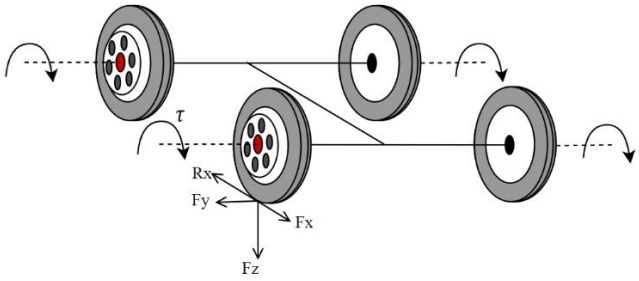


Figure 2.4 Wheel dynamics on ground level. F_x, R_x, F_y, F_z, and τ are tractive force, reactive force, lateral force, gravitational force, and applied torque, respectively.

As illustrated in the Fig.2.4, F_x is the tractive force which is occurred between the wheels and the ground as the consequence of applied torque (τ). The wheel should overcome the static friction to start rolling [6].

$$F_x \geq \mu_{ss} * R_x \tag{2.3}$$

In Eq. (2.3), μ_{ss} is static frictional coefficient that is utilized when the surfaces are stationary. It is selected as 0.8 for rubber - concrete interaction. On the other hand, μ_{ks} is known as the sliding frictional coefficient or kinetic frictional coefficient which occurs after the motion started between the objects. It is approximated around 0.45 [7].

2.2.2 Resistant Forces

All the force vectors on wheel motion which are introduced in Fig. 2.4 should be calculated to estimate torque. Hence, there are three different high level force equations considered which are rolling resistance force, ground slope effect, and aerodynamic resistivity.

1. Wheel rolling resistance force

$$R_r = C_{rr} W_G \tag{2.4}$$

where C_{rr} is rolling resistance coefficient, which is 0.01 for rubber wheel on concrete, and 0.08 for on glazed ceramic floor.

2. Ground slope

$$R_g = W_{\mathrm{G}} \sin \alpha \tag{2.5}$$

In Eq. (2.5) α stands for inclination on ground. In our test runs it is assumed as zero.

3. Aerodynamic Drag Force

This force acts in opposite direction as decelerative effect on robot's frontal area which is approximately measured as $A = 2.77 ft^2$ $(20^2 in)$. C_d is the air resistance coefficient and is assumed to be 1.05 for cubic shaped objects. Air density (ρ) is $0.00238 \left[\frac{slug}{cu\ ft}\right]$.

$$R_d = \frac{1}{2} C_d\, A\, \rho\, v^2 \tag{2.6}$$

Thus, the total resistant force is gathered as given in Eq. (2.7).

$$R_{x_{total}} = R_r + R_g + R_d \tag{2.7}$$

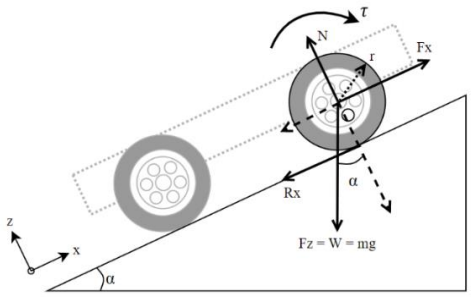


Figure 2.5 Wheel dynamics at α degree slope.

In our case,

1. Rolling resistance:

$$R_r = 0.08 * 80\ lb = 6.4\ lbf.$$

2. Ground slope affect at different inclination (α):

- $\alpha = 0$,

$$R_g = 80\ lb * \sin 0 = 0\ lbf;$$

- $\alpha = 15$,

$$R_g = 80\ lb * \sin 15 = 20.7\ lbf;$$

- $\alpha = 20$,

$$R_g = 80\ lb * \sin 20 = 27.4\ lbf;$$

- $\alpha = 40$,

$$R_g = 80\ lb \sin 40 = 51.4\ lbf.$$

3. Aerodynamic drag force:

$$R_d = \frac{1}{2} 1.05\ 2.77 ft^2\ \ 0.00238 \frac{slug}{cu\ ft} * \left(2.2 \frac{ft}{s}\right)^2 = 0.17\ lbf$$

Finally, the sum of the resistive forces is derived at different slopes (α) as follows.

- $\alpha = 0$,

$$R_{x_{total}} = R_r + R_g + R_d = 6.4\ lbf + 0\ lbf + 0.17\ lbf = 6.57\ lbf$$

- $\alpha = 15$,

$$R_{x_{total}} = 27.3\ lbf$$

- $\alpha = 20$,

$$R_{x_{total}} = 34\ lbf$$

- $\alpha = 40$,

$$R_{x_{total}} = 58\ lbf$$

The gross weight which is 80 pounds is distributed to wheels homogenously. In this case each wheel has gravity force (F_z) under approximately one-fourth of mass. In order to specify motor parameters for each wheel, it is required to answer how much tractive torque is demanded to beat the resistance. The total torque (τ_t) and power (P) is calculated as follows.

$$\sum F = m\mathrm{a} \tag{2.8}$$

$$F_x - \mu_{ss} R_x = \frac{W_\mathrm{G}}{4}\mathrm{a}_x \tag{2.9}$$

$$F_y - R_y = \frac{W_\mathrm{G}}{4}\mathrm{a}_y, \left(F_y = 0\right) \tag{2.10}$$

$$W_\mathrm{G}\mathrm{a}_x = F_{x_{total}} - \mu_{ss} R_{x_{total}} \tag{2.11}$$

$$W_\mathrm{G}\mathrm{a}_x = \frac{\tau_t}{r} - \mu_{ss}\frac{\tau_{Rx}}{r} \tag{2.12}$$

$$\tau_t = rW_\mathrm{G}\mathrm{a}_x + \mu_{ss}\tau_{Rx} \tag{2.13}$$

$$P_{mechanical} = 0.0118\ \tau_t\ \omega \tag{2.14}$$

$$P_{electrical} = P_{mechanical}\left(\frac{1}{n}\right) \tag{2.15}$$

In Eq. (2.14) 0.0118 is conversion factor. The description of the factor is illustrated in following equilibriums.

$$1\ [hp]\ =\ 745.699872\ [watt] \tag{2.16}$$

$$P_{hp} = \frac{\tau_t\ \omega}{5252}(lbft\ rev/min) = \frac{1}{12}\frac{\tau_t\ \omega}{5252}[lb.in\ rev/min] \tag{2.17}$$

$$P_w = P_{hp}\ 745.7 = 745.7\frac{1}{12}\frac{\tau_t\ \omega}{5252} \tag{2.18}$$

$$P_w = 0.0118\ \tau_t\ \omega\ [lb.in\ rev/min] \tag{2.19}$$

Table 2.3 Estimated torque and power values at different climbing angles

Parameter	**Value**				**Unit**
Ground Slope	0^o	15^o	20^o	40^o	$[degree]$
Angular velocity (ω)	63	63	63	63	$\left[\frac{rev}{min}\right]$
Acceleration (a)	1.47	1.47	1.47	1.47	$\left[\frac{ft}{s^2}\right]$
Wheel Radius (r)	0.33	0.33	0.33	0.33	$[ft]$
Total resistant force (R_x)	6.57	27.3	34	58	$[lbf]$
Torque (τ_t)	40.1	46.8	48.3	54.7	$[lb.in]$
Power ($P_{mechanical}$)	29.8	37.8	35.9	40.7	$[watt]$
Power ($P_{electrical}$)	35.1	44.5	42.2	47.9	$[watt]$
Efficiency (n)	85	85	85	85	[%]

Motor performance curve can be predicted in terms of the given formulas below by deploying design and performance parameters. In Eq. (2.20) τ_s is maximum stall torque while motor is not rotating which is approximated to 100 [lb.in]. ω_0 indicates maximum velocity of the motor without load and it is assumed to be 900 [rev/min]. Lastly, I_0 and I_s are no-load current and stall current, respectively.

$$\tau = \tau_s(1 - \frac{\omega}{\omega_0}) \tag{2.20}$$

$$I = I_0 + (I_s - I_0)(1 - \frac{\tau}{\tau_s}) \tag{2.21}$$

$$P = 0.0118\,\tau\,\omega\frac{1}{n} \tag{2.22}$$

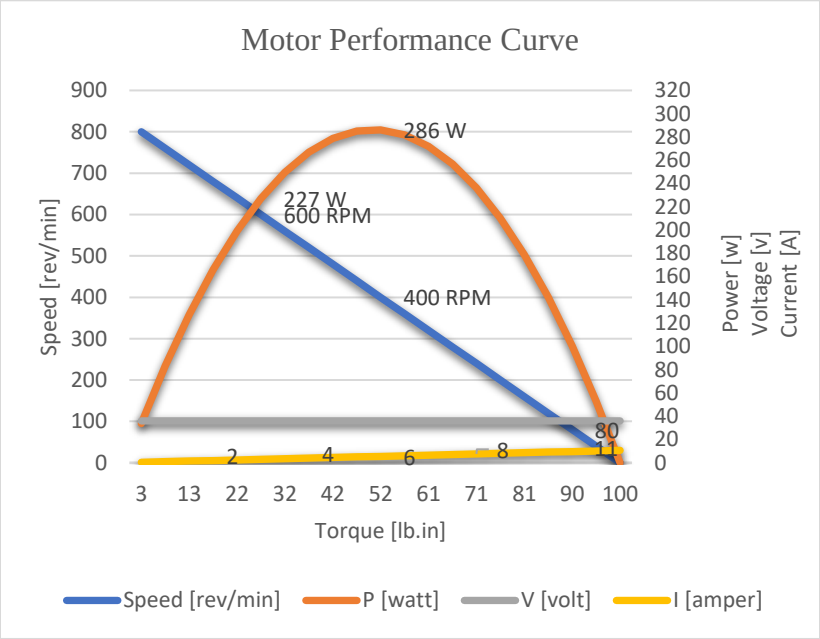


Figure 2.6 Estimated motor performance graph

2.2.3 Pivot Turning

First, skid-steering operation is preferably conducted on a flat indoor surface. Fig. 2.7 illustrates DeLRo's geometrical dimensions, and "O" is assumed as geometrical center and the center of Mass (CoM) of the bot. Hence, this enables a lightweight computation in system dynamics. Dimensions are as same as the real design, and the mass distribution is presumed as homogenous.

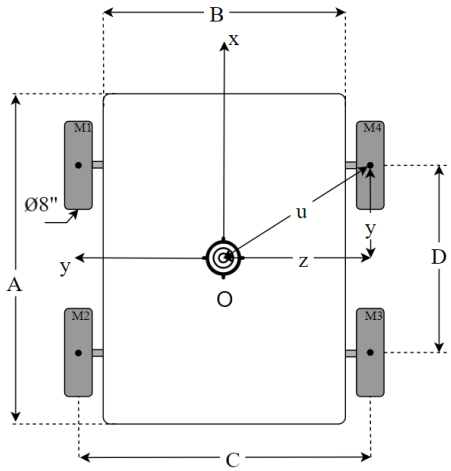


Figure 2.7 DeLRo 2-dimensional dynamic scheme

Table 2.4 DeLRo's 2D spatial specifications

Dimensions		**Value**	**Unit**
A	*length of robot*	30	$[in]$
B	*width of trunk*	21.5	$[in]$
C	*width of robot*	24	$[in]$
D	*distance between wheels*	16	$[in]$
u	*distance between central point and wheel*	14.5	$[in]$
z	*horizontal length toward wheel axis*	12	$[in]$
y	*vertical length toward wheel axis*	8	$[in]$
Ø	*diameter of wheel*	8	$[in]$

Aforementioned forces are shown in Fig. 2.8 to present a descriptive picture of tractive (F_{x_i}), lateral (F_{y_i}), reactive (R_{x_i}), and friction forces (F_{xr_i}) for $i = 1,2,3,4$. Omega (**Ω**) is spin rate or angular turning speed of the body. To remember, in Eq. (2.3) traction force should achieve the reactive force to enable the wheel's forward rotation. However, this principle remains plain while lateral forces participate in resistance torque (τ_{Rx}) beside the tractive torque (τ_t) [8].

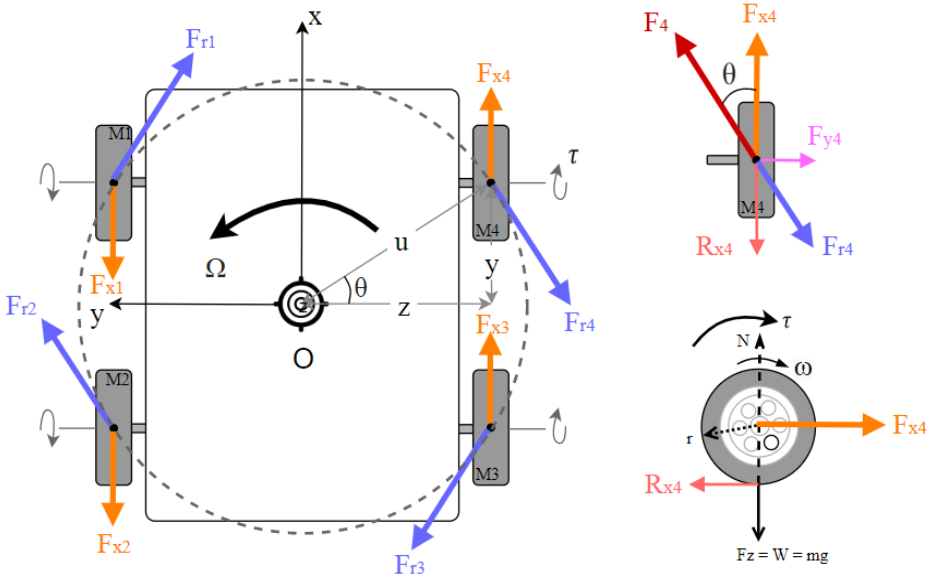


Figure 2.8 DeLRo's pivot turning diagram

$$\tau_{Rx} = \tau_{Rx_1} + \tau_{Rx_2} + \tau_{Rx_3} + \tau_{Rx_4} \tag{2.23}$$

$$\tau_t = \tau_{t_1} + \tau_{t_2} + \tau_{t_3} + \tau_{t_4} \tag{2.24}$$

Eq. (2.25) is fundamental for pivot turning.

$$\tau_t \geq \tau_{Rx} \tag{2.25}$$

Therefore, wheel tractive torque is given in Eq. (2.26) for $i = 1,2,3,4$.

$$\tau_{t_i} = \frac{\tau_t}{4}; \tag{2.26}$$

$$\tau_{Rx_i} = \frac{\tau_{Rx}}{4}; \tag{2.27}$$

Resistant torque is also presented above. Total moment value is the key point to initiate the movement of the agent. In order to gather the equivalent formula of pivot turning, the total moment in DeLRo's center is obtained as in Eq. (2.28)

$$\frac{\tau_{t_i}}{r} z \geq \frac{W_G}{4} \mu_{ks}\, y \tag{2.28}$$

According to the equation above the minimum torque value for each wheel is estimated as greater than 24 [$lb.in$].In this case the total traction thrust cannot be under 96 [$lb.in$] for rubber wheel on concrete surface which utilizes the kinetic (or sliding) friction coefficient approximation, μ_{ks}, is around 0.45.

In addition to tractive torque, it is also required to calculate the spin rate (Ω) which is the rotation speed at zero radius skid steer turn on 0-degree ground.

$$\Omega = \frac{\omega \, \mathrm{r}}{u} \cos\theta \ \left[\frac{rev}{min}\right] \tag{2.28}$$

$$\Omega = \frac{2\,\pi}{60} \frac{\omega \, \mathrm{r}}{u} \cos\theta \left[\frac{rad}{s}\right] \tag{2.29}$$

According to Eq. (2.29) the period of a complete turn can be determined. For instance, once a wheel rotates at 100 RPM and a full circle is 2π radian, full rotation is fulfilled in 2.62 seconds.

$$\frac{2\pi \ [\mathrm{rad}]}{2.4 \left[\frac{\mathrm{rad}}{\mathrm{s}}\right]} = 2.62 \ [s] \tag{2.30}$$

Table 2.5 Estimated spin rates at several wheel angular speeds

Parameter	**Value**				**Unit**
Angular velocity (ω)	80	90	100	110	$\left[\frac{rev}{min}\right]$
Spin rate (Ω)	18.3	20.6	22.3	25	$\left[\frac{rev}{min}\right]$
Spin rate (Ω)	1.9	2.2	2.4	2.6	$\left[\frac{rad}{s}\right]$
Torque (τ_x)	96	96	96	96	$[lb.in]$
Power ($P_{mechanical}$)	90.62	102	113.3	124.6	$[watt]$
Power ($P_{electrical}$)	113.3	127.4	141.6	155.8	$[watt]$
Efficiency (η)	85	85	85	85	[%]

Eventually, 350 [watt] BLDC hub (in-wheel) motor is selected. One of the reasons is that presenting high torque capacity while eliminating dependency on additional shafting and transmission components Hence, choosing a BLDC driver circuit becomes advantageous. Another usefulness is in-circuit hall-effect sensor for concise position readings in the circuit feedback loop.

