芝浦工業大學
SHIBAURA INSTITUTE OF TECHNOLOGY

Study on a Feasible Planner for Automated Driving Personal Vehicle

By

Muhammad Zulfaqar Bin Azmi

Thesis Submitted to the Graduate School of Engineering and Science,
in Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

September 2021

Thesis Committee
Shibaura Institute of Technology

Professor Toshio ITO
Advisor

Professor Gen ENDO
Member, Tokyo Institute of Technology

Professor Nobuto MATSUHIRA
Member

Professor Hiroshi HASEGAWA
Member

Associate Professor Toshiya HIROSE
Member

Keywords: Artificial potential field, dynamic path planning, indoor outdoor transition, intelligent personal mobility, modular hardware.

"Be in this world, as though you were a stranger or wayfarer."

~Prophet Muhammad s.a.w.~

ACKNOWLEDGMENTS

All praise is for Allah. I praise Him, seek His aid, and ask for His forgiveness, for His Will, Grace, and countless blessings enable me to complete this thesis. Indeed, COVID-19 changes everything, and it teaches me to appreciate things since the future is predictable.

I am very grateful to the Shibaura Institute of Technology (SIT) for the opportunity to study at this incredible university. Definity is not possible without the awarded scholarship of SIT's hybrid twinning program and Japanese Government Scholarship (MEXT/Monbukagakusho).

Thanks to my supervisor Professor Toshio Ito for all the support and knowledge along my Ph.D. journey. He has always been welcoming since day one and always trusted me despite my lack of abilities. With the help of Professor Ito, I can broaden my way of thinking. I get to travel to good countries worldwide and experience the field that I am passionate about.

I am forever indebted to my wife, Dr. Rasyidah Hanan Mohd Baharin, for her unconditional support throughout all my difficulties, despite having to struggle with her Ph.D. studies as well. She has always been patient with me, and she made me a better person.

I thank my beloved umi dan baba, Mdm. Arizah Ibrahim and Mr. Azmi Yusoff, brother Muhammad Firdaus Azmi, and sister Fatihah Azmi, for always

believing in every decision that I've made. To my parents-in-law, Mdm. Ragayah Omar and Mr Mohd Baharin Harun, I am grateful for the prayer and support.

I appreciate all the presence of my friends in Japan, Malaysia, and other countries in the world. I get to learn a lot of Japanese culture from the members in Itolab, and I get to know the world's update from my friend all over the world.

Finally, I thank all the committees in my Doctoral evaluation, internal examiners Prof. Nobuto Matsuhira, Prof. Hiroshi Hasegawa and Assoc. Prof. Toshiya Hirose, and external examiner Prof. Gen Endo from Tokyo University of Technology, for their time and effort in reviewing my thesis submission. Their comments have allowed me to improve my writing, and with this experience, I hope I can help others.

ABSTRACT

The complete experience of an automated driving system (ADS) often requires sophisticated hardware. As the personal mobility vehicle (PMV) is classified as a small vehicle, incorporating such a setup is challenging without direct modification and tempering of the original vehicle. Therefore, the modular design architecture approach for autonomous vehicles is imperative to prevent total malfunction of PMV in the case of system failure.

In chapter 3 of this study, the author developed a standalone-modular autonomous PMV (SMAD-PMV), which the operation is limited to two sensors, namely LiDAR and omnidirectional camera, both retrofitted externally using a custom metal bracket without needing internal modifications. The SMAD-PMV's motion planner was investigated in three cases: a global path planning deployment and two local planning approaches, the dynamic artificial potential field (APF) with the virtual obstacle (VO) method, and the region-based planning strategy (RBPS). The RBPS corresponds to the modularity aspect of the SMAD-PMV, where the planner behaves locally based on a predetermined behavior. The validation was done in a combined outdoor and indoor environment, with the SMAD-PMV excel in planning a trajectory and navigating it in real-time without crashing into any obstacle or landmarks. The system has been proven to operate satisfactorily despite the significantly reduced number of sensors.

Chapter 4 related map orientation to the path planning results. The point cloud map is discretized and rotated, and comparison between two path planning algorithms, APF, and A* algorithm, is included for validation. It is observed that

for indoor cases, the map with right-angled orientation results in a path that is well distanced from the wall.

Chapter 5 proposes improving the APF algorithm by including VO to local escaped minima and the dynamic potential map to store dynamic obstacle information. The result and discussion section also highlight the perspective of RBPS in reducing the number of VO needed.

Finally, Chapter 6 concluded the thesis content, and all objectives are revisted once again. The future works are also highlighted.

TABLE OF CONTENTS

	Page
THESIS COMMITTEE	i
DEDICATIONS	ii
ACKNOWLEDGMENTS	iii
ABSTRACT	v
LIST OF TABLES	x
LIST OF FIGURES	xi
LIST OF ABBREVIATIONS	xiii
CHAPTER	
1 **INTRODUCTION**	1
1.1 Background and Motivation	1
1.1.1 Personal Mobility Vehicle (PMV)	2
1.1.2 PMV accidents	5
1.1.3 Smart PMV	5
1.2 Problem Statement	6
1.3 Objective	8
1.4 Novelty	9
1.5 Scope	9
1.6 Thesis organization	10
2 **RELATED WORKS**	12
2.1 Introduction	12
2.2 Automated Driving System Design Philosophy	12
2.2.1 Standalone vs. Connected System	12
2.2.2 Modular vs. End-to-End Architecture	13
2.2.3 Selection of the System and Architecture	17
2.3 Attempts on Modular-Based ADS	17
2.4 Novelty of Modular PMV	19
2.5 Path Planning Technique	19
2.5.1 Global path planning	21
2.5.2 Local path planning	22
2.5.3 Challenging situation: Glass Wall	22
2.5.4 Criteria for choosing APF and A* for testing the algorithm	23
2.6 Conclusion	24

3 **DESIGN AND DEPLOYMENT OF STANDALONE-MODULAR AUTOMATED DRIVING PERSONAL MOBILITY VEHICLE** 25
3.1 Introduction 25
3.1.1 SMAD-PMV Prototype 26
3.1.2 Sensors, Power, and Computing System 31
3.1.3 Controller's hardware interface 34
3.2 Software Architecture 34
3.2.1 Mapping and Localization 36
3.2.2 Path planning 38
3.2.3 Region-based planning strategy (RBPS) 38
3.2.4 Controller 41
3.3 Experimental Procedure 42
3.4 Result and Discussion 45
3.5 Conclusion 51

4 **PROPOSED PATH PLANNING METHOD 1: AUGMENTED BY ROTATION MAP** 53
4.1 Introduction 53
4.2 Path Planning 54
4.2.1 Artificial Potential field 54
4.2.2 A* Algorithm 56
4.3 Proposed Method for Map Augmentation 57
4.3.1 Non-augmented vs Augmented 57
4.4 Experimental Setup 60
4.5 Result and discussion 61
4.6 Summary 65

5 **PROPOSED PATH PLANNING METHOD 2: DYNAMIC ARTIFICIAL POTENTIAL FIELD WITH VIRTUAL OBSTACLE** 67
5.1 Introduction 67
5.2 Dynamic Map 68
5.3 Virtual Obstacle 69
5.4 Improving dynamic APF with RBPS 70
5.5 Experimental Setup 71
5.6 Result and Discussion 73
5.6.1 Experiment 1: Utilizing dynamic APF and VO is path planning problem 73
5.6.2 Experiment 2: Utilizing RBPS in dynamic APF 78
5.6.3 Comparing improved APF and A* 82

5.7 Conclusion 84

6 **CONCLUSION AND FUTURE WORKS** 90
6.1 Summary of Research Contributions 90
6.2 Future Works 92

REFERENCES/BIBLIOGRAPHY 94
APPENDICES 100
LIST OF PUBLICATIONS 107

LIST OF TABLES

Table **Page**

4.1 Time taken to compute the total APF map. 56
4.2 Time taken to compute one-time obstacle map (with distance penalty) for A*. 57

A.1 Suaoki S270 150Wh Portable Mobile Battery 100
A.2 Velodyne VLP-16 specifications 101
A.3 Kodak Pixpro 360 Specification 102
A.4 Asus UX461 Specification 103
A.5 Voltage regulator Specification 104
A.6 Arduino Mega (ATmega2560) specifications 105
A.7 Steering servo (Gearwurx Torxis) 106

LIST OF FIGURES

Figure **Page**

1.1 Three phases of a *whole trip*. 2
1.2 Thesis organization. 11

2.1 Modular architecture of the Automated Driving System. 15
2.2 End-to-end architecture of the Automated Driving System. 16

3.1 The base model, Suzuki Senior Car ET4D as advertised and commercially available[1]. 26
3.2 The SMAD-PMV prototype showing 3D printed caps and metal bracket. 27
3.3 Three-dimensional geometry of the retrofitted mount showing sensors platform and PC platforms, while the steering mechanism is highlighted in light green color. The five points (one ①, two ②s and 2 ③s) indicate locations in which the steering mechanism has degrees of rotational freedom. 29
3.4 The front look of the vehicle where the system is mounted. 30
3.5 Installation of handle's holder. 30
3.6 The installation of Accel Servo at the acceleration pedal. 31
3.7 LLC is positioned under the PC's platform. 32
3.8 The diagram of the connection between the computer and the LLC. 35
3.9 Control Flow. The module is broadly categorize into three, i.e. mapping and localization, motion planning and controller. 35
3.10 The flowchart of RBPS strategy. 40
3.11 Illustration of region in front of automatic door. 41
3.12 Illustration of deployment map with the top view from google map. 43
3.13 Result of modifying lidar FoV configuration when the vehicle is in the predefined region. 47
3.14 (a) covers automatic door, first corner and narrow area. (b) covers second corner. (c) covers small ramp. 49
3.15 The point cloud map of the campus area with referenced trajectory (red line) and generated trajectory (black line). The region planner can be strategically defined on the automatic door area. The green box is where the SMAD-PMV halts while waiting for the automatic door to open. 51