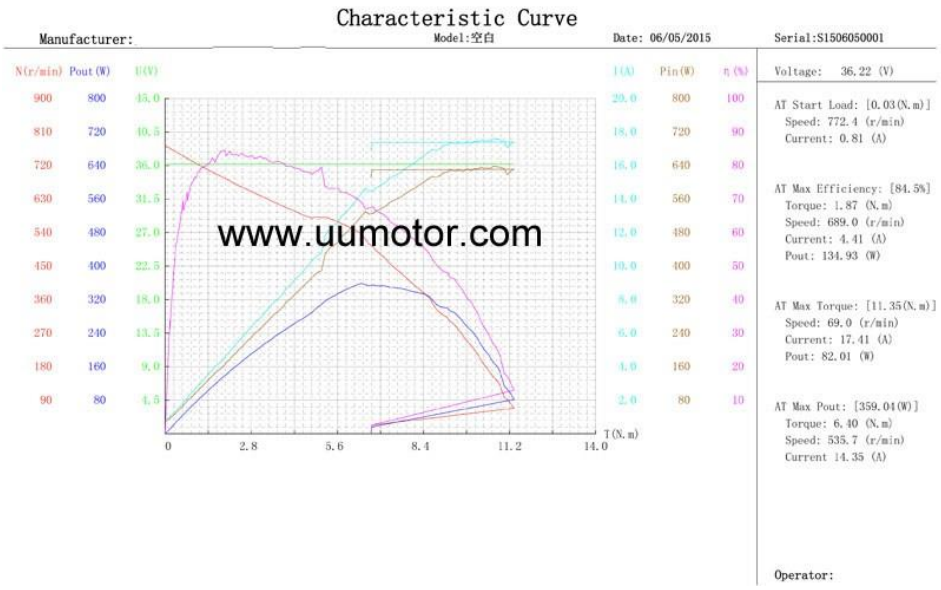


Figure 2.9 Vendor's motor performance curve

BLDC hub motors have a wide range of applications in electric bicycles, scooters, cars, and so on. Hence, hub motors, which are also known as direct-driven hub wheels are directly driven through magnetic field between stator coils and rotor magnets, present good acceleration, convincing lifecycle, and high efficiency and power [9].

Table 2.6 Brushless DC hub motor specifications

Motor Ratings and Parameters	Value	Unit
Power	350	$[Watts]$
Torque	90	$[lb.in]$
Voltage	39	$[Volts]$
Current	10	$[A]$
Diameter	8	$[in]$
Weight	6.4	$[lb]$
Efficiency	85	[%]
Speed	825	$[rev/min]$

Chapter 3: 4-Wheeled Robot Kinematic Model

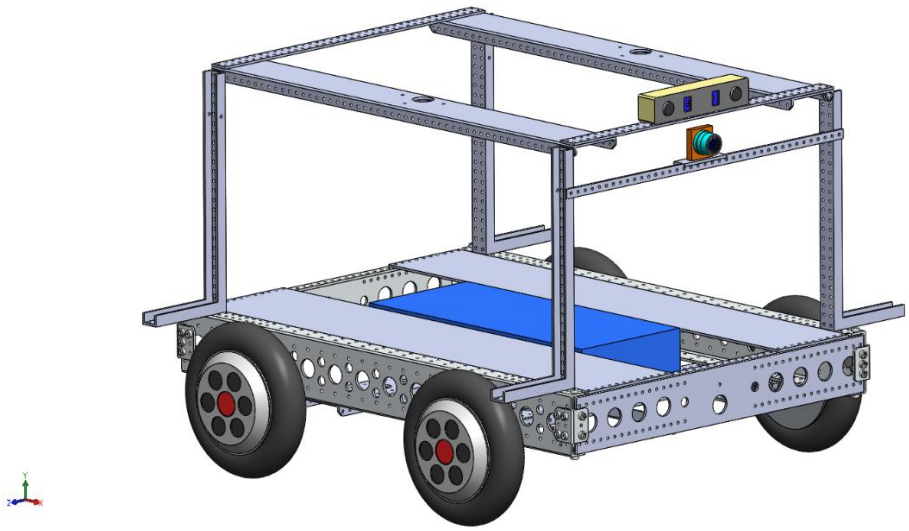


Figure 3.1 DeLRo's simulated view on SolidWorks©

In this section, the mathematical equilibriums of robot kinematics are described and state-space model of 4-wheeled skid-steering system is developed. Hence, it is aimed to estimate the velocity, acceleration, position, and orientation of the bot.

3.1 DeLRo Kinematic Background

DeLRo is a four wheeled robot and it is considered moving in two-dimensional global coordinate system which is defined as (X_G, Y_G) in Figure 3.3.

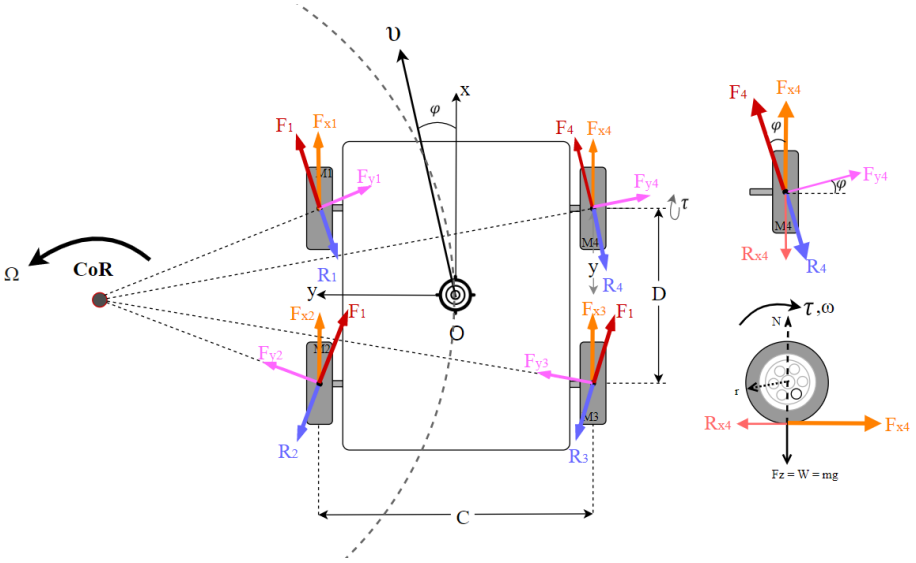


Figure 3.2 Force diagram. CoR represents the center of rotation.

$$F_{xL} = F_{x1} + F_{x2}, \qquad F_{xR} = F_{x3} + F_{x4} \tag{3.1}$$

$$R_{xL} = R_{x1} + R_{x2}, \qquad R_{xR} = R_{x3} + R_{x4} \tag{3.2}$$

$$F_{y_f} = F_{y1} + F_{y4}, \qquad F_{y_r} = F_{y2} + F_{y3} \tag{3.3}$$

$$W_{\mathrm{G}}\mathrm{a}_x = F\cos\varphi - \mu_{ss}\left(R\cos\varphi - F_y\sin\varphi\right) \tag{3.4}$$

$$W_{\mathrm{G}}\mathrm{a}_y = F\sin\varphi - \mu_{ss}\left(R\sin\varphi + F_y\cos\varphi\right) \tag{3.5}$$

Therefore, vehicle's moment equation can be written as in Eq. (3.6). I_O represents moment of inertia of the entire bot, $\ddot{\varphi}$ denotes spin acceleration $(\dot{\Omega})$, and M_r means resistive moment.

$$M_r = \frac{C}{2}R_{xR}\cos\varphi - \frac{C}{2}R_{xL}\cos\varphi + \frac{D}{2}F_{y_f}\cos\varphi - \frac{D}{2}F_{y_r}\cos\varphi \tag{3.6}$$

$$I_O\ddot{\varphi} = \frac{C}{2}(F_{xR} - F_{xL}) - M_r \tag{3.7}$$

While it is non-steering system, wheels are powered through left and right group of motor drivers. Each wheel is equipped with hall effect sensors which produces 5V peak to peak electrical square waves.

3.2 Model Analysis and Simulation

3.2.1 Skid-Steering Model

In cartesian coordinate system, Center of Mass (CoM) is represented as O reference point that is the geometrical midpoint of DeLRo which has symmetrical properties in x and y axes. Center of Rotation (CoR) is identified to illustrate turning kinematics (x_{CoR}, y_{CoR}) [10], [11], [12]. Hence, it allows comprehensive modeling and simulating of waypoint-oriented tasks that is explained in Chapter 5 in details.

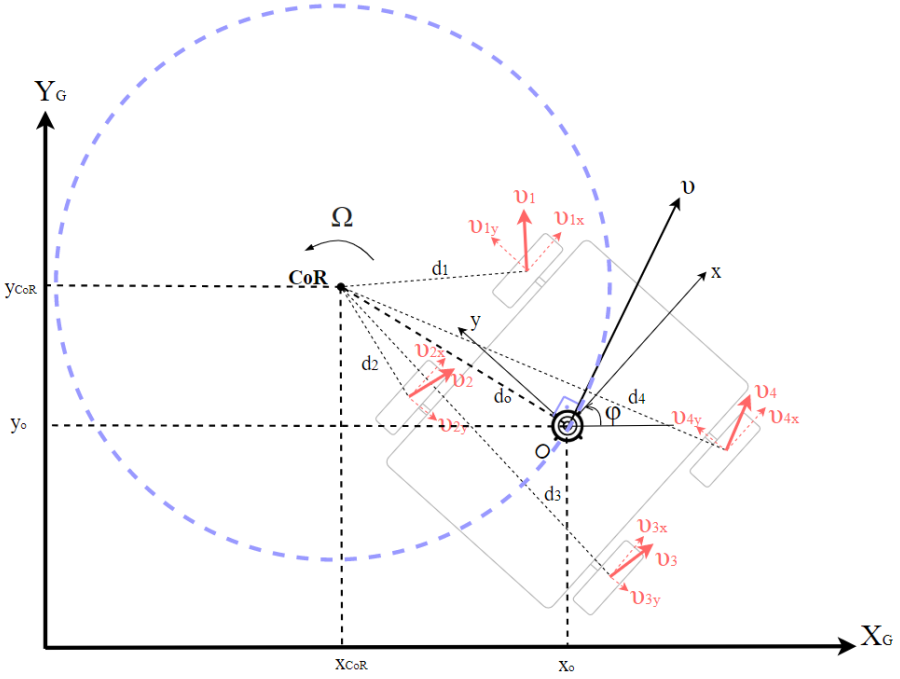


Figure 3.3 DeLRo kinematic model

Pose vector is defined as in Eq. (3.8). X, Y, φ denote lateral coordinate, longitudinal coordinate, and orientation respectively.

$$q = \begin{bmatrix} X \\ Y \\ \varphi \end{bmatrix} \tag{3.8}$$

Hence, vehicle's body velocity is given in terms of lateral and longitudinal speeds in Eq. (3.9).

$$v = \left[v_{x,}\, v_y\right]^T \tag{3.9}$$

and velocities of wheels can be denoted as in Eq. (3.10) and Eq. (3.11).

$$v_1 = \left[v_{1x,}\, v_{1y}\right]^T, v_2 = \left[v_{2x,}\, v_{2y}\right]^T \tag{3.10}$$

$$v_3 = \left[v_{3x,}\, v_{3y}\right]^T, v_4 = \left[v_{4x,}\, v_{4y}\right]^T \tag{3.11}$$

It is designed that set of left wheels (M1 and M2) has the common accelerating capability, and therefore longitudinal velocities are equal, as the right wheels (M3 and M4). Similarly, lateral speeds are identical.

$$v_{1x} = v_{2x}\,, v_{3x} = v_{4x} \tag{3.12}$$

$$v_{1y} = v_{3y}\,, v_{2y} = v_{4y} \tag{3.13}$$

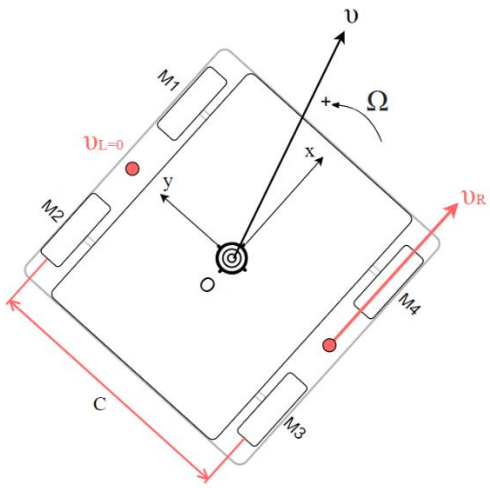


Figure 3.4 Relationship between turning spin rate (Ω) and wheel angular speed (ω)

As depicted in Fig. 3.4, it can be considered that velocity of the left sided motors is degraded to zero to find the body orientation.

$$v_L = v_{1x} = v_{2x}, v_R = v_{3x} = v_{4x} \tag{3.14}$$

Eq. (3.14) can also be written in Eq. (3.15) to culminate to body longitudinal speed by accumulating linear velocities of the motors.

$$v_x = \frac{v_L + v_R}{2} \tag{3.15}$$

$$v_L = \frac{v_1 + v_2}{2}, v_R = \frac{v_3 + v_4}{2} \tag{3.16}$$

$$v_L = \frac{\omega_1 + \omega_2}{2} r, v_R = \frac{\omega_3 + \omega_4}{2} r \tag{3.17}$$

$$v_R - v_L = \Omega C \tag{3.18}$$

$$\frac{(\omega_3 + \omega_4)}{2} r - \frac{(\omega_1 + \omega_2)}{2} r = \Omega C \tag{3.19}$$

Based on the given parameters in Table 2.5, forward kinematic equation can be written in Eq (3.20).

$$\begin{bmatrix} v_x \\ v_y \\ \Omega \end{bmatrix} = \frac{r}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{C} & -\frac{1}{C} & \frac{1}{C} & \frac{1}{C} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \tag{3.20}$$

where v_x is longitudinal speed of the bot is the accumulation of the speeds of the wheels. Since lateral velocities have comparatively less greatness, are approximated to zero in the calculations [13].

Inverse kinematic equation can be also derived in Eq. (3.21).

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} 1 & 0 & -C \\ 1 & 0 & -C \\ 1 & 0 & C \\ 1 & 0 & C \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \Omega \end{bmatrix} \tag{3.21}$$

Simultaneous position of the bot is calculated through x-y local coordinate base. In this case, a relationship can be created between global X-Y coordinate system and the robot's inherent x-y positioning frame. Thus, this approach allows to track immediate pose of the platform on a proposed trajectory [14].

$$\dot{X} = \dot{x}_o = v_x \cos\varphi - v_y \sin\varphi \tag{3.22}$$

$$\dot{Y} = \dot{y}_o = v_x \sin\varphi + v_y \cos\varphi \tag{3.22}$$

$$\dot{\varphi} = \Omega \tag{3.23}$$

$$\dot{q} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \dot{x}_o \\ \dot{y}_o \\ \Omega \end{bmatrix} = \begin{bmatrix} cos\varphi & -sin\varphi & 0 \\ sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \Omega \end{bmatrix} \tag{3.24}$$

Linear speed of the wheels ($v_i = \{v_1, v_2, v_3, v_4\}$) is represented in terms of multiplication of distance vectors which is between wheels ($d_i = \{d_1, d_2, d_3, d_4\}$) and CoR ($x_{CoR}, y_{CoR}$), and wheel angular velocities ($\omega_i = \{\omega_1, \omega_2, \omega_3, \omega_4\}$).

$$v_{ix} = r\,\omega_i \tag{3.25}$$

$$v_i = d_i\,\Omega \tag{3.26}$$

$$v_{ix} = -d_{iy}\,\Omega, \quad v_{iy} = d_{ix}\,\Omega \tag{3.27}$$

Although we ignore the lateral slipping, it must be considered as a non-holonomic constraint. In this case, d_{ox}, the longitudinal component of d_o which is distance between CoR and CoM, denotes the slip that is depicted in Fig. 3.2 and is given in Eq. (3.28).

$$v_y + d_{ox}\,\Omega = 0 \tag{3.28}$$

and if we reorganize Eq. (3.24) to obtain v_y by multiplying $\dot{X}$ with $(-\sin\varphi)$ and $\dot{Y}$ with $(\cos\varphi)$, we accumulate the Eq. (3.29).

$$(-\sin\varphi)\dot{X} = (-\sin\varphi)\left(v_x \cos\varphi - v_y \sin\varphi\right) \tag{3.29}$$

$$(\cos\varphi)\dot{Y} = (\cos\varphi)\left(v_x \sin\varphi + v_y \cos\varphi\right) \tag{3.30}$$

$$-\sin\varphi\,\dot{X} + \cos\varphi\,\dot{Y} = v_y \tag{3.31}$$

Rewriting Eq. (3.21) only for a wheel's rotational speed, it is given in the following equation.

$$\omega_1 = \frac{1}{r}(v_x - C\dot{\varphi}) \tag{3.32}$$

If we repeat the same calculation that is done for Eq. (3.29), this time we can derive v_x in Eq. (3.35).

$$(\cos\varphi\,)\dot{X} = (\cos\varphi)\big(v_x\cos\varphi - v_y\sin\varphi\big) \tag{3.33}$$

$$(\sin\varphi\,)\dot{Y} = (\sin\varphi\,)\big(v_x\sin\varphi + v_y\cos\varphi\big) \tag{3.34}$$

$$\cos\varphi\,\dot{X} + \sin\varphi\;\dot{Y} = v_x \tag{3.35}$$

Hence, the wheel angular velocity can be described in terms of the components of $\dot{q}$ from Eq. (3.24) and it is given in Eq. (3.38).

$$\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} = \frac{1}{r}\begin{bmatrix}\cos\varphi & \sin\varphi & -C\\ \cos\varphi & \sin\varphi & -C\\ \cos\varphi & \sin\varphi & C\\ \cos\varphi & \sin\varphi & C\end{bmatrix}\begin{bmatrix}\dot{X}\\ \dot{Y}\\ \dot{\varphi}\end{bmatrix} \tag{3.36}$$

$$\omega_1 = \frac{1}{r}\big(\cos\varphi\,\dot{X} + \sin\varphi\;\dot{Y} - C\dot{\varphi}\big) \tag{3.37}$$

$$\dot{q} = \begin{bmatrix}\dot{X}\\ \dot{Y}\\ \Omega\end{bmatrix} = \frac{r}{4}\begin{bmatrix}\cos\varphi & \cos\varphi & \cos\varphi & \cos\varphi\\ \sin\varphi & \sin\varphi & \sin\varphi & \sin\varphi\\ -\frac{1}{C} & -\frac{1}{C} & \frac{1}{C} & \frac{1}{C}\end{bmatrix}\begin{bmatrix}\omega_1\\ \omega_2\\ \omega_3\\ \omega_4\end{bmatrix} \tag{3.38}$$

Linear non-holonomic constraint matrix can be identified as in Eq. (3.39).

$$[-\sin\varphi \quad \cos\varphi \quad d_{ox}]\begin{bmatrix}\dot{x}_o\\ \dot{y}_o\\ \Omega\end{bmatrix} = A^T(q)\dot{q} = 0, \qquad \Omega = \dot{\varphi} \tag{3.39}$$

where $A^T(q)$ is defined as $[-\sin\varphi \quad \cos\varphi \quad d_{ox}]$. Our control input parameters are wheel angular velocity.

DeLRo, as explained before and shown in Fig. 3.4, is designed and built based on left (M1, M2) and right sided (M3, M4) controllable motor sets. This means, wheels are driven through

individual motor drivers, and each wheel group is commanded in term of a common control parameter. Generalized pseudo velocity (η) is defined [15] in Eq. (3.40).

$$\eta = \begin{bmatrix} \upsilon_x \\ \upsilon_y \\ \Omega \end{bmatrix} \tag{3.40}$$

$$\dot{q} = S(q)\eta \tag{3.41}$$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\varphi} \end{bmatrix} = S(q) \begin{bmatrix} \upsilon_x \\ \upsilon_y \\ \Omega \end{bmatrix}, \qquad S(q) = \begin{bmatrix} cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.42}$$

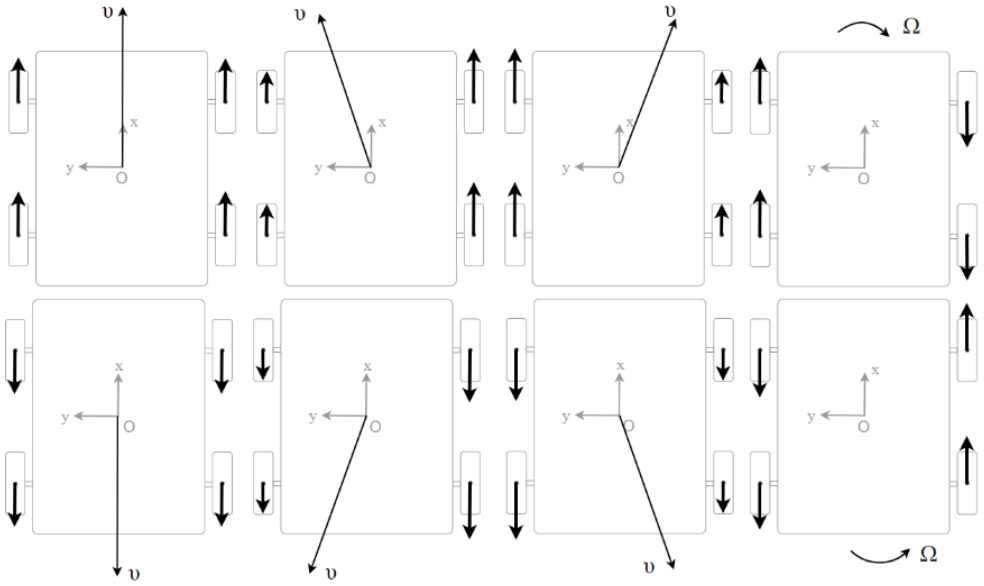


Figure 3.5 Motions of DeLRo.

3.3 Lagrange Dynamical Equations

Mostly for non-holonomic systems, utilization of energy equations instead of time variant equilibriums is an efficient way to model vehicle kinematics. Lagrange methodology is the one to approximate to actual performance [16]. General Lagrangian equation is given in Eq. (3.43).

$$L = T - U, \qquad U = 0 \tag{3.43}$$

where T is the kinetic energy of the agent, and U is potential energy, which is permanent and can be assumed zero since the planar motion is executed. Thus, Lagrange formulation is deployed to predict DeLRo's dynamic behavior as presented in Eq. (3.44).

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right)^{T} - \left(\frac{\partial L}{\partial q}\right)^{T} = u \tag{3.44}$$

In the given formula, u represents external forces. Overall kinetic energy of the bot is given in Eq. (3.45). Where T_G and T_O represent vehicle's overall and rotational kinetic energy.

$$T = T_G + T_O \tag{3.45}$$

$$T = \frac{1}{2} W_G v^2 + \frac{1}{2}\, I_O \Omega^{\,2} \tag{3.46}$$

where W_G is DeLRo's gross weight, v is the linear velocity of the robot, Ω denotes spin rate, I_O is moment of inertia of the entire body and ω_i is wheel angular speed.

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}}\right) - \left(\frac{\partial L}{\partial q}\right) = \left[\frac{d}{dt}\left(\frac{\partial T_G}{\partial \dot{q}}\right) + \frac{d}{dt}\left(\frac{\partial T_O}{\partial \dot{q}}\right)\right] - \left[\frac{\partial T_G}{\partial q} + \frac{\partial T_O}{\partial q}\right] \tag{3.47}$$

$$\frac{d}{dt}\left(\frac{\partial T_G}{\partial \dot{q}}\right) = \frac{d}{dt}\left(\frac{\partial}{\partial \dot{q}}\left(\frac{1}{2} W_G\left(\dot{X}^2 + \dot{Y}^2\right)\right)\right) = W_G \ddot{X} + W_G \ddot{Y} \tag{3.48}$$

$$\frac{d}{dt}\left(\frac{\partial T_O}{\partial \dot{q}}\right) = \frac{d}{dt}\left(\frac{\partial}{\partial \dot{q}}\left(\frac{1}{2}\, I_O \dot{\varphi}^2\right)\right) = I_O \ddot{\varphi} \tag{3.49}$$

$$\frac{\partial T_G}{\partial q} = \frac{\partial}{\partial q}\left(\frac{1}{2} W_G\left(\dot{X}^2 + \dot{Y}^2\right)\right) = 0 \tag{3.50}$$

$$\frac{\partial T_O}{\partial q} = \frac{\partial}{\partial q}\left(\frac{1}{2}\, I_O \dot{\varphi}^2\right) = 0 \tag{3.51}$$

Therefore, overall Lagrangian equilibrium results in Eq. (3.52).

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}}\right) - \left(\frac{\partial L}{\partial q}\right) = W_G \ddot{X} + W_G \ddot{Y} + I_O \ddot{\varphi} \tag{3.52}$$

$$\begin{bmatrix} W_G & 0 & 0 \\ 0 & W_G & 0 \\ 0 & 0 & I_O \end{bmatrix} \begin{bmatrix} \ddot{X} \\ \ddot{Y} \\ \ddot{\varphi} \end{bmatrix} - 0 \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\varphi} \end{bmatrix} = u \tag{3.53}$$

Considering non-holonomic constraints which is mentioned in Eq. (3.39), dynamic model can be represented in terms of $M(q)$ mass-inertia, $V(q,\dot{q})$ Coriolis and centrifugal forces, $F(\dot{q})$ friction forces, $G(q)$ gravity forces, $E(q)$ torque transformation, and $A(q)$ constraint coefficient matrices [17].

$$M(q)\ddot{q} + V(q,\dot{q}) + F(\dot{q}) + G(q) = E(q)\tau - A^T(q)\lambda \tag{3.54}$$

Hence, taking second derivative of pose vector (q) which is defined in Eq. (3.8), the following is derived.

$$\ddot{q} = \dot{S}(q)\eta + S(q)\dot{\eta} \tag{3.55}$$

$$M(q)\left(\dot{S}(q)\eta + S(q)\dot{\eta}\right) + V(q,\dot{q}) + F(\dot{q}) + G(q) = E(q)\tau - A^T(q)\lambda \tag{3.56}$$

By multiplying $S(q)$ with both sides of the Eq. (3.56), we can simplify the equation.

$$S^T(q)\left[M(q)\left(\dot{S}(q)\eta + S(q)\dot{\eta}\right) + V(q,\dot{q}) + F(\dot{q}) + G(q)\right] = S^T(q)[E(q)\tau - A^T(q)\lambda] \tag{3.57}$$

$$M(q)\left(S^T(q)\dot{S}(q)\eta + S^T(q)S(q)\dot{\eta}\right) + S^T(q)V(q,\dot{q}) + S^T(q)F(\dot{q}) + S^T(q)G(q) = S^T(q)E(q)\tau - S^T(q)A^T(q)\lambda \tag{3.58}$$

It can be assumed that $S^T(q)A^T(q)\lambda = 0$ while $d_{ox} = 0$ since the agent is symmetrically designed (see Fig. 2.7), $G(q) = 0$ for planar motion on 0 degree slope ground, $V(q,\dot{q})$ is neglected, and $S^T(q)\dot{S}(q) = 0$ [18]. Hence, the simplified form of Eq. (3.58) can be derived as follows.

$$M(q)S^T(q)S(q)\dot{\eta} + S^T(q)F(\dot{q}) = S^T(q)E(q)\tau \tag{3.59}$$

Through the dynamic equations in Eq. (2.8-2.12), we can complete Lagrange formula.

$$M_r = I_O\dot{\Omega} = \frac{C}{2}R_{xR}\cos\varphi - \frac{C}{2}R_{xL}\cos\varphi + \frac{D}{2}F_{y_f}\cos\varphi - \frac{D}{2}F_{y_r}\cos\varphi \tag{3.60}$$

$$W_{\mathrm{G}}\mathrm{a}_x = F_x\cos\varphi - \mu_{ss}\left(R\cos\varphi - F_y\sin\varphi\right) \tag{3.61}$$

$$W_{\mathrm{G}}\mathrm{a}_y = F_y\sin\varphi - \mu_{ss}\left(R\sin\varphi + F_y\cos\varphi\right) \tag{3.62}$$

$$\begin{bmatrix} W_G & 0 & 0 \\ 0 & W_G & 0 \\ 0 & 0 & I_O \end{bmatrix}\begin{bmatrix} \mathrm{a}_x \\ \mathrm{a}_y \\ \dot{\Omega} \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{1}{r} \\ 0 & 0 \\ \frac{-C}{2r} & \frac{C}{2r} \end{bmatrix}\begin{bmatrix} \tau_L \\ \tau_R \end{bmatrix} - \mu_{ss}\begin{bmatrix} R_x \\ 0 \\ 0 \end{bmatrix} \tag{3.63}$$

$$S^T(q) = \begin{bmatrix} cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.64}$$

$$\mathbf{M} = S^T(q)M(q)S(q)\dot{\eta} \tag{3.65}$$

$$\mathbf{F} = S^T(q)F(\dot{q}) \tag{3.66}$$

$$\mathbf{E} = S^T(q)E(q) \tag{3.67}$$

Thus, by substituting $S^T(q)$ and $S(q)$, we can derive $\mathbf{M}$ in Eq. (3.68), and in terms of Eq. (2.10) and Eq. (2.26), the following equilibriums can be written.

$$\widetilde{\mathbf{M}} + \tilde{\mathbf{F}} = \tilde{\mathbf{E}}\boldsymbol{\tau} \tag{3.68}$$

$$\widetilde{\mathbf{M}} = \begin{bmatrix} W_G & 0 & 0 \\ 0 & W_G & 0 \\ 0 & 0 & I_O \end{bmatrix}\begin{bmatrix} cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{\Omega} \end{bmatrix} \tag{3.69}$$

$$\widetilde{\mathbf{M}} = \begin{bmatrix} W_G & 0 & 0 \\ 0 & W_G & 0 \\ 0 & 0 & I_O \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{\Omega} \end{bmatrix} = \begin{bmatrix} W_G & W_G & I_O \end{bmatrix}\begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{\Omega} \end{bmatrix} \tag{3.70}$$

$$\widetilde{\mathbf{M}} = \begin{bmatrix} W_G & W_G & I_O \end{bmatrix}\begin{bmatrix} \dot{v}_x \\ \dot{v}_y \\ \dot{\Omega} \end{bmatrix} = \frac{1}{r}\begin{bmatrix} \cos\varphi & \cos\varphi & \cos\varphi & \cos\varphi \\ \sin\varphi & \sin\varphi & \sin\varphi & \sin\varphi \\ -\frac{C}{2} & -\frac{C}{2} & \frac{C}{2} & \frac{C}{2} \end{bmatrix}\begin{bmatrix} \tau_{x1} \\ \tau_{x2} \\ \tau_{x3} \\ \tau_{x4} \end{bmatrix} - \mu_{ss}\begin{bmatrix} R_x \\ R_y \\ R_\Omega \end{bmatrix} \tag{3.71}$$

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\varphi} \\ \dot{v_x} \\ \dot{\omega_1} \\ \dot{\omega_2} \\ \dot{\omega_3} \\ \dot{\omega_4} \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \Omega \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{W_G r} & \frac{1}{W_G r} & \frac{1}{W_G r} & \frac{1}{W_G r} \\ -\frac{C}{4I_O r} & -\frac{C}{4I_O r} & \frac{C}{4I_O r} & \frac{C}{4I_O r} \\ -\frac{C}{4I_O r} & -\frac{C}{4I_O r} & \frac{C}{4I_O r} & \frac{C}{4I_O r} \\ -\frac{C}{4I_O r} & -\frac{C}{4I_O r} & \frac{C}{4I_O r} & \frac{C}{4I_O r} \\ -\frac{C}{4I_O r} & -\frac{C}{4I_O r} & \frac{C}{4I_O r} & \frac{C}{4I_O r} \end{bmatrix} \begin{bmatrix} \tau_{x1} \\ \tau_{x2} \\ \tau_{x3} \\ \tau_{x4} \end{bmatrix} \quad (3.72)$$

Chapter 4: DeLRo's Control System

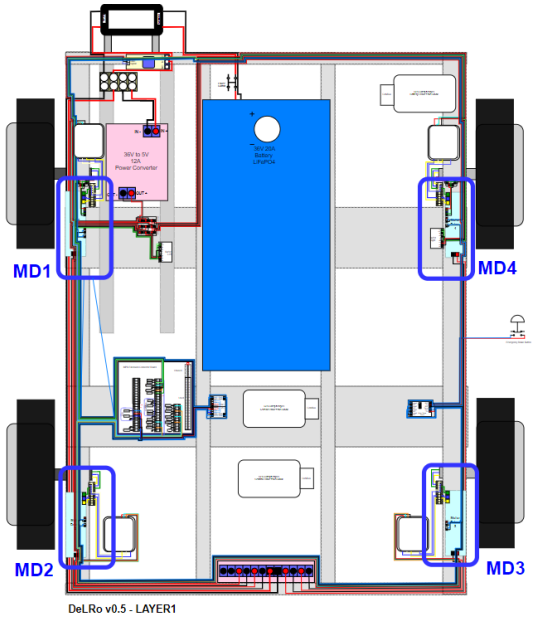


Figure 4.1 DeLRov0.5 interior layout (MDx: Motor driver)

4.1 State Space Modeling

Mathematical modeling is key point to design and analyze a control system that enables to predict system behaviors.