4.1 Attractive and Repulsive force field. 54
4.2 Landmarks within the map before being discretized. 58

4.3 Discretized map with the original orientation. 58
4.4 Discretized map with the transformation applied. 59
4.5 Comparison between non-augmented and augmented map. During map preprocessing phase, it is better to adjust the orientation to allow standardize result. 62
4.6 Comparing the characteristic of path in non-augmented and augmented map using A* and APF. 63
4.7 Comparing the characteristic of path in non-augmented and augmented map using A* and APF. 64
4.8 Denenchofu areas (coordinate 35.597130232185606, 139.66728981970635), as shown in Google map. 65

5.1 Generated dynamic potential map. 68
5.2 Illustration of region generating predefined virtual obstacle point for indoor. 71
5.3 Result of APF for dynamic obstacle case. 74
5.4 Result of APF with dynamic map for glass wall area case. 75
5.5 Dynamic map visualized. 76
5.6 Result of Fig. 5.3(b) (red) and 5.4(b) (blue) in point cloud map 77
5.7 b2f2 static potentials and the region location 78
5.8 APF searches for the goal, (x_g, y_g). The result of with and without region is compared. 79
5.9 The b5f1 static potentials and the region location. 80
5.10 APF searches for the goal, (x_g, y_g). The result of with and without region is compared. 81
5.11 Comparing the characteristic of path in non-augmented and augmented map using A* and APF. 85
5.12 The results of Fig. 5.11 shown in point cloud map. 86
5.13 Another Comparison of the characteristic of path in non-augmented and augmented map using A* and APF. 87
5.14 The results of Fig. 5.13 shown in point cloud map. 88
5.15 Time taken for all 6 scenarios 89

A.1 Velodyne VLP-16. 100
A.2 Velodyne VLP-16. 101
A.3 Kodak PixPro360. 102
A.4 Asus UX461. 103
A.5 HiLetgo DC-DC regulator. 104
A.6 Arduino Mega. 105
A.7 Gearwurx Torxis Servo Motor. 106

LIST OF ABBREVIATIONS

SMAD	Singular Modular Automated Driving
ADS	Automated Driving system
APF	Artificial potential field
AUV	Unmanned Underwater Vehicle
AV	Autonomous Vehicle
VO	Virtual obstacle
RBPS	Region-based planning strategy
MDA	Modular design architecture
MA	Modular Architecture
PO	Planning Operation
SO	Sensor Operation
MO	Monitoring Operation
CO	Control Operation
LLC	Low-level Controller
PMV	Personal mobility vehicle
PP	Path Planning

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

A sustainable transportation system is essential to ensure people's accessibility. One of the significant challenges to this is the completion of infrastructures that allows people to complete a whole trip, which is comprised of three phases; the first mile, the transit, and the last mile as shown in Figure 1.1. The first and last mile issues can be described as the limitation in ferrying people from a transportation hub (e.g., bus stop) to their final destination (e.g., home). For example, in some rural areas, the lack of access to transit services may restrict user access to public transport. Furthermore, the same issue may be occurring in a public area such as parks, or shopping malls, where it is not accessible by a larger vehicle may suffer. The first and last mile has always been one of the many challenging areas of research in ADS. Two examples of the problem include maneuvering amongst human crowds or narrow spaces, and indoor localization. The challenges also include real-time performance and price. The majority of ADS in the market usually address 'the transit' phase. Therefore, it is ideal to address all three phases concurrently to experience the "full ADS" or point-to-point travel.

The elderly and disabled are the categories of people who are experiencing difficulties from these phases. Based on World Health Organization's report on disabled people, approximately 1 out of 5 people in this world is suffering from some form of disability, in which a part of them considered severe[2]. The risk increases as people get aged. Based on the Population Division of the United

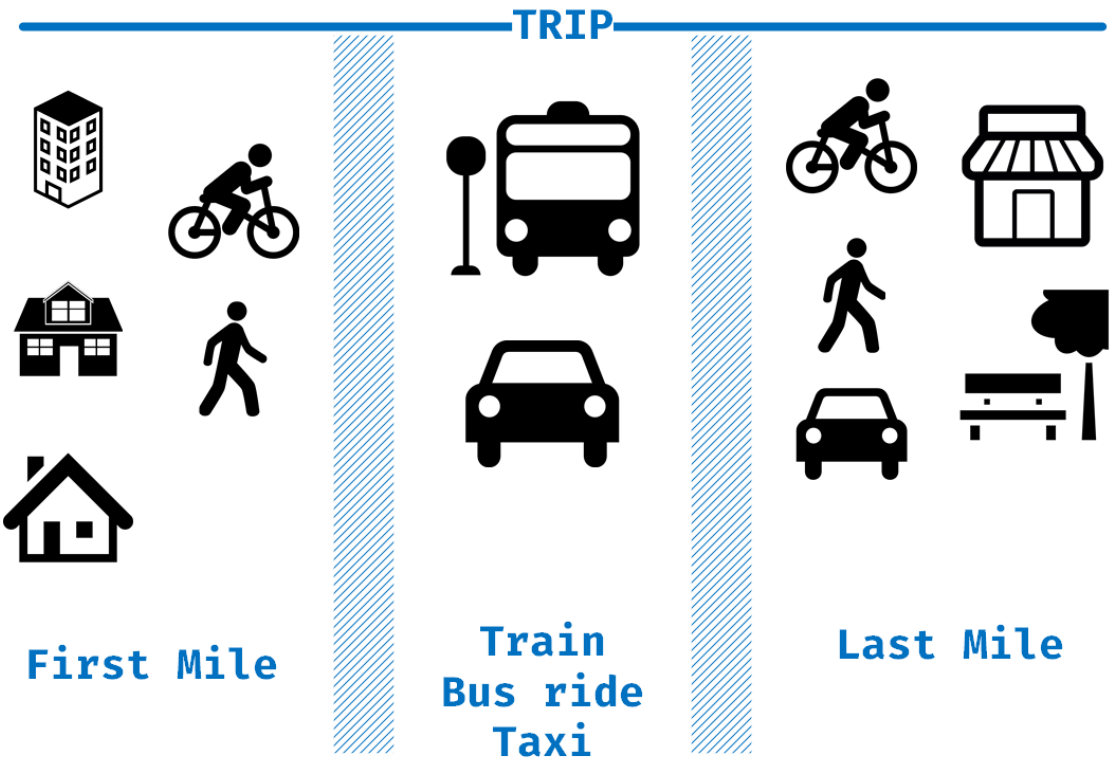


Figure 1.1. Three phases of a *whole trip*.

Nations Department of Economic and Social Affairs, the aging population trend is increasing everywhere in the world, with Japan placing on the top of the list[3]. This phenomenon proves that plenty of people from these numbers may require additional assistance within the two phases mentioned above.

1.1.1 Personal Mobility Vehicle (PMV)

Several kinds of research using a vehicle that can navigate in an area accessible only to pedestrians or cyclists have been conducted. A golf cart is one example of it. With more open source software available to create one's environment simulating a golf cart for autonomous system development is made possible [4]. Members of the Singapore-MIT Alliance for Research and Technology (SMART) have conducted

over six days of trial for their self-driving golf cart. During this time, it has ferried the visiting tourists along the path, which is usually trafficked by pedestrians and cyclist[5]. However, these types of vehicles can only be operated at a specific place that is inaccessible by the public. Although there are suggestions for using bicycle services to solve the last mile problem[6], it does not entirely address the issues, as the bicycle is not convenient in accessing certain areas, such as a room with a door-width entrance.

Some other means of transportation are more suitable to be used in the last-mile phase. The most convenient one is the personal mobility vehicle (PMV). There are two main types of elderly-friendly PMV in Japan; the powered wheelchair and the motored scooter. Based on the specification defined in Japan Industrial Standards (JIS) T9208, the maximum allowed speed for PMV is 6 km/h[7]. The regulation allows them to be treated as a pedestrian and can be operated freely in most places accessible by one. Its low speed resulted in operation on the main road impossible, but PMVs can also be operated onboard most trains, as long as the train station provides adequate facilities, such as elevator and ramps, in addition to adhering to all of the following criteria:

- The dimension is no more than 1.2 m $\times$ 0.7 m $\times$ 1.2 m (length $\times$ width $\times$ height).
- Able to make a turn at a *right-angled road* with 0.9 m width with up to 5 attempts.
- Able to make a turn at a *right-angled road* with 1.0 m width without any reversing.

- Able to perform 180° rotation when the width of the road is less than 1.8 m.
- Display *3-star rating* for the rotation performance or *improved handled type* electric wheelchair sticker.

The sales of the PMV were recorded at nearly half a million, with 15,000 units are being added annually[7]. The reason for the sales increase is due to more of the elderly community viewing PMV as a feasible alternative form of transportation.

Besides providing mobility for people with difficulties walking, the following are some other potential benefits of PMVs to consumers and society[8]:

- *Lower total operation cost compared to larger vehicles.* The price for a PMV is less than ¥400,000 and does not require large expenditure for licensing and maintenance.
- *Low noise and less-to-no emission.* PMV can operate in near silence, and as it did not have any internal combustion, it will not produce a significant amount of toxic fumes.
- *Lower trip times for short-distance travel.* The vehicle can continuously be operated up to 10 km. This distance is slightly too far for walking, whereas the time and cost taken to drive or take public transportation are relatively high. For example, walking from Shibaura Institute of Technology's Omiya campus to Higashi Omiya station might take 20 minutes to walk. The same trip by PMV can take less than 10 minutes, assuming the PMV is parked nearby.