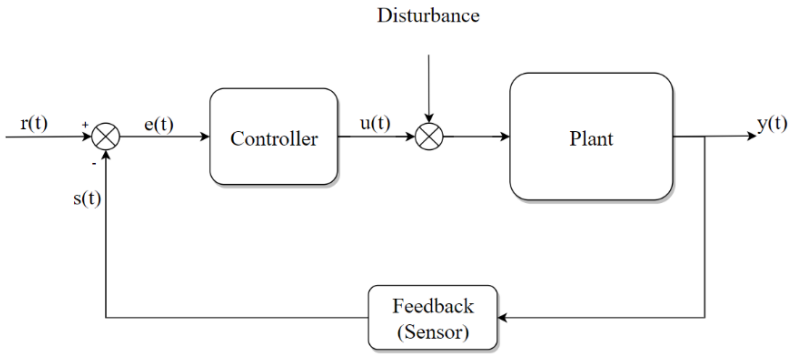


Figure 4.2 Typical closed-loop control system

4.1.1 Motor Control Modeling

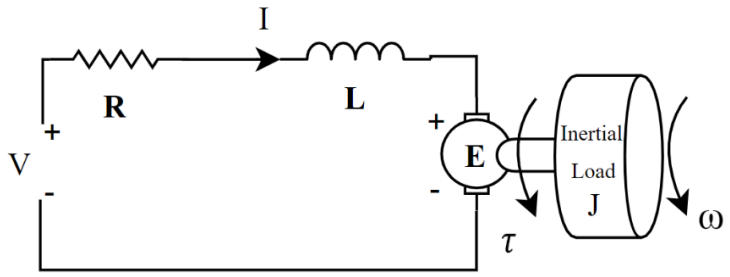


Figure 4.3 DC Motor electrical circuit

A general DC motor circuit is illustrated in Figure 4.3. As our brushless motor mechanism does not have any gearbox or external shafting components, it is assumed that the load is directly driven by motor stator magnetic interaction. The torque is transmitted to the load with minimum loss which is proportional to friction coefficient (B). Kirchhoff voltage equation is given in Eq. (4.1).

$$V(t) = I(t)R + L\frac{dI(t)}{dt} + E(t) \tag{4.1}$$

DeLRo's proposed wheels are BLDC hub (in-wheel) motors which are described with details in Chapter 2. In order to design this system, it is required to analyze the mathematical model of electromechanics with bottom-up approach. Therefore, motor modeling is the base for the entire agent.

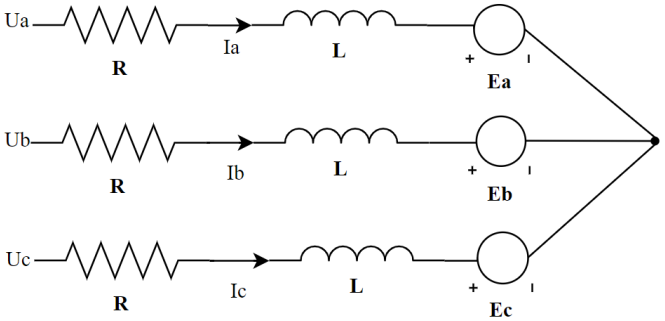


Figure 4.4 Simplified BLDC hub motor circuit

A brushless DC motor consists of three phases which are represented with components of motor phase voltages, Ua, Ub, Uc, phase currents, Ia, Ib, Ic, resistance, R, inductance, L, and back EMF, E, in Figure 4.4. Thus, the motor phase voltage equation is derived in Eq. (4.2).

$$\begin{bmatrix} U_a(t) \\ U_b(t) \\ U_c(t) \end{bmatrix} = \begin{bmatrix} R & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & R \end{bmatrix} \begin{bmatrix} I_a(t) \\ I_b(t) \\ I_c(t) \end{bmatrix} + \begin{bmatrix} L & 0 & 0 \\ 0 & L & 0 \\ 0 & 0 & L \end{bmatrix} \frac{d}{dt} \begin{bmatrix} I_a(t) \\ I_b(t) \\ I_c(t) \end{bmatrix} + \begin{bmatrix} E_a(t) \\ E_b(t) \\ E_c(t) \end{bmatrix} \quad (4.2)$$

$$E(t) = K_e \omega(t) \quad (4.3)$$

In equilibrium (4.3) K_e denotes back EMF constant and $\omega(t)$ represents rotor speed. And the motor generates a torque which is the total electromechanical torque [lb.in] is given by Eq. (4.4).

$$\tau_e(t) = \frac{1}{\omega(t)} [E_a(t) I_a(t) + E_b(t) I_b(t) + E_c(t) I_c(t)] \quad (4.4)$$

The torque can also be described as in Eq. (4.5).

$$\tau_e(t) = J \frac{d\omega(t)}{dt} + B\omega(t) + \tau_l(t) \quad (4.5)$$

where J, B, and τ_l are motor inertia, friction coefficient, and load torque, respectively. Motor torque constant (K_t) is given with the denominator $I(t)$ armature current in following equation.

$$K_t = \frac{\tau_e(t)}{I(t)} \left[\frac{lb.in}{A}\right] \quad (4.6)$$

From Eq. (4.6) $I(t)$ is derived in Eq. (4.7).

$$I(t) = \frac{\tau_e(t)}{K_t} = \frac{J d\omega(t)}{K_t dt} + \frac{B\omega(t)}{K_t} + \frac{\tau_l(t)}{K_t} \quad (4.7)$$

$V(t)$ is obtained in Eq. (4.8) by substituting Eq. (4.7) and Eq. (4.3) in Eq. (4.1). Hence,

$$V(t) = \left[\frac{J d\omega(t)}{K_t dt} + \frac{B\omega(t)}{K_t} + \frac{\tau_l(t)}{K_t}\right] R + L \frac{d}{dt} \left[\frac{J d\omega(t)}{K_t dt} + \frac{B\omega(t)}{K_t} + \frac{\tau_l(t)}{K_t}\right] + E(t) \quad (4.8)$$

$$= R\frac{Jd\omega(t)}{K_t dt} + RB\frac{\omega(t)}{K_t} + R\frac{\tau_l(t)}{K_t} + L\frac{Jd^2\omega(t)}{K_t dt^2} + LB\frac{d\omega(t)}{K_t dt} + L\frac{d\tau_l(t)}{K_t dt} + \frac{R}{K_t}\tau_l(t)$$

$$= \frac{LJ}{K_t}\frac{d^2\omega(t)}{dt^2} + \frac{(RJ+LB)}{K_t}\frac{d\omega(t)}{dt} + \frac{(RB+K_eK_t)}{K_t}\omega(t) + \frac{L}{K_t}\frac{d\tau_l(t)}{dt} + \frac{R}{K_t}\tau_l(t) \quad (4.9)$$

Assuming the model is not associated with any load, $\tau_l(t)$ is taken zero value in Eq. (4.9).

$$V(t) = \frac{LJ}{K_t}\frac{d^2\omega(t)}{dt^2} + \frac{(RJ+LB)}{K_t}\frac{d\omega(t)}{dt} + \frac{(RB+K_eK_t)}{K_t}\omega(t) \quad (4.10)$$

As we obtain a linear model of BLDC motor, in order to accumulate the transfer function of the system Laplace transform can be utilized in Eq. (4.12).

$$V(s) = \int_0^\infty V(t)e^{-st}dt \quad (4.11)$$

$$\frac{V(s)}{\omega(s)} = \frac{LJ}{K_t}s^2 + \frac{(RJ+LB)}{K_t}s + \frac{(RB+K_eK_t)}{K_t} \quad (4.12)$$

$$T(s) = \frac{\omega(s)}{V(s)} = \frac{K_t}{LJs^2 + (RJ+LB)s + (RB+K_eK_t)} \quad (4.13)$$

Multiplying $(\frac{1}{LJ})$ by Eq. (4.13), general transfer function expression is derived in Eq. (4.14).

$$T(s) = \frac{\omega(s)}{V(s)} = \frac{K_t/LJ}{s^2 + \frac{(RJ+LB)}{LJ}s + \frac{(RB+K_eK_t)}{LJ}} \quad (4.14)$$

In this case, there is few assumptions considered to have a simplified equation. Friction coefficient B is approximated to zero ($B \cong 0$), $RJ \gg LB$ and $K_eK_t \gg RB$. Therefore, reorganizing Eq. (4.14) we get simplified transfer function of BLDC Motor circuit in Eq. (4.15).

$$T(s) = \frac{\omega(s)}{V(s)} = \frac{\frac{K_t}{LJ}}{s^2 + \frac{R}{L}s + \frac{K_eK_t}{LJ}} \quad (4.15)$$

State-space representation of this equation can be obtained by state variables $\omega(t)\ and\ \dot{\omega}(t)$ in following statement.

$$\begin{bmatrix}\dot{\omega}(t)\\ \ddot{\omega}(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\frac{K_e K_t}{LJ} & -\frac{R}{L}\end{bmatrix}\begin{bmatrix}\omega(t)\\ \dot{\omega}(t)\end{bmatrix} + \begin{bmatrix}0\\ \frac{K_t}{LJ}\end{bmatrix}V(t) \tag{4.16}$$

$$y(t) = [1 \quad 0]\begin{bmatrix}\omega(t)\\ \dot{\omega}(t)\end{bmatrix} \tag{4.17}$$

As described in Eq. (4.3) and Eq. (4.6), motor torque constant, K_t, and motor voltage constant, K_e, are calculated through the measurement tests as conducted in [19].

$$K_e = \frac{\mathrm{E}}{\omega_{test}}\left[\frac{Volts}{rev/min}\right] \tag{4.18}$$

Table 4.1 BLDC motor constant measurements

Parameters	Value	Unit
$\mathrm{V}_{phase} = V_{l-l}/\sqrt{3}$	9.23	$[Volts]$
I (no load)	0.12	$[A]$
R	0.32	$[Ohms]$
$\mathrm{E} = \mathrm{V}_{phase} - \mathrm{IR}$	9.19	$[Volts]$
ω_{test}	500	$[rev/min]$
L	0.586	$[mH]$
J	5.95E-05	$[lbf.in.s^2]$
$\mathrm{K_e} = \mathrm{E}/\omega_{test}$	0.018	$\left[\frac{Volts}{rev/min}\right]$
$\mathrm{K_t} = \frac{\tau_e}{\mathrm{I}} = \mathrm{K_e}\sqrt{3}/(0.0118)$	2.64	$\left[\frac{lb.in}{A}\right]$

Parameters are calculated as shown in Table 4.1. Hence, Eq. (4.15) can be rewritten in Eq. (4.19).

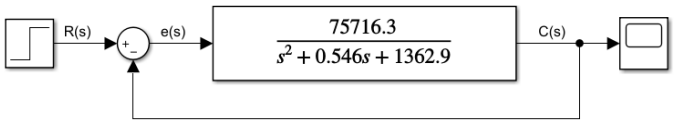
$$T(s) = \frac{\omega(s)}{V(s)} = \frac{\frac{2.64}{0.586\ 5.95\ 10^{-5}}}{s^2 + \frac{0.32}{0.586}s + \frac{0.018\ 2.64}{0.586\ 5.95\ 10^{-5}}} \tag{4.19}$$

$$T(s) = \frac{\omega(s)}{V(s)} = \frac{75716.3}{s^2 + 0.546s + 1362.9} \tag{4.20}$$

Thus, state-space model of the motor is also reorganized in Eq. (4.21) and Eq. (4.22).

$$\begin{bmatrix}\dot{\omega}(t)\\ \ddot{\omega}(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ -1362.9 & -0.546\end{bmatrix}\begin{bmatrix}\omega(t)\\ \dot{\omega}(t)\end{bmatrix} + \begin{bmatrix}0\\ 75716.3\end{bmatrix}V(t) \tag{4.21}$$

$$y(t) = [1 \quad 0]\begin{bmatrix}\omega(t)\\ \dot{\omega}(t)\end{bmatrix} \tag{4.22}$$



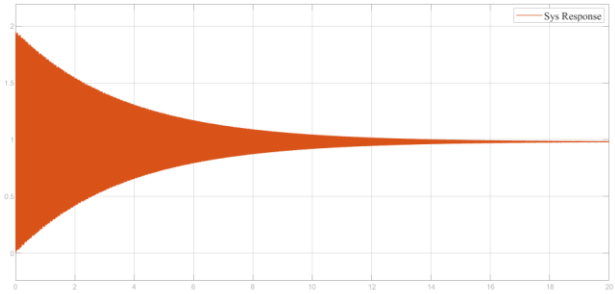


Figure 4.5 Simple closed-loop plant response

If we consider this closed-loop system, it is essential to study transient and steady-state response. Thus, transient state parameters which are represented by percent overshoot (P.O.), settling time (t_s), peak time (t_p), and rise time (t_r) enable to verify the approximations on circuit performance. As Figure 4.5 illustrates, the oscillation lasts until that steady-state value is reached and this is known as "Underdamped case".

$$G(s) = \frac{C(s)}{R(s)} = k\frac{{\omega_n}^2}{s^2 + 2\zeta\omega_n + {\omega_n}^2} \tag{4.23}$$

In Eq. (4.23) k is gain, ω_n is natural (oscillation) frequency of the system, and ζ is damping ratio.

$$C(s) = e(s)T(s) = [R(s) - C(s)]T(s) \tag{4.24}$$

$$G(s) = \frac{T(s)}{1 + T(s)} \tag{4.25}$$

If we substitute the motor transfer function in Eq. (4.25) and apply step input function (R(s)=1/s) as depicted in Fig. 4.5 we get second order system in Eq. (4.26).

$$C(s) = \frac{75716.3}{s(s^2 + 0.546s + 77092.2)} \tag{4.26}$$

$$C(s) = 1.018\frac{77092.2}{s(s^2 + 0.546s + 77092.2)} \tag{4.27}$$

Thus, oscillation frequency $\omega_n = \sqrt{77092.2} = 277.65\ [Hz]$, damping ratio $\zeta = 9.8\text{E} - 4$, and gain $k = 1.018$ are obtained in Eq. (4.27). In this case, transient specifications can be derived.

$$P.O. = 100\text{e}^{\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right)} \tag{4.28}$$

$$t_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} \tag{4.29}$$

$$t_r = \frac{\pi - \tan^{-1}\left(\sqrt{1-\zeta^2}/\zeta\right)}{\omega_n\sqrt{1-\zeta^2}} \tag{4.30}$$

$$t_s \approx \frac{4}{\zeta\omega_n}, \text{for 2\% if the damping ratio } \zeta \ll 1 \tag{4.31}$$

As a result, signal refinement is necessary in terms of tuning algorithms. We utilize Proportional-Integral-Derivative (PID) controller to minimize transient period in BLDC motor driving.

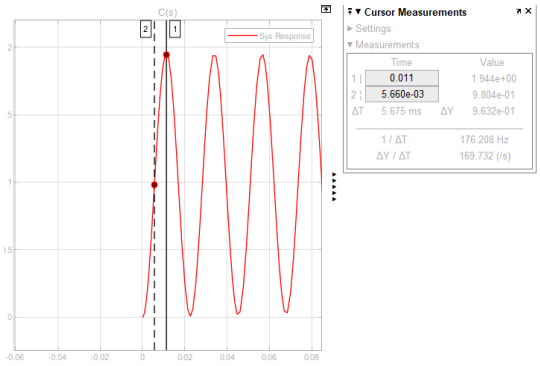


Figure 4.6 Underdamped case. Settling time $(t_s) = 14.7$ [s], peak time$(t_p) = 0.0113$ [s], rise time $(t_r) = 0.00566$ [s], percent overshoot $(\text{P. O.}) = 99.7\%$.

4.1.2 PID Control

Speed controlling is critical to maintain stable performance with a quick steady-state response. Hence, there are many tuning methods have been studied in past, such as Ziegler-Nichols [20], ITAE [21], Fuzzy PID [22], Neural Network [23], and so on. It is aimed to utilize Ziegler-Nichols in this part of the project.