1.1.2 PMV accidents

Despite its benefits, PMV is not accident-free. Based on the Association for Technical Aid Guide, most accidents related to motored scooters happen when people go shopping. About 75% of the contributing factor is due to inappropriate handling or negligence by the user. Among them, 40% are new users who are not familiar with how to handle the vehicle, and most of them are old people. Hence their reaction time during such critical situations is significantly reduced compared to younger people. The reason is further supported by the statistics showing that most fatal accidents (73.6%) happen in the intersection where the reaction time is the crucial factor in avoiding such mishaps. To avoid such accidents, the development of an intelligent PMV is indispensable.

1.1.3 Smart PMV

A smart PMV often combines small-scaled mobile vehicles with various perception sensors for detection and navigation. The research and development that specifically point to an intelligent PMV can be dated back to 1992. For example, The NEC Corporation Japan in 1992 has introduced the automated guided wheelchair, which localizes itself around the perimeter using infrared (IR) beams guided by a set of magnetic ferrite marker tape [9]. Although this attempt works very well in an indoor setup, it is not feasible for outdoor deployment due to the absence of difficulties of freely placing such marker taps. In addition to localization, the vehicle is expected to have flexibility, such as avoiding obstacles and navigating through cramped spaces. Hence, the development of smart PMVs is often expected to be an all-rounder, tackling both of these major points.

The previous development of the Autonomous Mobility Scooter from SMART has been a success when the research group conducted public trials in Singapore. Similar to their preceding golf cart project, this joint research aimed at establishing a comprehensive mobility-on-demand service. The autonomous scooter is equipped with similar software as their other self-driving vehicle research. In their research, the steering column was detached, and a servo motor was added to steer the scooter [10]. Though this allows for smoother navigation, it will consequently restrict the user's capability to control the vehicle whenever they do not desire to use the autonomous system.

Interestingly, researchers from the University of Electro-Communications, Tokyo, have also conducted studies using the 4-wheeled Suzuki scooter, which is a similar model as the base platform of this study [11]. In their study, the vehicle steering shaft is modified to install a self-driving mechanism consisting of a pulley and a DC motor. Three units of 2D laser scanner, wheel encoder, and GPS are used to localize their system. The study was conducted in the outdoor area and focused on the development of a feasible navigation system. However, they did not perform any indoor test.

1.2 Problem Statement

Often to build an ADS, an invasive method is used whereby most of the vehicle's parts are modified to accommodate the hardware installation. However, if any unforeseen error happened, the whole vehicle is not operable until the entire system is debugged and fixed. Furthermore, as the system needs to be economical, the number of sensors and hardware must be kept at a minimum. The consequence

of this is the reduced amount of computational capability. Therefore, firstly, a modular system and able to perform safely needs to be designed to accommodate these limitations.

Secondly, with the availability of many state-of-the-art path planning algorithms available, hierarchical system design can be made. However, with the reduced computational power, the decision-making tree also needs to be pruned, thus reducing much possible states. The classical planner is best suited due for this instance, has been proven to work. However, more investigation and testing on the suitable algorithm needs to be done.

The graph search-based planner often deals with an occupancy grid or any approximation of a discrete cell-grid space among the path planning algorithms. For example, the Djikstra Algorithm locates the single-source shortest path. However, the algorithm also suffers from performance issues. Its extension, the A* algorithm, solves this problem by considering the heuristic function [12]. The function enables fast node search. However, as the method searches for the shortest path, the generated trajectory tends to lead the vehicle to be too close to the obstacle. In the case of vehicles operating indoors, driving too close to any objects may be hazardous, as accidents may occur due to the possibility of humans exiting doors. Thus, an obstacle-distancing method for the vehicle is needed.

Thirdly, a common method to penalize proximity between vehicles and any known landmark in the map is by using a potential field [13]. One of them is the Artificial Potential Field (APF) method. Some studies employ APF to create the

potential of road boundaries which allows the vehicle to maintain within its lane on the road, simultaneously allowing a safe distance for any sudden appearing object [14, 15]. It also has been used to define human personal space [16]. However, the method also suffers in the wall proximity unless further parameter tuning done. At the same time, it is computationally expensive if the number of points is significant. Nevertheless, it can be optimized by transforming the field into a 2D grid map [17].

The author then considers other means to transform the map in this dissertation. The transformation of one type of space to another type is one technique to reduce computational complexity. Machine learning algorithms typically use the transformation method. For example, in Support Vector Machine, the field is transformed using the kernel trick, which increases the dimension of the field [18]. In data science, the data are augmented to increase data variation [19]. It is found that when augmented data are used to test the non-augmented model, the classification error is more prominent despite the same batch of images being used. It is crucial to study how augmentation affects data.

1.3 Objective

The three main objectives for this thesis are presented as following.

1. To develop a modular system for an automated driving PMV.
2. To investigate on the influence of map's orientation towards the result of path planning.
3. To develop a modified APF algorithm that solves the local minima trap which revolves around map augmentation.

Objectives (1), (2), and (3) corresponds to the first, second, and third problem statements, respectively, elaborated in the previous section.

1.4 Novelty

From traditional to modern systems, there is a significant amount of work that has been well-tested, which in the author's point of view, has a better performance compared to the one proposed in this study. However, the study is focusing on a novel non-invasive, modular prototype as the test platform for an automated driving mobility vehicle, where the base is a commercially available Senior Car in the market, highly accessible to the masses.

Another novel point is the map augmentation as an alternative to address path planning issues. There are some situations where the currently available path planning method is not suitable to be utilized, for example, when the map does not provide an accurate description of the location. The study will investigate how map augmentation (same map but different orientation) will help to solve this problem. This will be further discussed in Chapter 4.

1.5 Scope

The following are the scopes of this thesis:

- The control algorithm is assumed to work, and it will not be addressed in depth.
- Only the forward motion is covered.

- Path planning algorithm's tuning will not be covered.
- Path planning algorithm's performance analysis (speed et cetera is not covered.)
- The map augmentation analysis only covers the indoor area.

1.6 Thesis organization

The remainder of this thesis is organized as shown in Figure 1.2. First, the author focuses on a general automated driving PMV and its design, then moves on to the path planning method. The outline of each chapter their relevant contributions to the existing literature and publications are put into context. In Chapter 2, the related literature is discussed, giving the broad overview of the modular ADS design philosophy, followed by the introduction to path planning. Chapter 3 dissects the development of the PMV prototype used in this study, the Standalone Modular Automatic Driving-Personal Mobility Vehicle (SMAD-PMV). This includes the external mechanism retrofitted noninvasively to the PMV. Chapter 4 discuss the motivation of map augmentation by rotation and then validates the method by comparing its deployment between Static APF and A* algorithm using the SMAD-PMV prototype. Then, based on Chapter 4, the APF method was found to be more consistent and stable. Hence the work was extended to dynamic map APF in Chapter 5. The dynamic map APF solves a recurring problem in static APF, which is the local minima trap. Finally, concluding remarks and a brief summary of the thesis are presented in Chapter 6, where the author also provides an outlook to possible directions for future research.

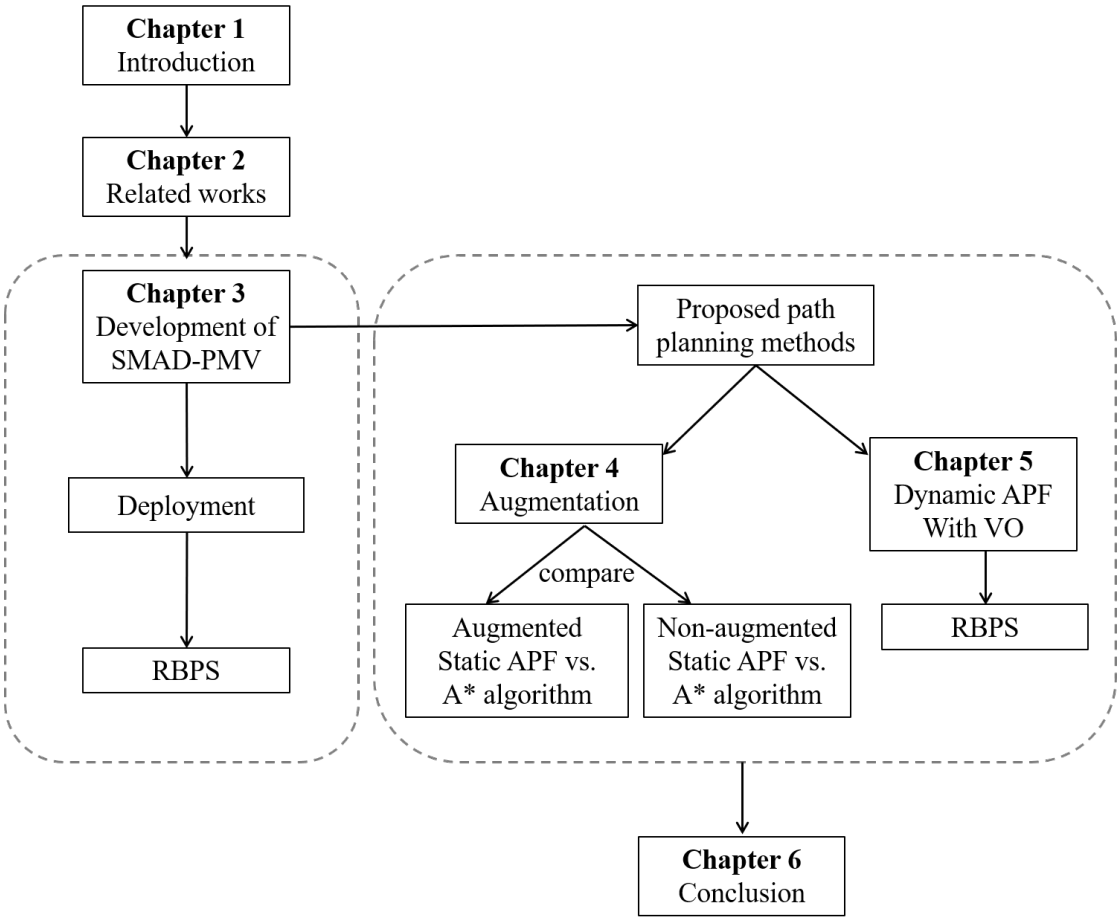


Figure 1.2. Thesis organization.