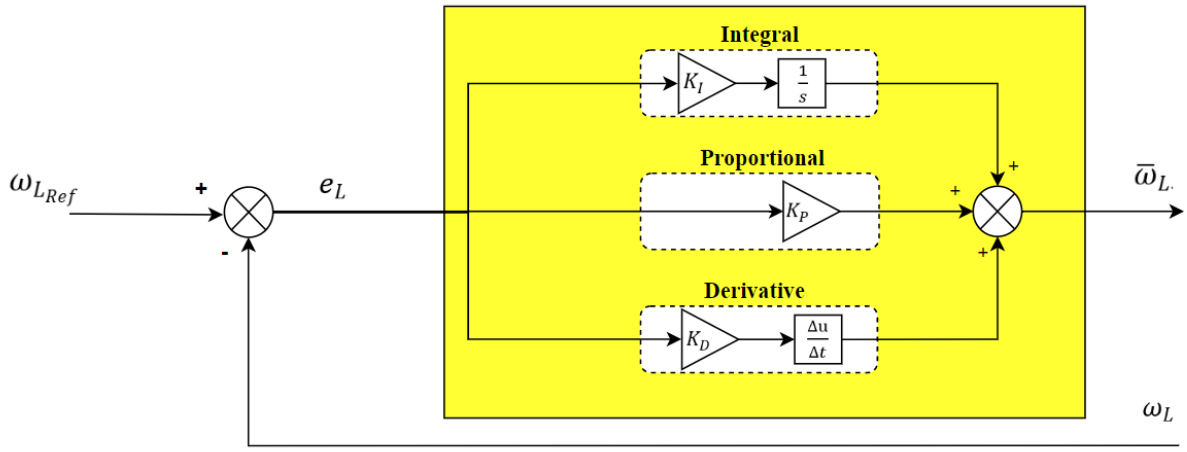


Figure 4.7 General PID controller block scheme

$$\bar{\omega}_L(t) = K_P e_L(t) + K_I \int_0^t e_L(\tau)\, d\tau + K_D \frac{de_L(t)}{dt} \tag{4.32}$$

$$\bar{\omega}_L(s) = K_P \left[E_L(s) + \frac{K_I}{K_P s} E_L(s) + \frac{K_D}{K_P} s E_L(s) \right] \tag{4.33}$$

$$C(s) = \frac{\bar{\omega}_L(s)}{E_L(s)} = K_P \left[1 + \frac{K_I}{K_P s} + \frac{K_D}{K_P} s \right] \tag{4.34}$$

In Eq. (4.32) $\bar{\omega}_L(t)$ is output signal of the controller which refines rotational speed of the left wheel set to the target level, $\omega_{L_{Ref}}(t)$. According to the performance requirements (Table 2.3), the wheels rotates at 63 [rev/min] which is set point of the agent's velocity. $e_L(t)$ denotes error signal between encoder output, ω_L, and desired angular speed to be minimized by the

controller. $K_P, K_I,$ and K_D represent proportional, integral, and derivative gains, respectively. Hence, the equation can be organized as in Eq. (4.35) giving parameters to adjust PID controller performance.

$$C(s) = \frac{\overline{\omega}_L(s)}{E_L(s)} = K_P \left[1 + \frac{1}{T_I s} + T_D\, s\right] \tag{4.35}$$

Table 4.2 Ziegler-Nichols (ZN) method of tuning

Control	K_P	T_I	T_D
P	$0.5K_u$	-	-
PI	$0.45K_u$	$T_u/1.2$	-
PID	$0.60K_u$	$T_u/2$	$T_u/8$

Basic ZN tuning procedure consists of four steps beginning with a simple Proportional (P) circuit.

1. Set T_I and T_D zero to find ultimate gain (K_u) and oscillation frequency ($1/T_u$).
2. Start adding to K_P value in a small scale until observing a constant oscillation on transient response.
3. Then $K_P = K_u$, and $T_u = \text{constant oscillation period } [s]$.
4. Finally, update the parameters (T_I and T_D) according to Table 4.2.

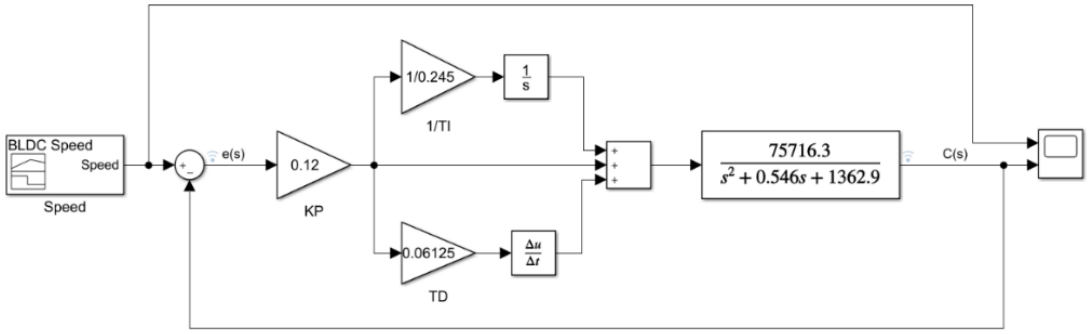


Figure 4.8 Single phase plant response: $K_u = 0.2,\ T_u = 0.49\ [s]$

In Figure 4.8 Simulink PID tuning model depicts a single-phase speed response of BLDC plant. Hence, it can be concluded that ZN method is approved on a simple closed-loop controller.

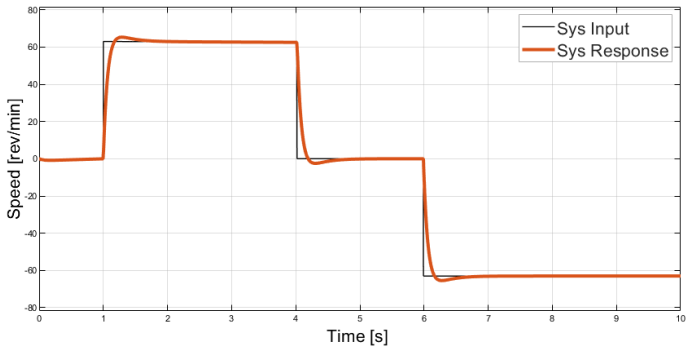


Figure 4.9 Simulated motor speed graph (MATLAB-Simulink©)

Considering Motor Processor Block (MPB), it is required to remember that there are many processes and delay time to simply drive the motors. Hence, the design of the complete circuit is as given in Figure 4.10. In accordance with test results, lag times are measured with respect to the process blocks (motor processor, DAC, 3-phase dc inverter, BLDC motor driver).

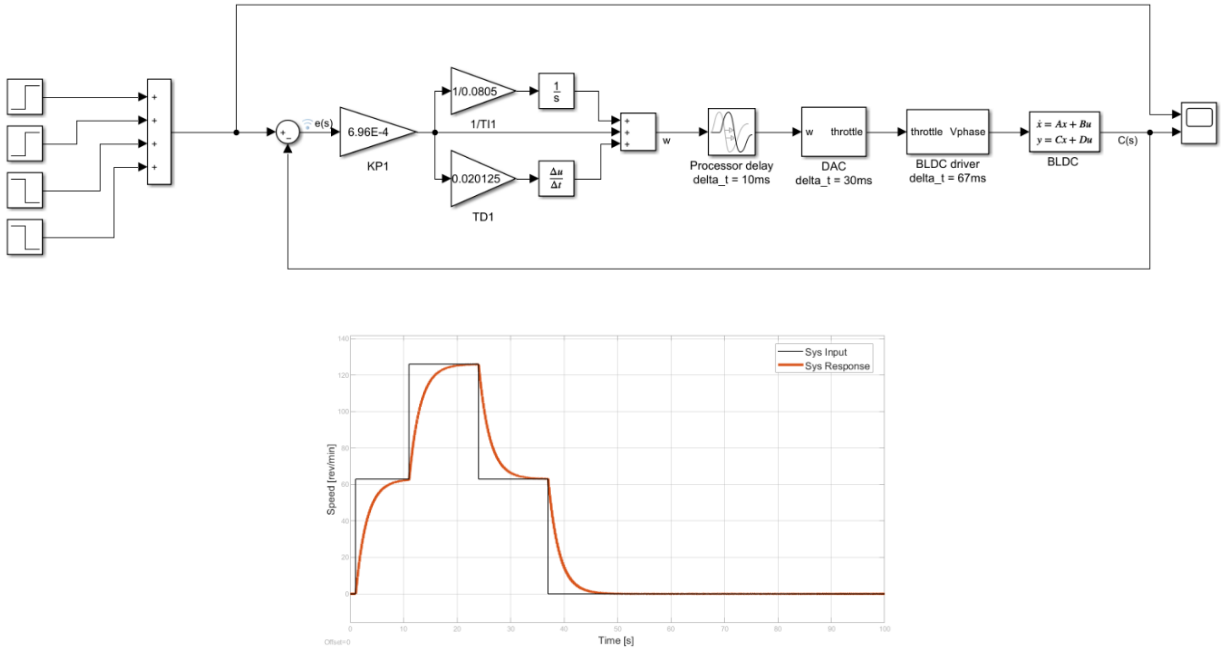


Figure 4.10 Complete design of the control block and estimated system response. $K_u = 0.00116,\ T_u = 0.17$ [s]

As shown in Fig. 4.10, transient response time is over 10 seconds.

4.2 Block Diagram of Operation

DeLRo has three main subsystems which are motor, central, and image processors units. Raspberry Pi3 and Nvidia Jetson TX2 programming boards are employed, and Controller Area Network (CANBus) communication protocol which allows to transmit data at 1[Mb/s] rate is established between the processors [24].

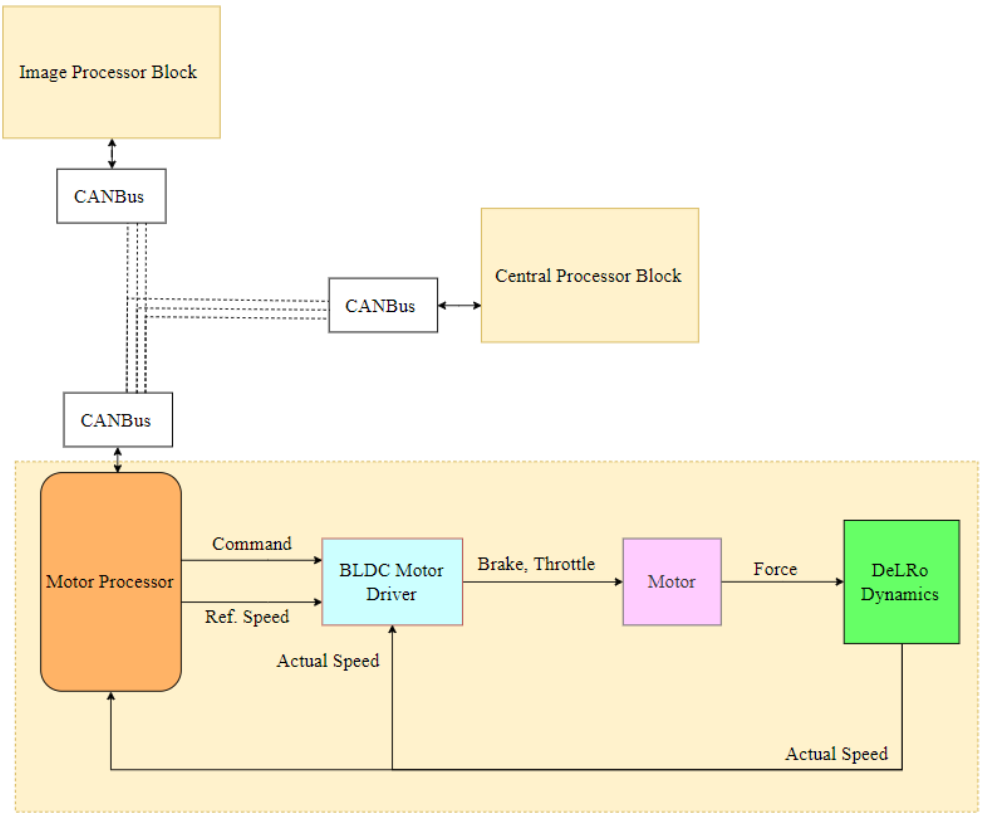


Figure 4.11 DeLRo's overall block diagram

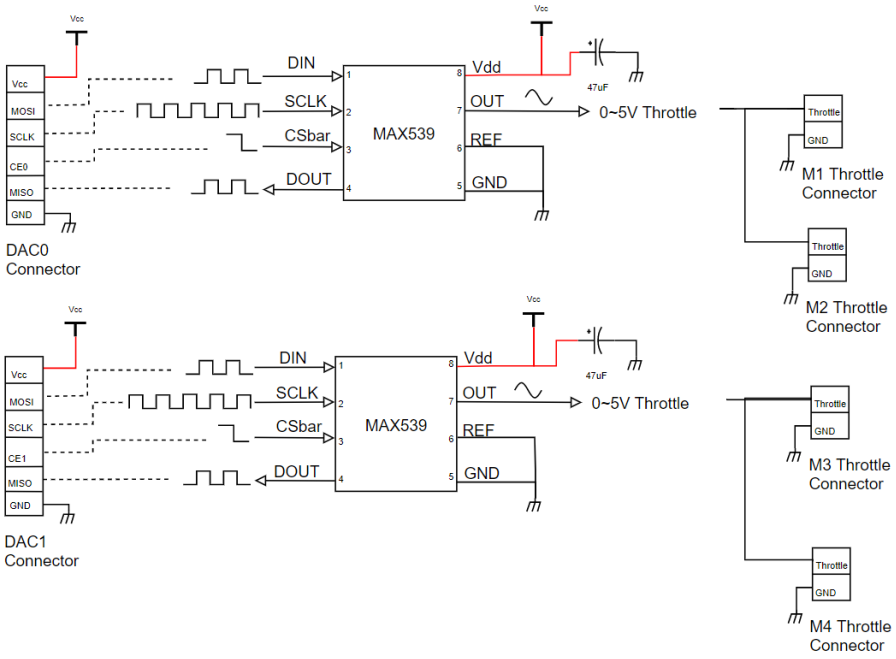


Figure 4.12 Throttle D/A converter circuit

As illustrated in Fig. 4.12, the motors are driven through analog voltage signal. Therefore, a digital to analog signal converter (DAC) MAX539 is deployed. Serial Peripheral Interface (SPI) protocol allows to transmit data from microprocessor to DAC. Processing lag time is measured as 30ms.

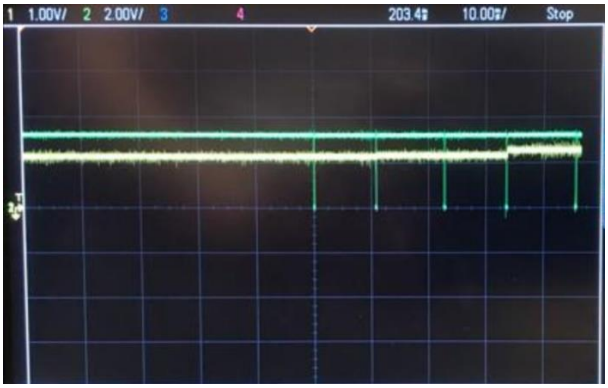


Figure 4.13 DAC response lag time.

In Fig. 4.13 each grid represents 1 [volt] for yellow line and 2 [volts] for SPI datain pin in vertical, and 10ms time resolution in horizontal. In this case, it is seen that 30ms time delay between input and output pin. Unipolar configuration is to get positive output.

$$V_{out} = +2(V_{dd})\left(\frac{datain}{2^{data\_length(datain)}}\right) \tag{4.36}$$

In order to rotate the wheels at 63 [rev/min], it is required to produce 1.350 [volts] in the output of the DAC output.

$$V_{out} = +2(V_{dd})\left(\frac{bin\_to\_dec(001000101001)}{2^{data\_length(001000101001)}}\right) \tag{4.37}$$

As Fig. 4.15 shows a basic circulatory path simulation is executed. Path diameter is around 3.20 $[ft]$, vehicle moves at 3.1 $[ft/s]$, and angular velocities of the wheels are 63 and 120 $[rev/min]$.