CHAPTER 2
RELATED WORKS

2.1 Introduction

In this chapter, an extensive literature review is conducted to obtain an overview of the study's objectives and the proposed solution. For that purpose, the list of related works related to the modular hardware design of the automated driving system (ADS) and the path planning (PP) solutions will be shown and discussed.

2.2 Automated Driving System Design Philosophy

ADS describes the term automated driving as a whole; in other words, it covers all the fields of autonomous vehicles and self-driving. ADS encompass a variety of field, which includes remote sensing, computer science, and et cetera. The operation domain also covers underwater, land, and air. Fortunately, all of them follow some similar classification of design and philosophy's, which will be discussed in the following subsections.

2.2.1 Standalone vs. Connected System

A high-level classification of ADS design philosophy can be divided into two; *standalone* and *connected* systems. A standalone system-based design places everything into a single vehicle, which allows it to operate independently without any external processing element. Currently, this is the conventional approach that is practiced by most modern ADSs. However, the main disadvantage is that the installed system heavily relies on the onboard sensors to read environmental information; therefore, the flexibility becomes limited once the related sensors are

removed.

In contrast, the connected system is not yet widely practiced, but it can potentially change the future direction of ADS. This system type is often referred to as *vehicle to infrastructure* (V2I) or more comprehensively *vehicle to everything* (V2X). Its practice can overcome several bottlenecks of the standalone system, such as improving sensor operation ranges, blind-spot monitoring, and computation limitation. This is possible due to the general concept of V2I/V2X that allows the vehicle to get information from stationary sensors attached to infrastructures or even from the surrounding pedestrian's mobile phone data. However, many factors deter the system from being fully realized.

First, the connectivity limitation. In this sense, it means that the system must have access to a high-speed, low latency 5G or 6G internet connection, which is still under constant development. Aside from latency issues, the ethical privacy is also becoming a concern, such as the requirement of publicly sharing one's mobile device information and data mishandling. Due to these limitations, the standalone system is still considered the most practical and realistic development platform since the proposed system needs to be flexible in terms of the location where it is being operated. In addition, it should not depend on additional network features to function.

2.2.2 Modular vs. End-to-End Architecture

There are two options of architecture that are adopted to complement either of the philosophies; *modular* and *end-to-end driving*.

2.2.2.1 Modular Architecture

A *modular design architecture* (MDA) consists of pipelines that separate each ADS module into a separate category and link them one-by-one starting from Sensor's Operation to the input that drives the actuators. Figure 2.1 illustrates one of many possible architectures for ADS[20, 21].

The core operation of a modular ADS can be categorized as follows; *sensor operation* (SO), *planning operation* (PO), *monitoring operation* (MO) and *control operation* (CO). The high-level overview of the system is as follows.

- Sensors read through the environmental information and feed the raw data to the SO modules.
- In SO status, the raw sensor data is processed, and the 3D representation of the environment is built. Examples are localization and object detection, and recognition.
- The PO will obtain the processed information and create a tactical plan on how the ADS should behave at all times.
- CO will then execute these commands.
- Actuator will be driven based on all instructions given by the CO.
- MO will constantly monitor the status of all these operations to ensure the system is in healthy condition.

By separating each component, the problem now becomes a smaller scale and much simpler to solve. Furthermore, several critical scenarios can be anticipated

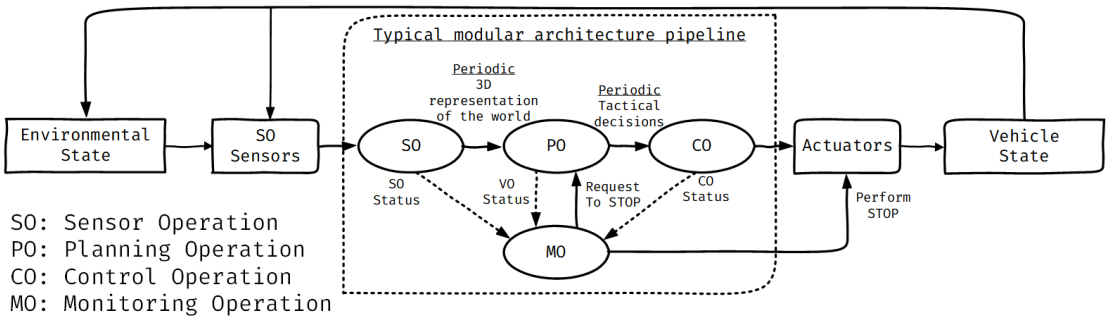


Figure 2.1. Modular architecture of the Automated Driving System.

earlier, and the safety mitigation to respond to these situations can be hard-coded as the rule-based system. This architecture, however, is prone to over-complexity and vulnerable to errors, as the system will not be able to react to the scenario that is excluded in the set of rules. In addition, misinformation from the SO may be fatal[22].

2.2.2.2 End-to-end Driving Architecture

The motivation behind the end-to-end driving architecture is to imitate a real-human driving behavior. The typical design of end-to-end ADS is as shown in Figure 2.2. Unlike MA, end-to-end driving does not separate each module into multiple different operations. Instead, the architecture utilized machine learning (ML) to generate vehicle motion commands from sensors directly. As a result, end-to-end driving relies on one of the following methods: supervised or online learning approach.

In the supervised approach, Deep Convolutional Neural Network (DCNN), the driving data is first collected from human experts. After that, these data are preprocessed and labeled. Next, a model is trained and then tested. The

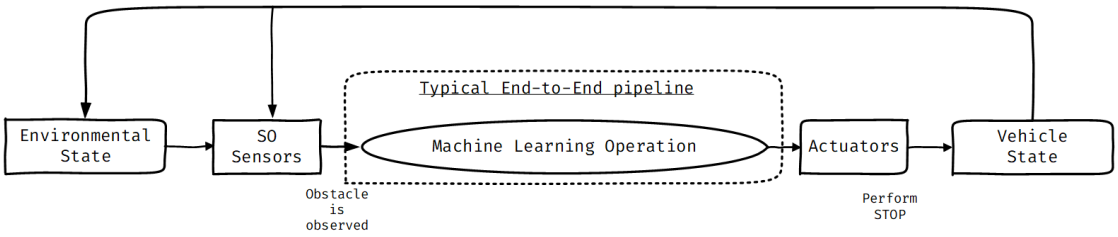


Figure 2.2. End-to-end architecture of the Automated Driving System.

sequence is a cycle of repeating process until a satisfactory result is achieved[23]. The disadvantage of the supervised approach is that it requires a vast amount of data to perform feasibly. Furthermore, it is incapable of learning by itself and relies only on the knowledge available in its model. Therefore, it can not handle unknown scenarios well.

On the contrary, the online learning approach utilizes Deep Reinforcement Learning (DRL) to learn how to drive. The learning agent is employed to interact with the environment via the sensors[24, 25, 26]. The agent's success or failure depends on the reward function defined in the DRL framework. The main challenge in end-to-end learning is the reward function itself, because the agent needs to be aware of the "correct" actions to yield a high reward during the learning process. The learning requires the agent to explore new actions that may have a risk of failure. In the real world, these actions might be dangerous. Therefore, the end-to-end driving is usually validated through simulation environment [25] or in a rural area where the number of pedestrians and traffic is little to none[26].

A common disadvantage in both end-to-end driving methods is the inability for the user to understand the model entirely, because it is basically a "black

box", where the internal workings are unknown. In other words, the system itself is complex and challenging to diagnose whenever issues occurred. Furthermore, attitude of repetitive training cannot be relied upon since it might result in casualties during the actual driving scenario.

2.2.3 Selection of the System and Architecture

Based on the brief explanations in the previous subsections 2.2.1 and 2.2.2, the standalone and modular approach are chosen as the system and design architecture, respectively. This is because the standalone system complements modular architecture well where both hardware and software modularity can be achieved simply by selecting a proper equipment and exploiting its functionality. Furthermore, the selection is deemed safer since rule-based safety measures can be hard-coded to the system depending on the social rule of the ADS's operational domain.

2.3 Attempts on Modular-Based ADS

Based on the selection that was made in the previous section, the author will now delve deeper into some previous works that attempts MDA on their ADS system. Overall, a complete experience of a vehicle equipped with ADS has two essential requirements; firstly, to search for a traversable path, and secondly, to travel along it in real time[27]. In order to accommodate these demands in a swift manner, it is not uncommon for developers to opt for high-end, sophisticated hardware to handle the computational load. More than often, the modifications are also made directly into the vehicle[28]. This could be disadvantageous for a PMV as the installation space for hardware is limited, proportional to the vehicle's compact

size. In addition, the hard-wired modifications could lead to total malfunction of the vehicle in case of a defect to one of the components. Similar occurrences could also happen from the software's point of view. Therefore, the PMV would benefit from a MDA. As previously mentioned, the basic idea of a modular approach is to segregate the system into several parts, each of which can function independently on its own but ultimately contributes to achieving a set of predetermined goals. This definition is valid for both hardware and software aspects of the vehicle. Modularity eliminates the need to troubleshoot the whole system in the event of a breakdown and is indispensable for expediting the reconfiguration process [29].

The MDA for AVs was previously reported in several studies [30, 31]. An early example is a preceding work by Smith [30] which proposed an "ultra-modular" autonomous underwater vehicle (AUV) dubbed as Morpheus, where the plastic-injected vessel and cabling system are deliberately separated from the other main components in order to avoid water damage. In a similar field of application, the researchers from Cornell University studied a dual-vessel configuration for AUV where one vessel could be easily serviced without disturbing the other [32]. In both cases, the modularity was centered on the hardware feature. Shifting the focus closer to the author's study, which is the land-operating vehicles, more recently, a California-based company PerceptIn has introduced their DragonFly Pod system that targets the Mobility-on-Demand (MoD) market [31, 33]. The DragonFly Pod characterized both the hardware and software with the MDA approach.

Even though these works were practical to build up the understanding of MDA for AVs, they were either carried out for a separate target or groups of

people different from the author's research purpose. Predominantly, the AUVs do not have the human-carrying capacity in addition to serving unmanned underwater expeditions, while the pod cars in principle are not meant for indoor operation and on pedestrian pavements. Moreover, several proclaimed modular vehicles were often custom-built or made from scratch, meaning they are not readily available to the general public [34, 35].

2.4 Novelty of Modular PMV

To the author's knowledge, there has been no explicit attempt to apply MDA in an autonomous PMV. The concept of intelligent PMV has been previously reported, but there were clear indications that the vehicle's interiors were altered for ADS integration [11, 10]. Researchers from the University of Electro-Communications, Japan, have chosen to modify the steering shaft mechanism by adding a belt pulley that connects to the motor output, resulting in the CARTIS TypeR.

Meanwhile, developers of the Self-Driving Mobility Scooter from the National University of Singapore opted to entirely disable the original steering column, repurposing it for a customized monitor display. One could argue that only slight internal modifications were done, but this is still against the modular philosophy aimed for the author's dissertation.

2.5 Path Planning Technique

Path Planning (PP) is a subset of motion planning methods, where its main task is to provide a direction for the vehicle to travel via a set of consecutive waypoints. ADS motion planning can be mainly divided (but not limited) into several layers,

namely route planning, threat assessment, decision making, candidate solution generation, and path deformation[36]. Path planning can fall under route planning and candidate solution generation. The path needs to be traversable, meaning that nothing should block the vehicle. Latombe [37] in his book has described the component of PP into the following:

- $A \subset W$: A single rigid body moving object, A, is a part of the world, W. W is denoted as the Euclidean space $\mathbb{R}^n$, where n is the dimension of the space. items $O \subset W$: The obstacles in W. It can be both a stationary and moving object.
- The state of A and O is known a priori.
- The position of O in W is explicitly known.

It is common to divide PP problems into two: global and local. Latombe's description does not explicitly divide these two. However, the following implication can be made. First, the global start and goal need to be known so that the planning algorithm in the vehicle knows where to head. It needs to find a suitable path to a destination, considering its current position. For example, a person is at his home. His home is his "current position". Then he decides to go to the supermarket. The supermarket is the place he wants to go, i.e., his "goal" or "destination". In a PP case, the planner needs to know about the goal. The planner also needs to explicitly know about the position of all landmarks in the area. The information on the landmarks is obtained from a map. Navigation without a map is like walking in the desert, where nothing can be used as the location indicator. With the map being available, the planner will finally compute a traversable path to the goal without colliding with any landmark. During this computation, the presence of

moving objects or blockage is not considered.

Once the global path is obtained, the vehicle will start moving to the goal. The path tracking algorithm generates the required motion by computing the necessary control inputs, for example, speed, so that the vehicle can move along the path. However, the actual world is ever-changing. So during the travel, the vehicle might meet some blockage that hinders its operation, such as a construction site. Therefore, the vehicle needs to compute a new trajectory that will allow it to continue heading towards the goal. The trajectory will allow the vehicle to move past the blockage by computing an avoidance route. This situation is known as local planning. Local planning establishes a temporary solution based on the time and location of the vehicle.

The basic definitions need to be clarified before proceeding. A path is defined as a sequence of points in free space, while a trajectory is a sequence of spatiotemporal states that varies based on the current environment condition [36]. A path is commonly associated with the global planner since the goal remains the same throughout the journey. Contrastingly the trajectory is often correlated to local planning because its execution and results depend on the environment's current situation and the vehicle's position.

2.5.1 Global path planning

The most commonly utilized global PP algorithm is the graph-based algorithm, such as Dijkstra and A* [13, 38, 39, 40]. These algorithms have the benefits of

allowing the vehicle to find the shortest route to arrive at the goal. They utilized a 2D grid map that describes the position of the landmark and object in the space. A common drawback of the graph-based algorithm is that they do not consider the vehicle's proximity to the landmark. Hence, its results tend to place the vehicle very near to them. However, these can be solved by precomputing the obstacle map, i.e., a reference map that considers landmark proximity such as the Voronoi diagram[13].

2.5.2 Local path planning

Aside from computing the global path, the graph-based algorithm can also be used for local trajectory planning, since in some cases, it is necessary to avoid an obstacle in the shortest amount of time. Another standard method is to employ the artificial potential field (APF) algorithm. The benefit of the algorithm is that it creates a force field that encapsulates the obstacle. During deployment, the vehicle will avoid going into the trajectory. An example of the use case is to avoid vehicles going into human personal space[16]. The APF is unsuitable for global planning since it is vulnerable to the local minima trap, especially in a large environment. The APF algorithm will be further explained in chapter 4.

2.5.3 Challenging situation: Glass Wall

Mapping algorithms are sometimes unable to provide an accurate description of the environment. Sensor's limitation commonly causes this. For example, it is widely known that LiDAR's laser will pass through any transparent material. Therefore

despite having a wall made of glass, the wall might not be present in the generated map. In a recent study by Titebu et al. [41], it found that when a laser scan passes through a transparent wall, the result shows that there is a variation of the neighboring distance and intensity scan as if it is "noisy". Then, by comparing the standard deviation of the scan with and without glass walls, the glass detection method is realized. A more straightforward approach is by detecting the glass frame. For example, Wang et al.[42] has attempted to detect windows by employing the Principal Component Analysis method to compute surface normal based on the building's facade.

2.5.4 Criteria for choosing APF and A* for testing the algorithm

APF and A* are both conventional PP methods that are simple to be implemented. There are other more advanced planners suitable for PP, such as hybrid A* and et cetera. However, these planner requires several layers of computationally demanding processing before it can be used. For example, the hybrid A*, the algorithm needs to cluster the points in a map, then compute the Voronoi Diagram[13]. Both of these require a high-performance system since multiple iterations are needed to get optimized results.

As previously mentioned, a map needs to be available for the planner to plan a path. However, if the map is unable to provide an accurate description of the environment, the planner will generate a non-feasible path. Since APF can generate a force field that encapsulates the landmark, the feature can be leveraged in finding a feasible path globally. In this study, the utilization of APF to find the global path is investigated. The result will be compared with the conventional

A* algorithm. The reason A* is chosen is due to its well-known performance in pathfinding tasks.

2.6 Conclusion

In this chapter, the previous works that serves as a fundamental understanding of this thesis is presented. There are mainly two types of system and design architecture in ADS. They are given as standalone – connected and modular – end-to-end, respectively. Based on the literature and also considerations from the theoretical point of view, the standalone system with MDA was selected as the basis to design the SMAD-PMV. The PP problem and the motivation behind the proposed method is also discussed. In the next chapter, the study will propose the framework of a Standalone Modular Automated Driving Personal Mobility Vehicle (SMAD-PMV) that implements the standalone, modular approach for an autonomous PMV.

CHAPTER 3
DESIGN AND DEPLOYMENT OF STANDALONE-MODULAR AUTOMATED DRIVING PERSONAL MOBILITY VEHICLE

3.1 Introduction

This section elaborates on the physical structure of the SMAD-PMV prototype. An intelligent vehicle is expected to function as close to vehicles operated by real humans. Therefore, the PMV's hardware can be figuratively compared to that of a human's. In this sense, the computing system serves as the "brain" that perceives, thinks, and gives instructions, while the sensors bestow "sight" and "hearing" abilities, providing necessary information for this "brain". Lastly, the "body" and "limbs" are the four-wheel PMV and steering mechanism, respectively, executing the autopilot commands from the computing system.

Next, the contribution of this chapter towards completing this thesis is highlighted, and generally for advancing this research field. Some of the previous studies discussed in Chapter 2 claim that the system is modular. However, they have made some modifications that are difficult to assemble and disassemble. The SMAD-PMV prototype does not require much change, except for swapping the original vehicle's front basket and mirror slot with the proposed mechanism. Swapping does not consider to be an invasive modification since the vacant space is used as it is. This decision principle allows the vehicle to achieve full modularity. The design and contribution of the SMAD-PMV are summarized as follows.