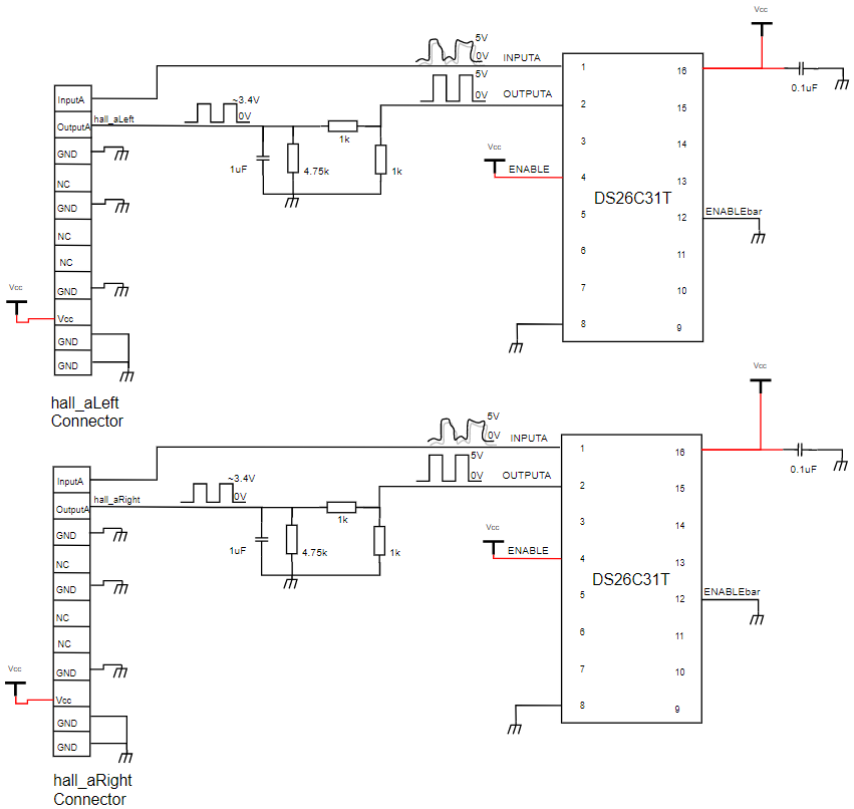


Figure 4.14 Hall-effect buffer amplifier circuit

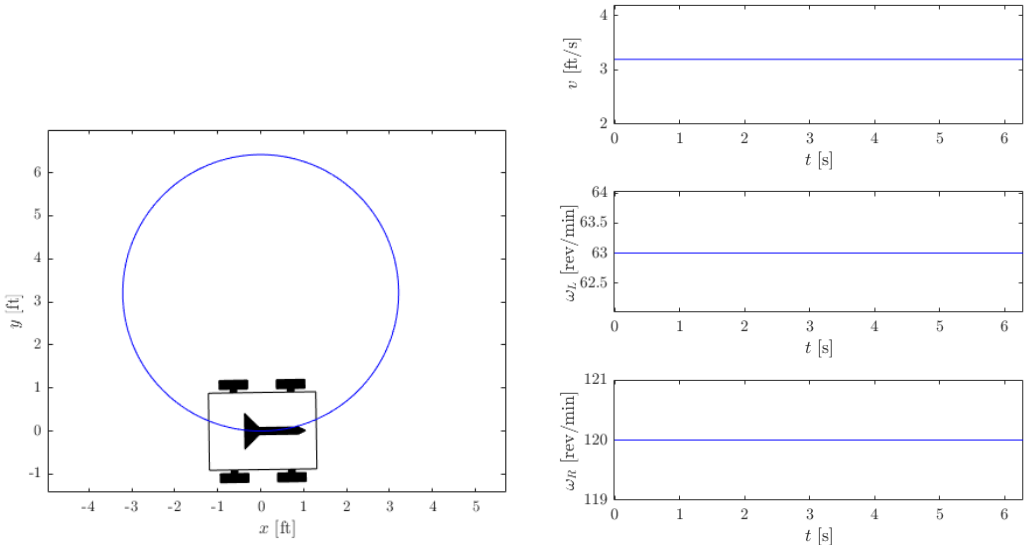


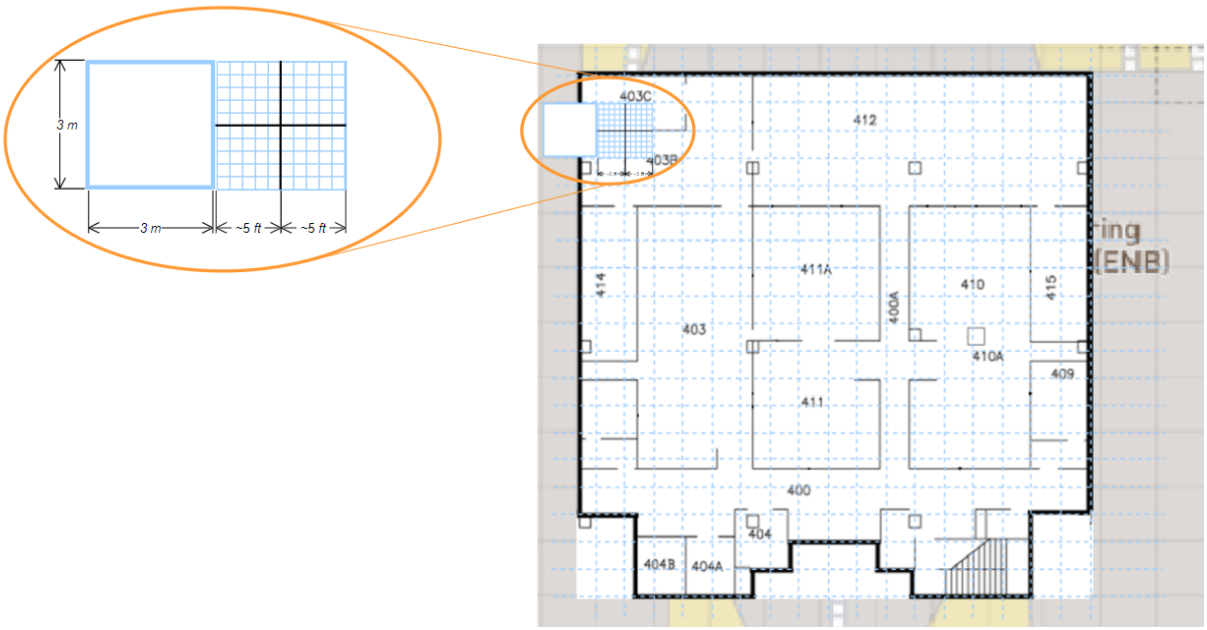
Figure 4.15 MATLAB-Simulink circle path simulation

Chapter 5: DeLRo Localization

Localization is one of the main challenges in mobile robotics. While outdoor tasks are commonly accomplished with GPS utilization, indoor positioning appears with challenges [25, 26]. Building structures have attenuative effects on microwave signals, and this requires additional equipment for an adequate signal transmission in the operational environments. Instead of wireless signal deployment, landmarks are created to compute the current location by placing on the walls.

Hence, it is aimed to acquire a deeper understanding in mobile indoor localization through Reinforcement Learning and Deep Neural Networks (DNN). In our scenario the robot possesses the floor layout in his database which consists of 10 sequential landmarks. Point-to-point navigation is studied autonomously by deploying avoiding collision, scene recognition, and environmental immediate mapping. After initialization of the agent, it proceeds through the halls and senses landmarks to verify its current flowing position.

Floor map is created as a 2-dimensional 95x90 array, and each value represents a $1\ ft^2$ area. As occupancy grid methodology is deployed, what3words grid location approach is the main reference that brings an innovative solution to remove the whole address determination chaos from human life [27]. Thanks to this three-word combination, it becomes more precise to address an instant point in general sense.



(a)

(b)

Figure 5.1 Projection of USF Engineering Building (ENB) (a) 4th floor on https://what3words.com, (b) Estimated trajectory in 4th floor of USF Engineering building.

Image processor block contains 2 camera units which are Mynt Eye 1030s stereo camera for proximity sensing, depth mapping and spatial coordinate tracking, and Leopard CSI camera for video processing.

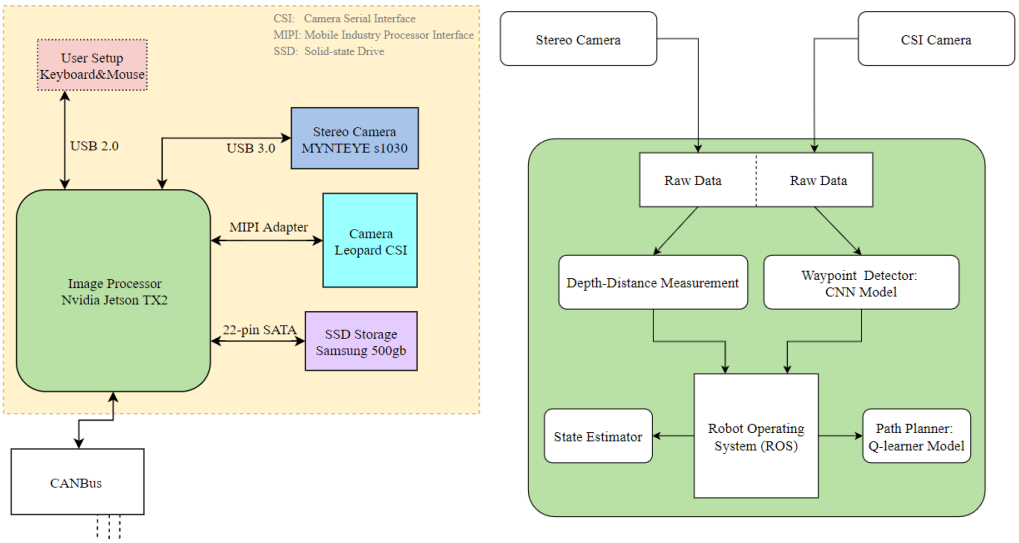


Figure 5.2 Image processor block components and system architecture

5.1 Indoor Localization

In our scenario robot has two floor (1st and 4th floors) layouts in database. After initialization of the robot, platform proceeds through the hall until sensing verification mark on the wall to compare current location with map in database. There are four waypoints of QR codes on the walls in 1st floor and ten waypoints in 4th floor which are recognizable by the robot. It has been designed that there are two cameras as sensor to complete a visionary based localization.

5.1.1 Hardware Setup

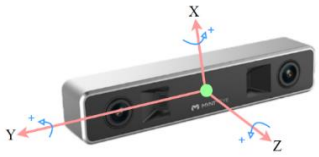


Figure 5.3 MYNT EYE S1030 stereo depth camera and IMU coordinate axis

Stereo camera has wider FOV compared the CSI camera. Stereo and depth resolutions are 752x480 at 60 fps and depth range is up to 18 meters. Six axis built-in IMU sensor allows to stream real time spatial location.

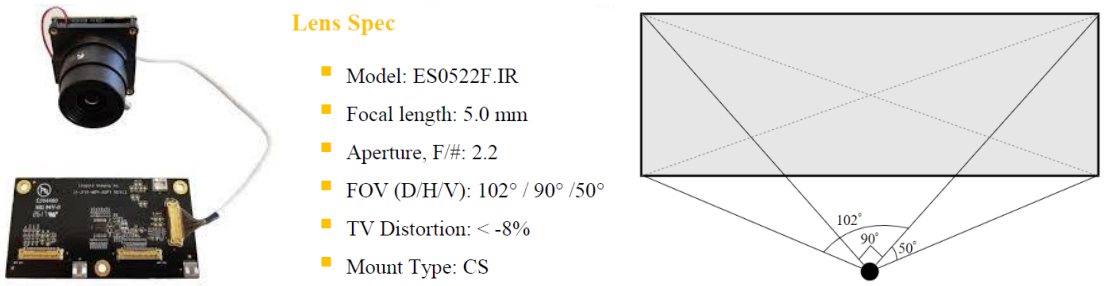


Figure 5.4 Leopard IMX185 CSI camera specifications

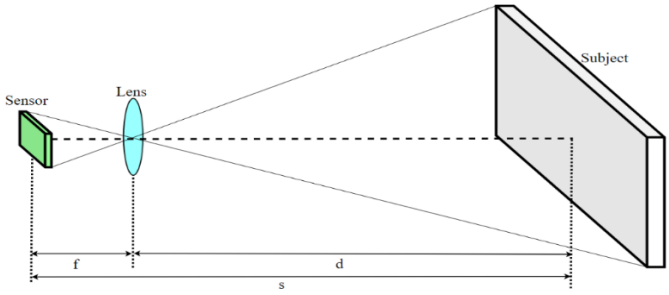


Figure 5.5 Geometrically distance projection between camera and object. *f*: focal length [mm], *d*: distance between lens and object [mm], *s*: distance between sensor and object [mm]

$$\frac{f}{d} = \frac{sensor\ dimension}{subject\ dimension} \tag{5.1}$$

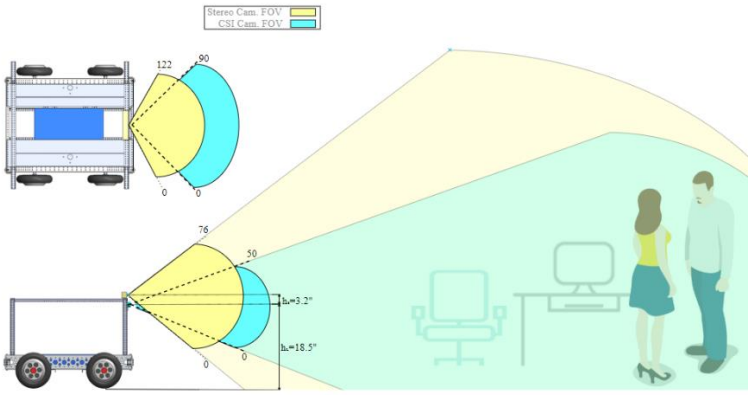


Figure 5.6 DeLRo's Field of View (FOV) representation.

Depth sensor and CSI camera are mounted on the nearest points to minimize nonsynchronous frames. Therefore, most of the cases, images are required to be overlapped from both sources. Depth camera is placed 3.2 inches above the CSI cam.

Figure 5.7 Screen shots of stereo and CSI cameras from DeLRo's perspective.

In order to perform the stereo camera, calibration is required by conducting in a few steps, which is proposed in Mynt Eye SDK installation guide [28] and practiced in a wider perspective by this study [29].

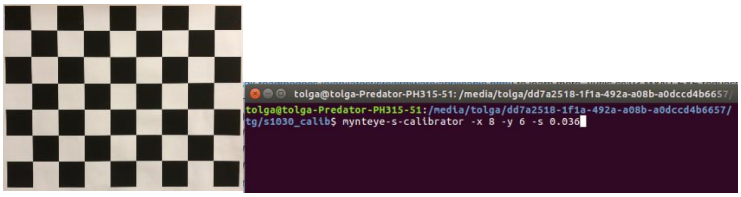


Figure 5.8 9x7x60mm chess board layout is used for calibration. In given prompt, the input format is: -x -*width* -y -*height* -s -*square in meter*.

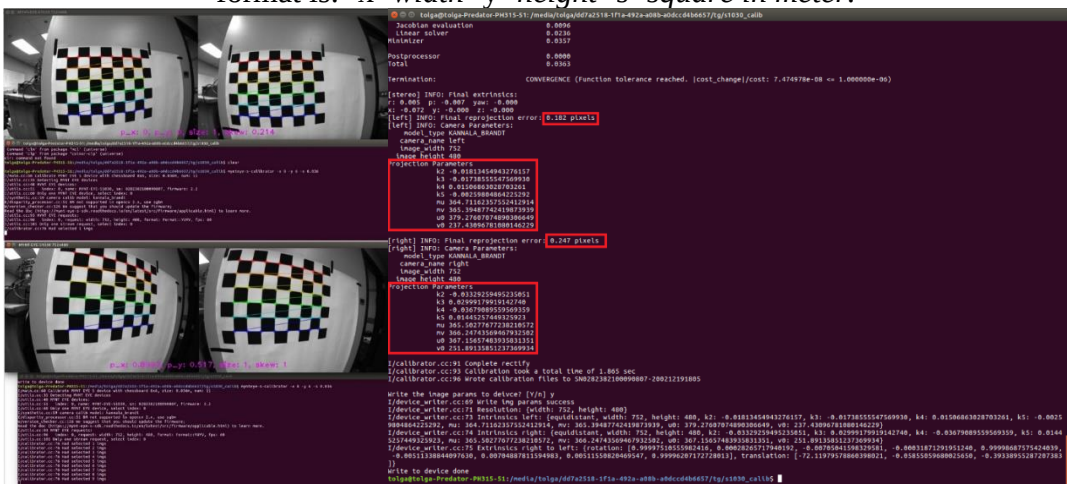


Figure 5.9 Reprojection errors and parameters.

Recommended reprojection error is under 1, and it is obtained 0.182 pixels for left and 0.247 pixels for right eyes that seems pretty good as illustrated in Fig. 5.9.

5.1.2. Object Detection

In this project, a deep learning method which Convolutional Neural Networks (CNN) is deployed to observe, detect, and evaluate the markers [30], [31]. Hence, various models and pretrained networks are tested. As there are 10 different waypoints to be classified which are shown in Figure 5.9, these custom patterns are created in a sophisticated way to observe how they are efficient and accurate instead of using such simplistic geometrical shapes.

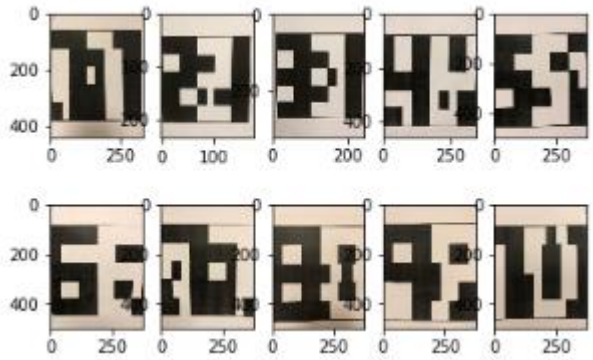


Figure 5.10 Custom indoor waypoints

Machine learning practices are usually run with multidimensional arrays which is the key element of CNN architecture is called "Kernel" or simply a matrix form. Kernel convolution which is also known as filter calculation is essential for image processing [32]. Filters also refer to 3D structures of multiple kernels stacked. For CNN it is used 2D kernels which is the same as dimensional property as filter. The convolution of two functions, $f(t) * g(t)$ in continuous time is described in Eq. (5.2).