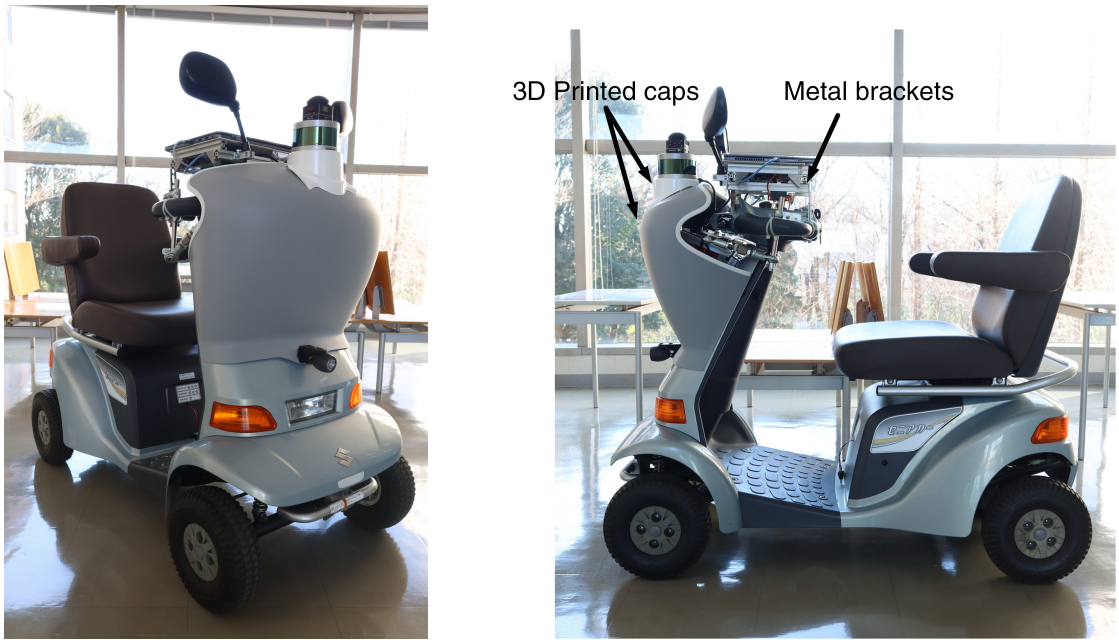
- For the hardware MDA, the original mechanics of the PMV are separated from two sensors and also from the main computer utilized for data collection and computation, respectively.
- For the software MDA, the author highlights a particular feature: the region-based planning strategy (RBPS) designed to solve predetermined virtual areas locally.
- The SMAD-PMV was tested in two main environments, outdoor and indoor. A sub-case for the wall issue was also studied.

3.1.1 SMAD-PMV Prototype

Figure 3.1. The base model, Suzuki Senior Car ET4D as advertised and commercially available[1].

The base PMV serving as "body" is a 4-wheeled direct-drive mobility scooter, the Suzuki Model ET4D, also known as the Senior Car in Japan. It was advertised and available commercially in the form shown in Figure 3.1. From this figure, it can be seen that the Senior Car originally came with an external basket attached as the

front storage compartment. It was removed to allow the installation of the ADS mechanism. As the basket itself was detachable without any internal modifications, in addition to the PMV being completely operable after its removal, this procedure is considered to be minimally invasive. Thus, the hardware modularity is retained. Figure 3.2 shows the implemented SMAD-PMV prototype.



(a) Isometric view. (b) Side view.

Figure 3.2. The SMAD-PMV prototype showing 3D printed caps and metal bracket.

Following that, a metal bracket with a platform was installed on the vacant screw ports, functioning as sensor placements. Another metal platform was retrofitted onto the same port as that of the side mirror, which holds the computing system, in this case, a personal computer (PC). If the computing system is miniaturized adequately by means of single-board computers and smaller sensors are used, these two supporting structures can be entirely eliminated. Finally, a

3D-printed cap made of polylactic acid (PLA) was attached, covering the front extrusion. This cap was made for an aesthetically pleasing exterior and bore no other significant purpose.

The "limbs" of an AV are the steering mechanism that allows the self-driving mode. The SMAD-PMV consists of two parts, the steering mechanism, and a foldable platform. Figure 3.3 shows the three-dimensional geometry showing the SMAD-PMV's and the mechanism's rotational points. The steering mechanism platform has four screws that are attached to the body. The vacant screw hole originally belongs to the PMV's basket. Compared to the original capacity of the basket, which is up to 5 kg [1], the total weight of SMAD-PMV's ADS mechanism is 4.78 kg, including the sensors.

To install the system, first, the mount is attached to the PMV's body, then the socket-type screws are used to fix it. Figure 3.4 shows where the screws are positioned.

Next, the handle holder is attached to the steering handle. Then, the rod ends which are marked as ③s in Figure 3.3 are inserted onto the handle's holder. The location is shown in Figure 3.5. The setup is designed specifically for Suzuki Model ET4D in mind. However, if the system is transferred to another brand or model, as long as the PMV is the front-handle scooter type, it is possible to design a conversion bracket to mount the SMAD-PMV mechanism.

During automatic driving mode, when the servo motor (which is marked as

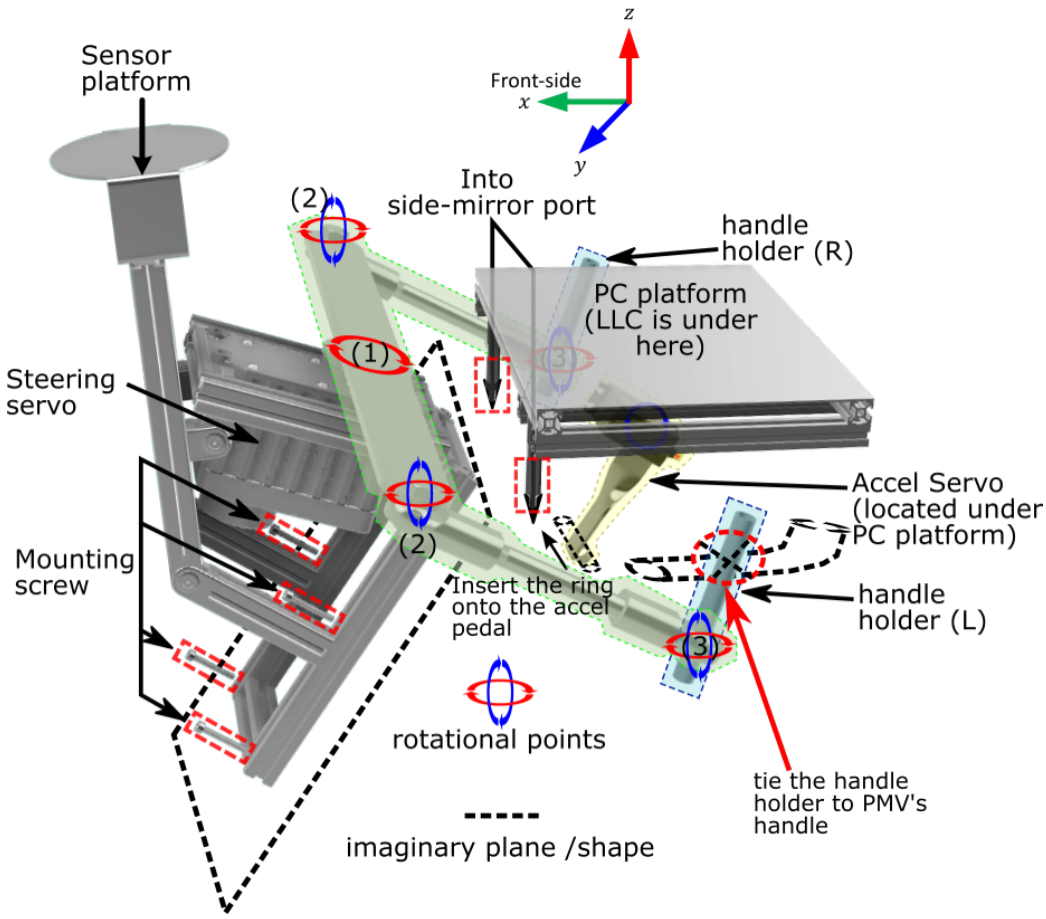


Figure 3.3. Three-dimensional geometry of the retrofitted mount showing sensors platform and PC platforms, while the steering mechanism is highlighted in light green color. The five points (one ①, two ②s and 2 ③s) indicate locations in which the steering mechanism has degrees of rotational freedom.

①) rotates, the entire long plate on the mechanism will turn, pulling point ② together. As point ② turns, it pulls point ③,causing them to move together, therefore completing the automated steering motion. It takes about 10 minutes to install and 6 minutes to remove all of the mechanisms in Fig 3.3. However, one only needs to disconnect both of the points ③ from the handle holder to achieve manual driving. For the steering servo, a 12 VDC high torque servo is utilized.

Figure 3.4. The front look of the vehicle where the system is mounted.

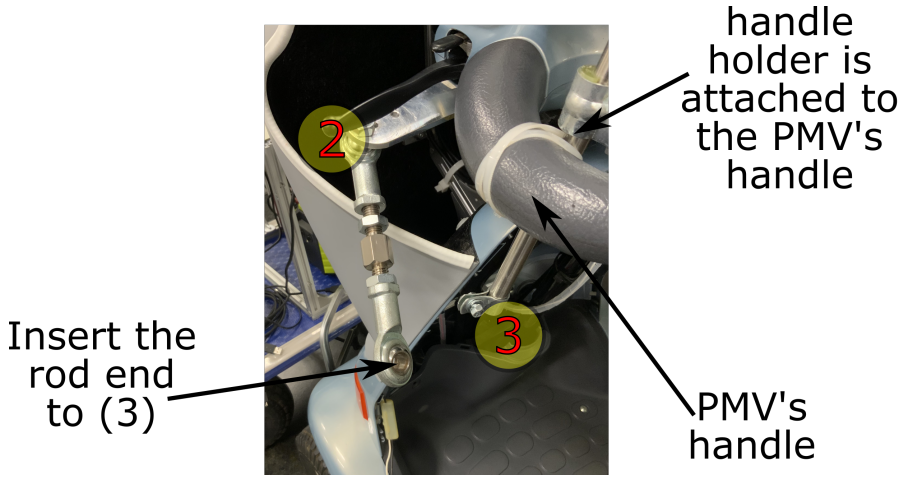


Figure 3.5. Installation of handle's holder.

The servo can perform a 120-degree rotation in one second.

Servo motor is also used to control the acceleration pedal, and it is placed under the PC platform. A plate is attached to the servo to connect it to the pedal. Figure 3.6 shows the actual installation of the accel servo. The PMV will move or stop depending on the pedal's position (pressed or depressed). When the

servo rotates in a counterclockwise direction, it will press the pedal down until the desired speed is achieved. The servo rotates in the clockwise direction to stop the vehicle, resulting in it depressing the pedal up.

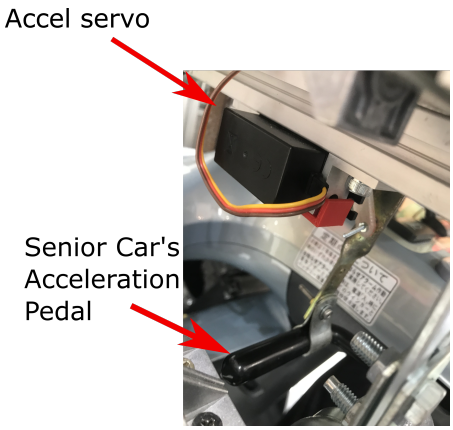


Figure 3.6. The installation of Accel Servo at the acceleration pedal.

To interpret computed control value to servo value, the Arduino Mega was utilized. This setup is called the low-level controller (LLC). Figure 3.7 shows the LLC setup, which is placed directly under the PC platform.

3.1.2 Sensors, Power, and Computing System

The "brain" of the automated driving vehicle is the computing system. There is no standardized configuration, and usually, opting for state-of-the-art hardware will reduce the burden of solving computing-intensive problems. Powerful computers also enable the interpretation of massive sensing data. The review paper by Shi et al. [43] gives more details of the computing system used by the current modern autonomous cars in depth. For this study, a single ASUS ZenBook Flip 14 UX461 [44] laptop is utilized to perform all computation during the automatic driving

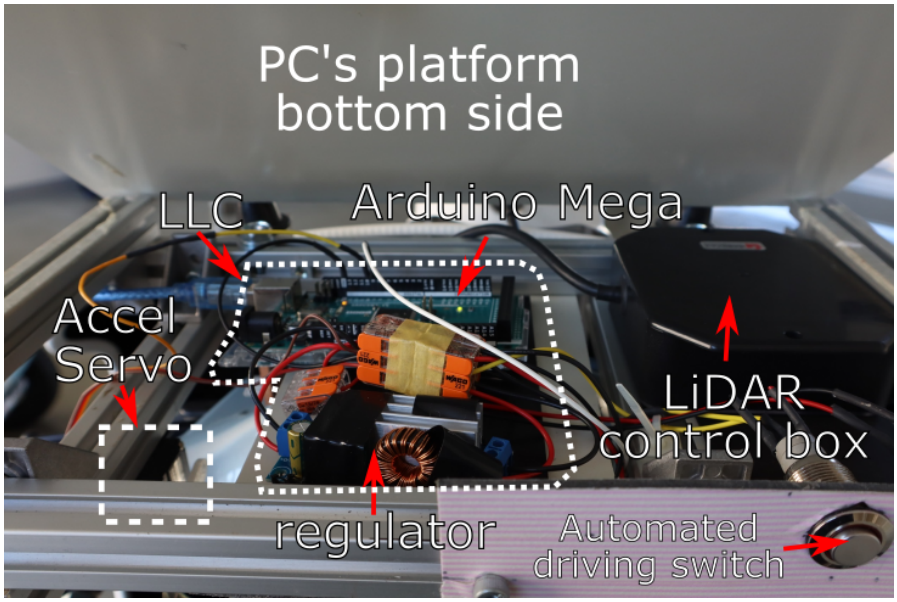


Figure 3.7. LLC is positioned under the PC's platform.

mode. However, the author needs to note that any low computational power system such as laptops or single-board computers such as Raspberry Pi can be utilized. In the study's case, the laptop was chosen due to the required features (i.e., graphical processing unit, monitor, keyboard, et cetera) being conveniently integrated into a single device.

In addition, power restrictions do not permit large hardware installation for a small-scaled vehicle, such as scooters and wheelchairs. These limitations do not hinder the operation of a larger vehicle. For example, an alternator constantly charges the internal combustion car battery, and the electric vehicle's battery stores an even larger electrical charge capacity. These functions allow the installation of high-performance systems that can work for extended periods. On the other hand, a personal mobility vehicle (PMV) can only work in a short amount of time[1]. Moreover, these models did not have self-recharging abilities. If the power is taken

directly from their batteries, the vehicle's own operation time will be diminished. Therefore, the system in this study uses an external battery as an independent power supply for the ADS mechanism to solve this issue.

Sensors are the "eyes" of the ADS since it needs to "see" and "interpret" the environment. The perception modules, which consist of various algorithms, analyze the sensors' data and build the environment's model. A typical ADS is usually well equipped with various sensor suites, including light detection and ranging (LiDAR), millimeter-wave RADAR, ultrasonic sensors, global positioning units (GPS), gyroscopes, and accelerometers et cetera. Larger vehicles require 360-degree coverage with these sensors since slight mistakes might be fatal[45]. For a great discussion about the sensor setup, [46] has provided their long-term work result. Their setup is set to cover various scenarios, and each sensor is assigned its task. However, for a low-speed vehicle such as PMV, such a sophisticated setup is not possible.

Since installing multiple sensors require sophisticated hardware for processing, space constraint also affect the selection of sensors, aside from the power restriction. For the SMAD-PMV, two sensors are utilized, namely the 16-layer mechanical LiDAR from Velodyne Lidar[47] and an omnidirectional camera from Eastman Kodak Company[48]. LiDAR can be utilized for obstacle detection and localization since it can provide accurate distance information. However, LiDAR cannot provide semantic information on an object, for example, light signal color. Therefore, the omnidirectional camera is used for object recognition. However, processing these sensors' data still requires many computation resources. To cater

to the computation issue, the author will employ RBPS, which will be discussed later in Section 3.2.3.

3.1.3 Controller's hardware interface

Figure 3.8 illustrates the interface between the computing system and the other hardware. The Arduino Mega and omnidirectional camera are connected to the laptop through Universal Serial Bus (USB) port. The connection allows both power and data transfer. Except for these two, all the hardware is powered up by a 13,500 mAh mobile battery, which can operate up to 3 to 4 hours during the SMAD-PMV deployment. In between the battery source and both servo motors, voltage regulars are included. These voltage regulators help to protect the motor from any sudden power surge. Especially for the accel servo, it reduces the voltage source from 12 V since the operating voltage of the servo is 5V. The signal of both servos is connected to the Arduino. The LiDAR is connected to the PC via TCP/IP. Since the laptop does not have a TCP/IP port, a converter is used to connect it through the USB-Type C port.

3.2 Software Architecture

The hardware framework specified in the previous section would be meaningless without a solid base of the software underlying the ADS, which will be elaborated on next. The software framework used in the SMAD-PMV provides three fundamental phases, namely mapping and localization, motion planning, and controller, where these phases may or may not be happening sequentially. The outline of the fundamental phases and their assigned processes are shown in Figure 3.9.

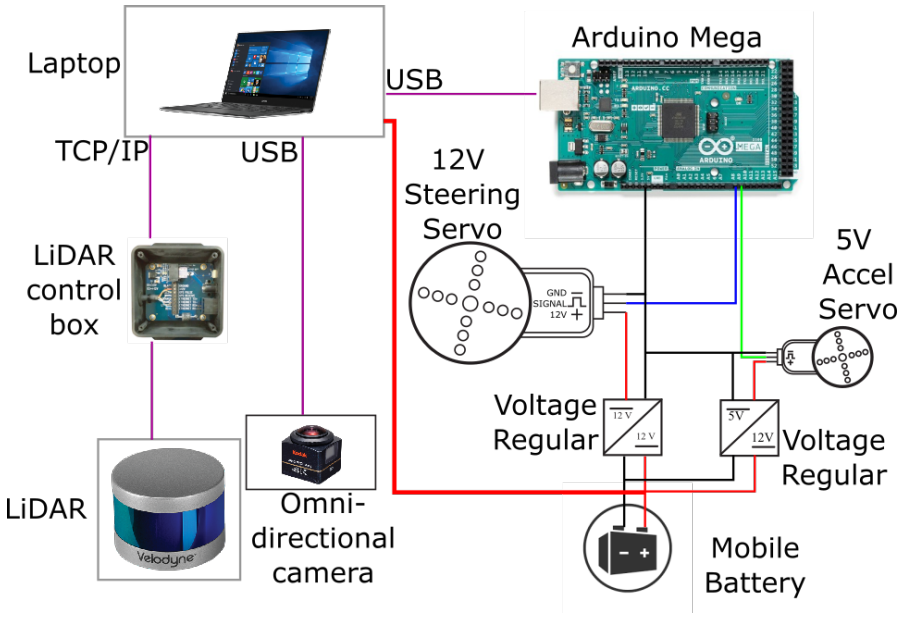


Figure 3.8. The diagram of the connection between the computer and the LLC.