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{5.2}$$

If $G(m,n)$ is a feature matrix, Eq. (5.4) in discrete time is formulated as follow.

$$(f * g)(m) = \sum_{j=-\infty}^{\infty} f(j)\, g(m-j) \tag{5.3}$$

$$G(m,n) = (f * g)(m,n) \tag{5.4}$$

$$G(m,n) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} f(j,k)\, g(m-j, n-k) \tag{5.5}$$

Cross-correlation 2D convolution form which is accepted by ML community for real-valued functions is shown in Eq. (5.6).

$$(f + g)(m,n) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} f(j,k)\, g(m+j, n+k) \tag{5.6}$$

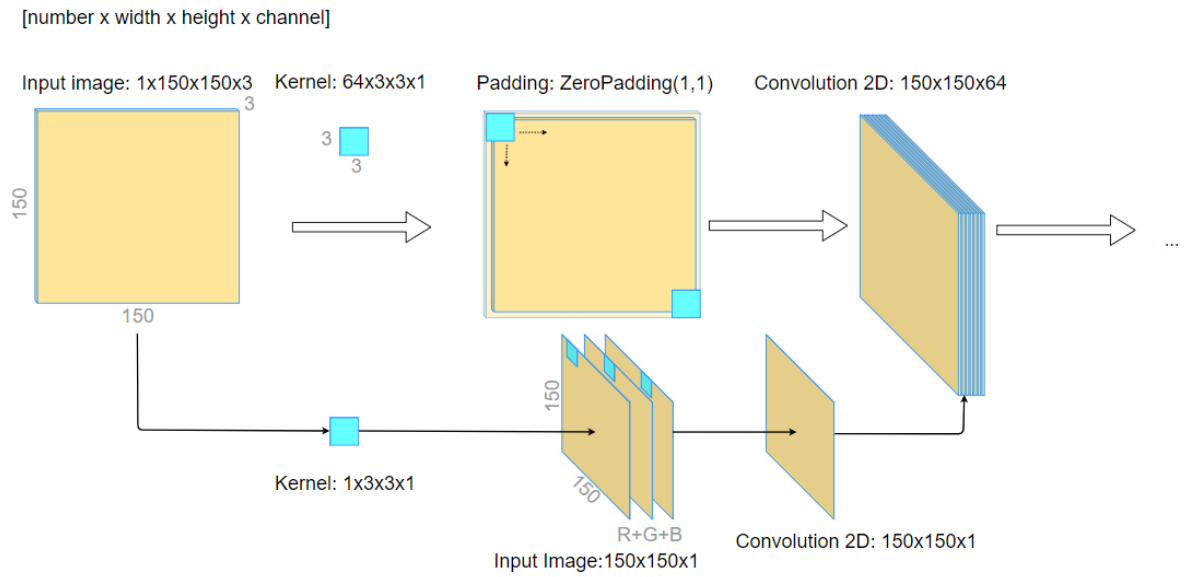


Figure 5.11 An illustration of essential 2-dimensional convolution on 3-channel input image.

Once the 3x3 filter is applied to image channels, then these layers are summed to a 150x150x1 single layer, and this process is returned for 64 kernels (150x150x64).

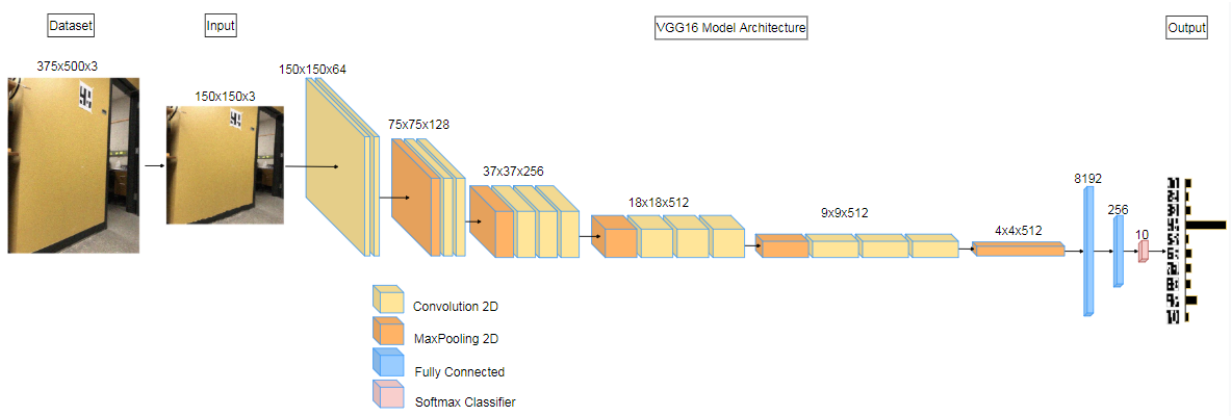


Figure. 5.12 VGG-16 pretrained model architecture.

Our dataset consists of 9991 training, 4994 validation, and 5000 test images in 10 classes. In order to avoid unnecessary effort on creating a new model from scratch which requires too much training and huge number of hyperparameters not a guaranteed way to succeed a reasonable accuracy [33], fine-tuning a pre-trained network is much more feasible.

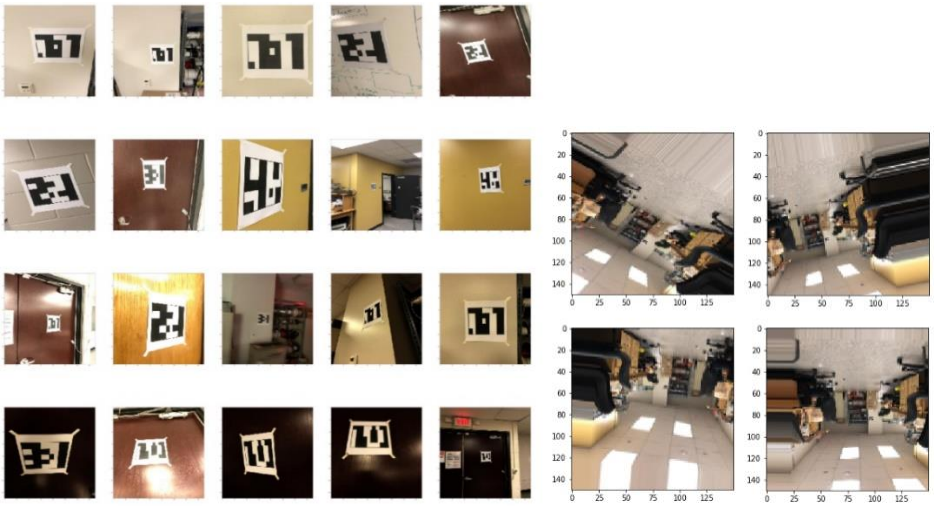


Figure 5.13 Examples from our custom dataset, and instances of an augmented image

In our CNN model, in order to observe the performance, the dataset is trained through VGG16 network without augmentation. As presented in Fig. 5.12, training accuracy reaches at 96%, and validation accuracy remains under 60%. This shows a typical overfitting case that can

be solved with several methods which are data augmenting, increasing the amount of image dataset, cross-validation, and adding dropout layer.

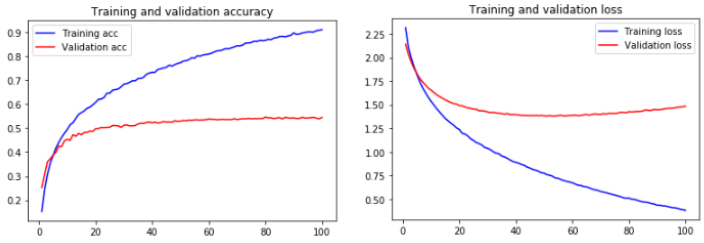


Figure 5.14 Accuracy and loss curves.

Considering we have less data, and patterns have common features, it is required to increase the data distribution for each instance. For instance, an image can be reproduced in various conditions such as lightning, angle, size, and morphology. At this point, this is achieved by utilizing “ImageDataGenerator” from Keras library [34]. Training and validation datasets are modified, and the result is illustrated in Fig. 5.14.

Figure 5.15 Improvement is remarkably visible, and variance is removed on the curves. It seems training accuracy falls to around 85% while validation accuracy reaches over 80% at 30 of epochs.

It is critical to fine-tune the higher layers due to overfitting concern. In this case, the last three layers are fine-tuned, and classifier is retrained.

```
Model: "vgg16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_4 (InputLayer)         (None, 150, 150, 3)       0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 150, 150, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 150, 150, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 75, 75, 64)        0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 75, 75, 128)       73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 75, 75, 128)       147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 37, 37, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 37, 37, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 37, 37, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 37, 37, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 18, 18, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 18, 18, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 18, 18, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 18, 18, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 9, 9, 512)         0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 9, 9, 512)         2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 9, 9, 512)         2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 9, 9, 512)         2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 4, 4, 512)         0
=================================================================
Total params: 14,714,688
Trainable params: 14,714,688
Non-trainable params: 0
_________________________________________________________________

Model: "sequential_7"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg16 (Model)                (None, 4, 4, 512)         14714688
_________________________________________________________________
flatten_1 (Flatten)          (None, 8192)              0
_________________________________________________________________
dense_11 (Dense)             (None, 256)               2097408
_________________________________________________________________
dense_12 (Dense)             (None, 10)                2570
=================================================================
Total params: 16,814,666
Trainable params: 2,099,978
Non-trainable params: 14,714,688
_________________________________________________________________
```

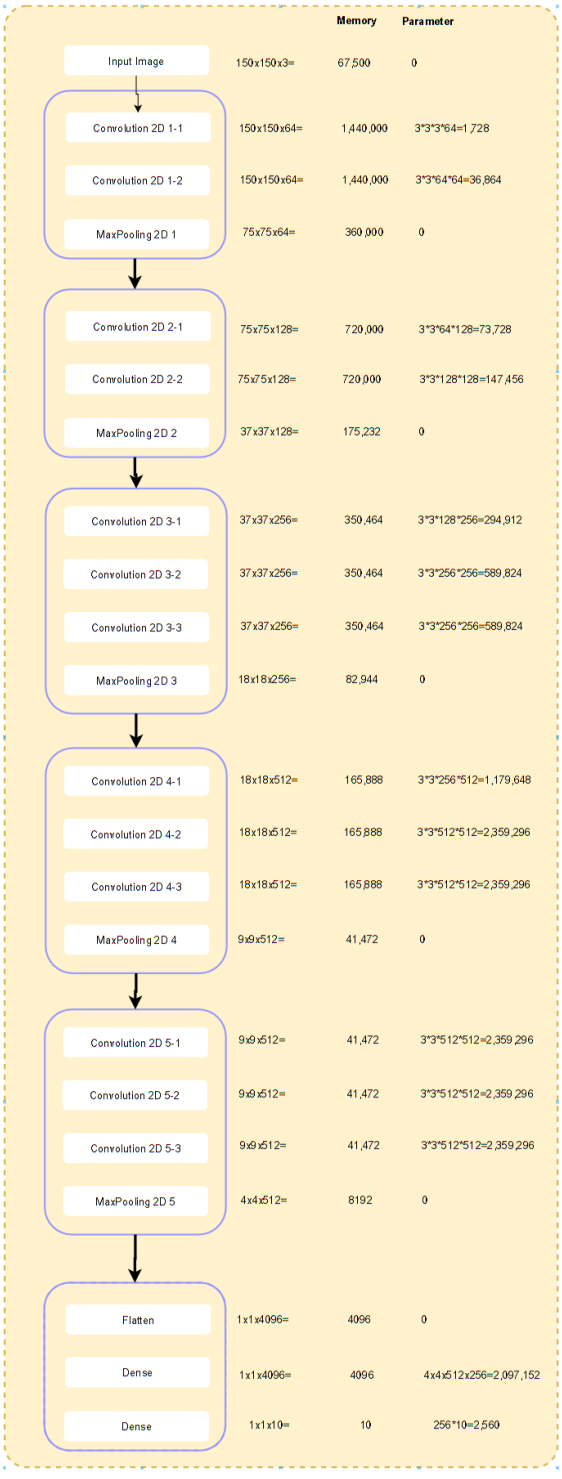
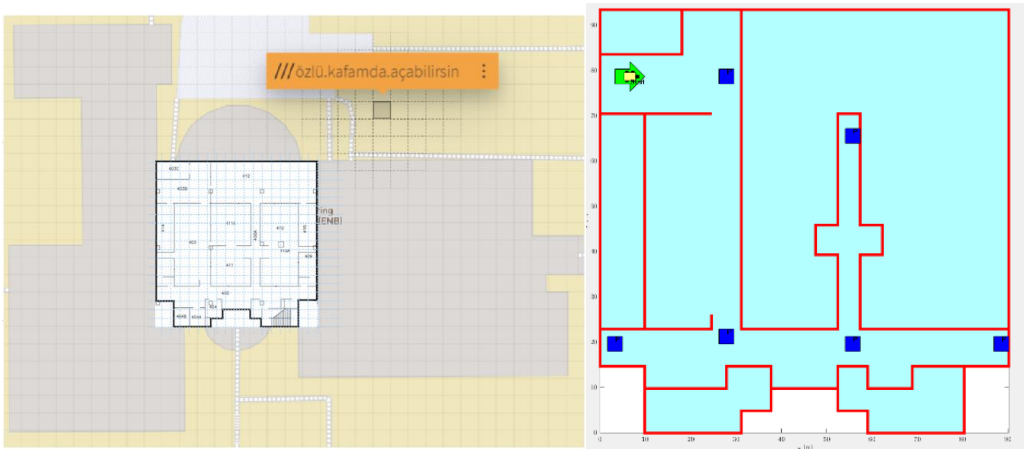


Figure 5.16 Total memory is approximately 27MB per image. Total parameters are 17M parameters.

5.1.3 Indoor Lab Environment

Markers are placed in 10 different points to be detected by the robot. As seen on Fig. 5.17 (b) starting position is illustrated with a green arrow, and waypoints are marked through blue squares which can have more than one navigation markers. The purpose is here to create a path in floor plan and detect the markers to verify it is current position throughout the process.

5.2 Mapping and Path Planner

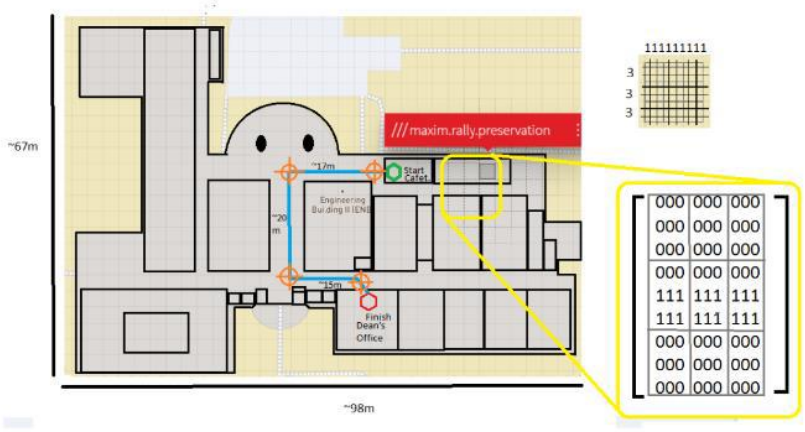


(a) (b)

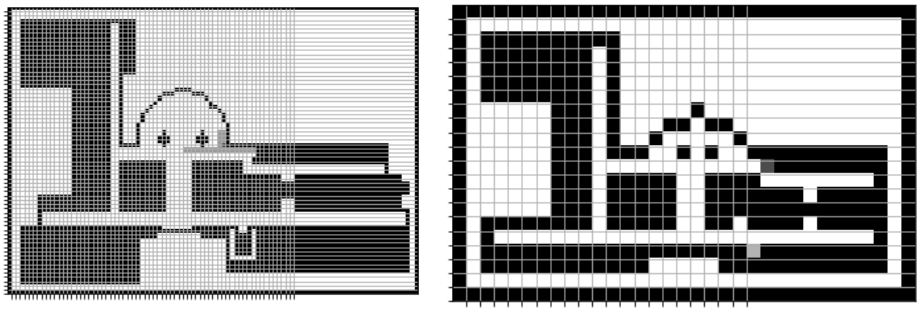
Figure 5.17 Projection of USF Engineering Building (ENB). (a) 4th floor on https://what3words.com, (b) Approximated grid map of ENB on MATLAB

A method for digitizing the floor plan which is what3words grid location system is employed. They composed this location method through a permanent three words representing a 3m by 3m square grid that removes the whole address determination chaos from human life. In terms of a three-word combination, it is possible to address a precise point simultaneously. As one of the system requirements, pose verification of operation is handled through two map configurations which are constructed in Python environment. The purpose of creating two different scaled layouts is to minimize computation time of path planning task. Due to size of the 67x98

matrix requires a great computation delay, it is minimized to 22x33 layout. Hence, the scaling is completed at 1/3 rate.



(a)



(b) (c)

Figure 5.18 Simplified grid maps. (a)ENB on what3words (b) $67x98x1\ m^2$area (c) $21x33x3\ m^2$ area

5.2.1 Occupancy Grid and Q Learner Design

There are two key elements which are environment and a dynamic agent that are considered as in part of a reinforcement learning system. The agent is a wheeled mobile robot in our designed environmental setting.

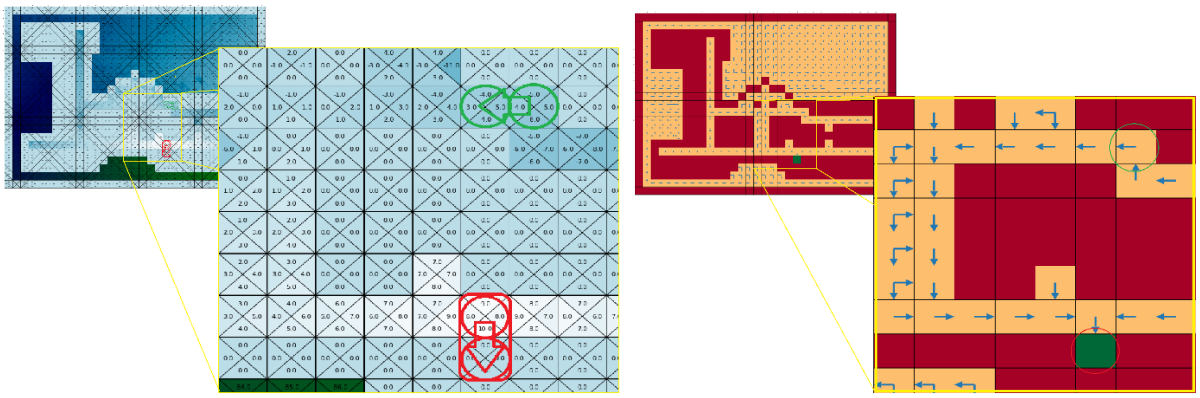


Figure 5.19 ENB 1st floor path planning

The primary step in running of the platform is to launch the relationship between state, action, and reward, since the agent moves into a state by choosing an action depending on the reward at time intervals. In this case, it is required to take an action because of a decision. This decision-making mechanism is called Markov Decision Process (MDP) [35].

$$M = (S, A, P, R, \gamma) \tag{5.7}$$

Table 5.1 Markov decision process parameters

Parameter	Definition
S	$Set\ of\ states$
A	$Set\ of\ actions$
P	$Transition\ probability\ P(s`\|s,a)$
R	$Rewards\ function, R(s) is\ reward\ value\ for\ state\ s$
γ	$Discount\ factor\ (0 < \gamma < 1)$

Bellman equation is given in Eq. (5.8)

$$V^{\pi}(s) = R(s) + \gamma \sum_{s \in S} P\left(s' \middle| s, \pi(s)\right) V^{\pi}(s') \tag{5.8}$$

Optimal value is gathered through the following function.

$$V^{*}(s) = R(s) + \gamma \max_{a \in A} \sum_{s \in S} P(s'|s,a) V^{*}(s') \tag{5.9}$$

- Policy evaluation:

$$\hat{V}^{\pi}(s) := R(s) + \gamma \sum_{s \in S} P\left(s' \middle| s, \pi(s)\right) \hat{V}^{\pi}(s') \ , \forall s \in S \qquad (5.10)$$

- Value iteration:

$$\hat{V}^{*}(s) := R(s) + \gamma \max_{a \in A} \sum_{s \in S} P(s'|s,a) \hat{V}^{*}(s') \ , \ \forall s \in S \qquad (5.11)$$

Transition function and reward function are also defined as in Eq. (5.12) and (5.13).

$$s' = T(s,a) \qquad (5.12)$$

$$r = R(s,a) \qquad (5.13)$$

In order to reach the optimal reward distribution, the greedy policy is executed over the probable actions. Through Q learning and MDP algorithms optimal paths are derived and evaluated. Dynamic programming principles are employed to obtain optimal policy giving maximum reward and short path over the grids from start point to the end. In figure (5.12) the result of policy iteration is demonstrated.

1. Initialization
2. $V(s) \in \mathbb{R}$ and $\pi(s) \in A(s)$ arbitrarily for all $s \in S$
3. Policy Evaluation
4. Repeat
5. $\Delta \leftarrow 0$
6. For each $s \in S$
7. $v \leftarrow V(s)$
8. $V(s) \leftarrow \sum_{s',r} p\left(s', r \middle| s, \pi(s)\right)[r + \gamma V(s')]$
9. $\Delta \leftarrow \max(\Delta, |v - V(s))$
10. until $\Delta < \Theta$ (a small positive number)
11. Policy Improvement
12. policy-stable ←True
13. For each $s \in S$
14. Previous action← $\pi(s)$
15. $\pi(s) \leftarrow \text{argmax}_a \sum_{s',r} p(s', r|s, a[r + \gamma V(s')])$
16. If previous action $\neq \pi(s)$,
17. then policy-stable ← False
18. If policy-stable, then stop
19. return $V \approx v_*$ and $\pi \approx \pi_*$
20. else go to 3

Chapter 6: Results

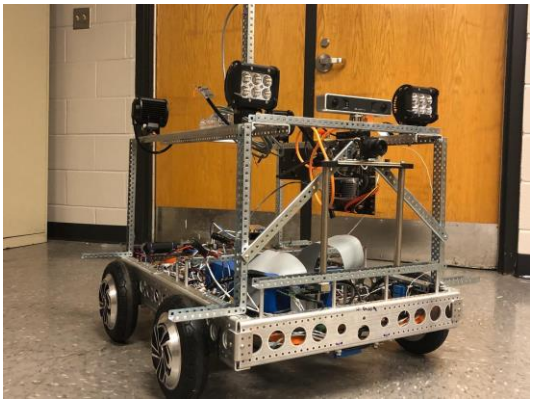


Figure 6.1 DeLRo's overall view

Figure 6.2 Test results of our trained dataset. Green rectangles represent that prediction is successful, while reds illustrate false predictions.

As Fig. 6.2 shows test results are presented with “Pred” and “Act” abbreviations in the titles, which represent predicted and actual labels from zero to nine. Test accuracy is obtained as 78%. One of the reasons is that some markers have quite similar properties. Label “7” and “1” is one of the instances, and “9” and “6” are identical, which have common features due to being vertically and horizontally flipped in datasets. These cases lead to uncertain prediction on test images. Finally, the next step is to improve the dataset to achieve accuracy over %95. This rate is not too far to reach, even 78% percent success is achieved through less dataset (around 1000 images for training, 500 images for validation and 500 images for testing).

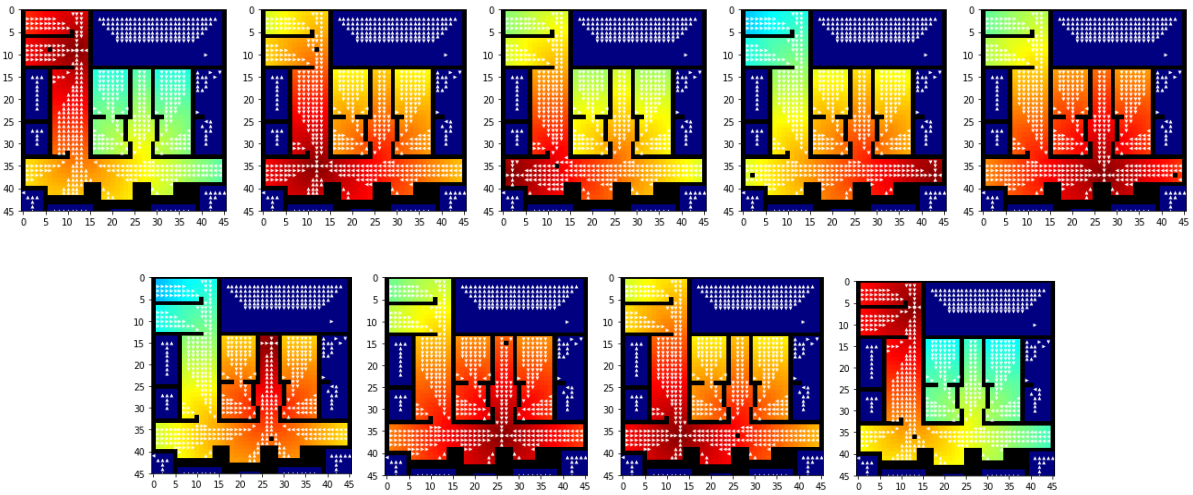


Figure 6.3 Q-Learner path generation results. Each image possesses a little black dot which represents our robot. It is divided into 9 steps, and each step contains marker detecting and path generating based on the floor plan.

It is aimed to create these separate paths and merge them into one route. It is achieved on Spyder (Python 3.7) environment, and since this is an offline process the route map is stored in central processor database.

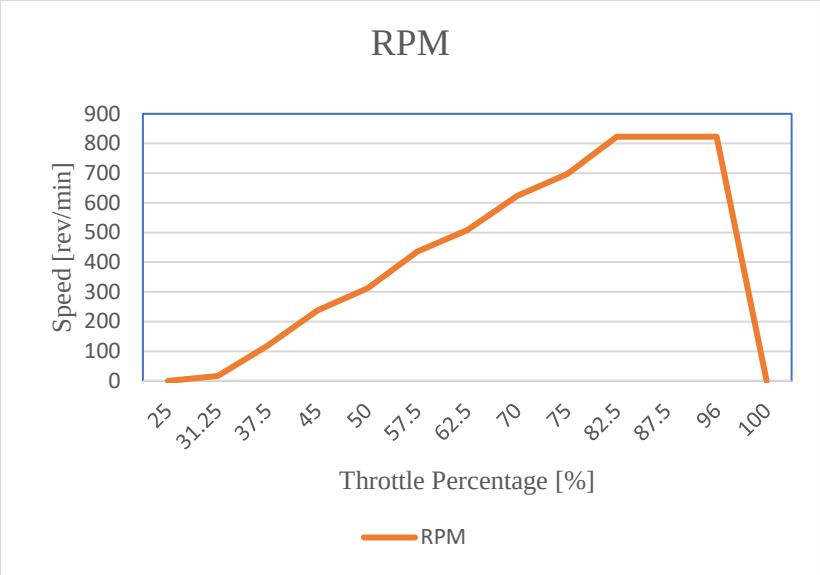


Figure 6.4 Throttle percentage vs. motor speed curve

References

[1] Federal Highway Administration/Federal Transit Administration (FHWA/FTA), "System engineering for intelligent transportation systems: An introduction for transportation professionals", U.S. Department of Transportation, 2007.

[2] Federal Highway Administration (FHWA), "Lesson 13: Walkways, sidewalks, and public spaces", *FHWA Course on Bicycle and Pedestrian Transportation,* U.S. Department of Transportation and the University of North Carolina Highway Research Center. https://safety.fhwa.dot.gov/ped_bike/univcourse/lesson13.cfm.

[3] American Association of State Highway and Transportation Officials (AASHTO), "Guide for the Development of Bicycle Facilities", *The Voice of Transportation,* 4th Edition, Washington, DC, 2012.

[4] OTAK, "Pedestrian facilities guidebook: Incorporating pedestrians into Washington's transportation system", September 1997.

[5] T. Wang, Y. Wu, J. Liang, C. Han, J. Chen, and Q. Zhao, "Analysis and experimental kinematics of a skid-steering wheeled robot based on a laser scanner sensor", *Sensors*, 2015, vol.15, no.5, pp.9681-9702.

[6] P. Lamon, A. Krebs, M. Lauria, R. Siegwart, and S. Shooter, "Wheel torque control for a rough terrain rover," *in Proc. ICRA,* May 2004, pp. 4682–4687.

[7] M. K. Mohamed, A. A. Elkattan, and W. A. Ali, "Friction coefficient displayed by rubber sliding against flooring tiles", *International Journal of Engineering and Technology IJET-IJENS*, vol.12, no.06, pp.144-149, December 2012.

[8] D. Flippo and D. P. Miller, "Advantages of anisotropic wheels in skid steer turns", *International Journal of Advanced Robotics and Automation,* 2016, vol.1, no.1, pp.1-5.

[9] D. Lu, J. Li, M. Ouyang, and J. Gu, "Research on hub motor control of four-wheel drive electric vehicle," *2011 IEEE Vehicle Power and Propulsion Conference*, Chicago, IL, 2011, pp. 1-5.

[10] K. Kozlowski, D. Pazderski, "Modeling and control of a 4-wheel skid-steering mobile robot", *International Journal of Applied Mathematics and Computer Science,* 2004, vol.14, no.4, pp.477-496.

[11] A. Mandow, J. L. Martinez, J. Morales, and J. L. Blanco, "Experimental kinematics for wheeled skid-steering mobile robots", *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems,* San Diego, CA, 2007, pp. 1222-1227.

[12] S. Arslan and H. Temeltaş, "Robust motion control of a four wheel drive skid-steered mobile robot," *2011 7th International Conference on Electrical and Electronics Engineering (ELECO)*, Bursa, 2011, pp. II-415-II-419.

[13] G. K. Fourlas, "Fault detection approach for a 4 - wheel skid steering mobile robot," *2013 IEEE International Conference on Industrial Technology (ICIT)*, Cape Town, 2013, pp. 64-68.

[14] V. Nazari and M. Naraghi, "A vision-based intelligent path following control of a four-wheel differentially driven skid steer mobile robot," *2008 10th International Conference on Control, Automation, Robotics and Vision*, Hanoi, 2008, pp. 378-383.

[15] X. Yun, "State space representation of holonomic and nonholonomic constraints resulting from rolling contacts," *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, Nagoya, Japan, 1995, pp. 2690-2694 vol.3.

[16] R. Dhaouadi and A. A. Hatab, "Dynamic modelling of differential-drive mobile robots using Langrange and Newton-Euler Methodologies: A unified framework", *Advances in Robotics & Automation*, 2013, Vol.2, no.2.

[17] G. Klančar, A. Zdešar, S. Blažič, and I. Škrjanc, "Wheeled mobile robotics: Chapter 2 – Motion modeling for mobile robots", *2017 Elseiver Inc., Butterwoth-Heinemann*, pp.13-59, 2017.

[18] L. Caracciolo, A. de Luca and S. Iannitti, "Trajectory tracking control of a four-wheel differentially driven mobile robot," *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, Detroit, MI, USA, 1999, pp. 2632-2638 vol.4.

[19] U. Ansari, S. Alam and S. M. u. N. Jafri, "Modeling and Control of Three Phase BLDC Motor Using PID with Genetic Algorithm," *2011 UkSim 13th International Conference on Computer Modelling and Simulation*, Cambridge, 2011, pp. 189-194.

[20] H. T. Al-Fiky, M. S. Asfoor, M. I. Yacoub and A. Sharaf, "Speed Control Modeling for In-Wheel Permanent Magnet Brushless DC Motors for Electric Vehicles," *2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR)*, Międzyzdroje, Poland, 2019, pp. 438-443.

[21] Ala Eldin Abdallah Awouda and R. B. Mamat, "Refine PID tuning rule using ITAE criteria," *2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, Singapore, 2010, pp. 171-176.

[22] H. Wang, P. Li, Y. Shu and D. Kang, "Double closed loop control for BLDC based on whole fuzzy controllers," *2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA)*, Beijing, 2017, pp. 487-491.

[23] J. Pongfai and W. Assawinchaichote, "Optimal PID parametric auto-adjustment for BLDC motor control systems based on artificial intelligence," *2017 International Electrical Engineering Congress (iEECON)*, Pattaya, 2017, pp. 1-4.

[24] Microchip, "MCP2551: High speed CAN transceiver", Microchip Technology Inc., 2010.

[25] L. Chen, B. Li, K. Zhao, C. Rizos, and Z. Zheng, “An improved algorithm to generate a Wi-Fi fingerprint database for indoor positioning,” *Sensors 2013*, vol. 13, pp.11085-11096, doi:10.3390/s1308118085.

[26] M. Liu, R. Chen, D. Li, Y. Chen, G. Guo, Z. Cao, and Y. Pan, “Scene recognition for indoor localization using a multi-sensor fusion approach,” *Sensors 2017*, vol. 17, no. 2847, dio:10.3390/s17122847.

[27] https://what3words.com/about/

[28] Myntai, “Mynteye S Sdk Documentation: Release 2.5.0”, 2019. Retrieved from: https://github.com/slightech/MYNT-EYE-S-SDK/blob/2.3.9/docs/src/tools/calibration_tool.rst

[29] R. Cappellaro, “Comparison of stereo visual inertial odometry algorithms for Unmanned Ground Vehicles”, 2019, Politecnico di Torin, MS thesis. Retrieved from: https://webthesis.biblio.polito.it/11648/

[30] R. Girshick, J. Donahue, T. Darrell, J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5)”, *2014 IEEE Conference on Computer Vision and Pattern Recognition,* 2013, pp. 580-587.

[31] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet classification with deep convolutional neural networks”, *NIPS*, 2012.

[32] T. Yang, Q. Ren, F. Zhang, B. Xie, H. Ren, J. Li, Y. Zhang, “Hybrid camera array-based UAV auto-landing on moving UGV in GPS-denied environment”, *Remote Sens. 2018*, vol. 10, no. 1829.

[33] F. Chollet, “Deep Learning with Python”, Manning Shelter Island, 2018.

[34] F. Chollet, “Keras”, Github, 2015. Retrieved from: https://github.com/fchollet/keras

[35] M. L. Puterman, “Markov Decision Processes”, Wiley, 1994