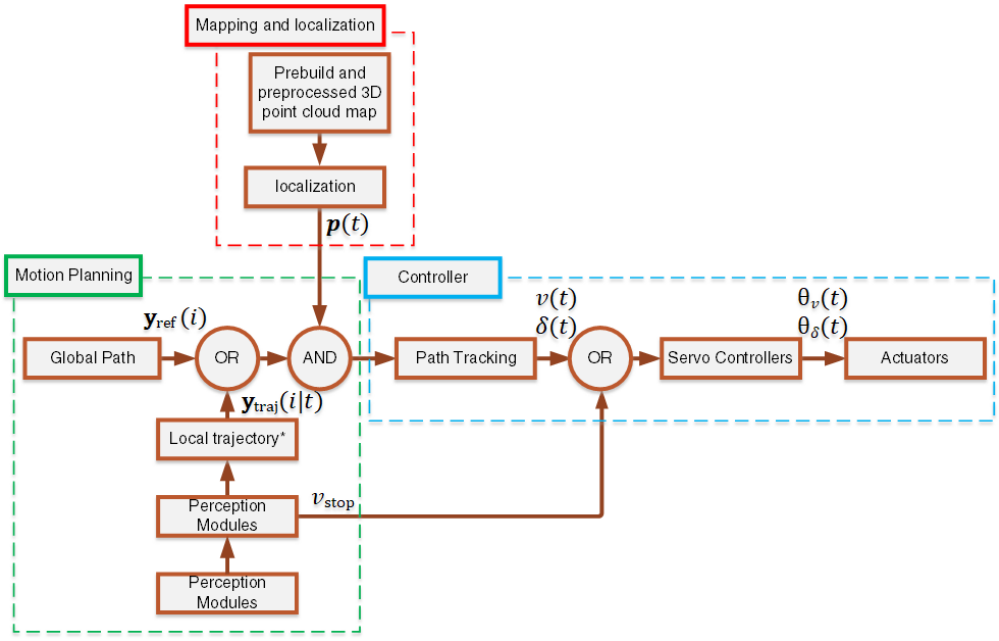


Figure 3.9. Control Flow. The module is broadly categorize into three, i.e. mapping and localization, motion planning and controller.

where

- $y_{ref}(i) = [x_i, y_i, z_i]$ is the predefined global path.
- $y_{traj}(i|t)$ is the generated trajectory based of behaviour planning's decision at time t.
- $\mathbf{p}(t)$ is the position of SMAD-PMV at time, t.
- $v(t)$ and v_{stop} is velocity at time t and emergency stop ($v_{stop} = 0$) respectively.
- $\delta(t)$ is the steering angle at time t.
- $\theta_\delta(t)$ is the steering servo angle at time, t.
- $\theta_v(t)$ is the accel servo angle at time, t.

From this figure, a few pivotal points can be summarized, notably on behavior planning. On the right side is the "Controller" phase, where the physical driving commands are executed. The commands are affected by ADS behaviors, which are often predetermined and solely based on data provided by the perception module in the "Motion Planning" phase on the bottom-left side. Therefore, it is crucial to define proper behaviors in advance to ensure a safe driving experience. Specific locations in the point cloud map generated during the "Mapping and Localization" phase on the top-left side also determine these behaviors. The three main phases shown in the dashed boxes will be explained in detail in the following subsections.

3.2.1 Mapping and Localization

A map is one of the key features of ADS that enable mobile robots and driverless technologies to operate autonomously. The map allows ADS to localize itself, plan

feasible travel paths, and understand its operational domain. Firstly, the 3D point cloud map is pre-built and preprocessed in the mapping and localization phase, followed by the localization process. The pre-built map, which consists of spatial yet dense 3D points, represents a reconstruction of the ADS' operational domain to the real-world environment. LiDAR sensor is used to build the 3D map because it can provide distance information of the static objects (also known as landmarks) in the environment with high accuracy.

LiDAR initially makes two consecutive temporal scans of the landmarks in the same operational domain, given $S(t)$ and $S(t-1)$. A mapping algorithm then compares the two scans and matches them by computing the necessary transformation. The process is known as scan matching, and there are two ways to perform this; real-time or preprocessed offline. However, it is difficult to obtain the map in real-time due to the process being prone to accumulated errors with the increasing course of time[49]. Furthermore, the computation will be heavier, and the operational domain size gets wider as more time passes. Therefore, this study employs the second option, which is to preprocess the point cloud map offline.

After defining the point cloud map, the ADS immediately begins the localization process. During localization, the ADS will estimate its motion by gradually referencing the map. The mechanism behind localization is based on a similar scan matching principle as that of mapping. However, instead of comparing two consecutive LiDAR scans, the current scan, $S(t)$, is compared with a referenced point cloud map, and the matching probability is calculated. The algorithm then searches for the vehicle's location within a map based on this probability, leveraging

the dense 3D points information of the point cloud map. Two main algorithms perform scan matching, namely the Iterative Closest Point (ICP) or the Normal Distribution Transform (NDT). In this study, NDT has been chosen due to its proven matching performance. The author points the reader to the following paper for further information on the algorithm [50]. It should be pointed out that the localization computation is a low burden process because it only needs to compute the transformation, and no data storage is performed (because the map has already been built).

Last but not least, here, Autoware[51] is utilized for building the point cloud map and localizing the vehicle.

3.2.2 Path planning

In this subsection, the background of the path planning algorithm, the Artificial Potential Field (APF) method, is briefly discussed. APF is a path planning method that is inspired by the motion of electrical charges. The goal and obstacles are set to emit the attractive force and repulsive force, respectively. APF has the benefit of allowing one to penalize the obstacle range, creating a boundary with a minimum size that corresponds to the ADS' platform size.

3.2.3 Region-based planning strategy (RBPS)

ADS must always be prepared to respond to any situation that requires decision-making when the conditions are met or triggered. The decisions will depend on the spatiotemporal behavior of the surrounding, which is predetermined during

the behavior planning module of the motion planning phase. In addition, some behavior is unchanging anywhere. For example, the vehicle needs to stop when something is blocking it. It might also be location-specific, such as to not overtake in narrow spaces. Therefore, if the behavior or the decision is not declared beforehand, the ADS might show an abnormal response when it met an unknown situation.

It is challenging to plan all possible outcomes concerning behavior planning because many of them can be non-linear or unpredictable. An example of this would be a walking person making an abrupt stop right in the middle of the road. Examples of effective methods to generate candidate behavior solutions are FSM[52], Markov models[53], or reinforcement learning[54]. The reader can refer to the following paper for a more comprehensive review [55].

In this study, an RBPS strategy is proposed to handle behavior planning. The strategy works by defining the operational domain as the global area and several specifically chosen regions as its subset. Then, the global decision-making rules and their associate behavior are implemented. After that, separate rules are declared to the subsets based on their location's characteristics. Figure 3.10 shows the process flow of the RBPS.

The perception modules interpret raw sensor data, and their results are sent to the region-based planner. The region-based planner is the behavior planning framework where the RBPS is implemented. The module also takes the output from the NDT matching because it needs the localization data to evaluate whether

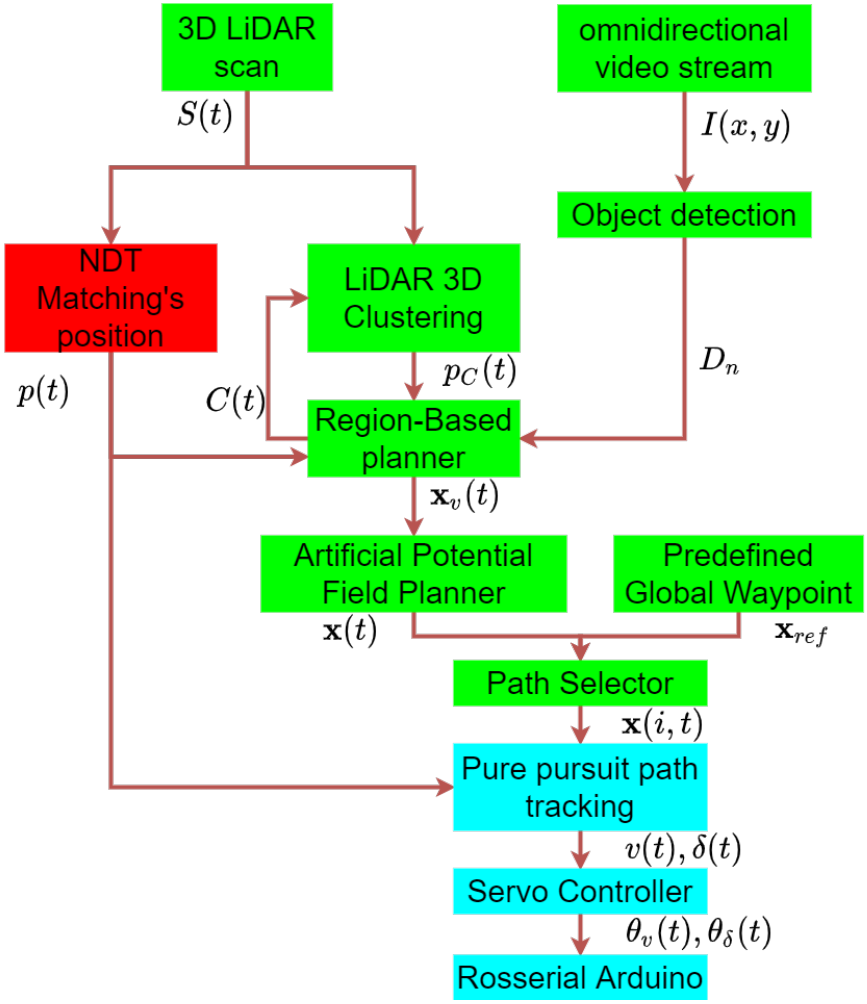
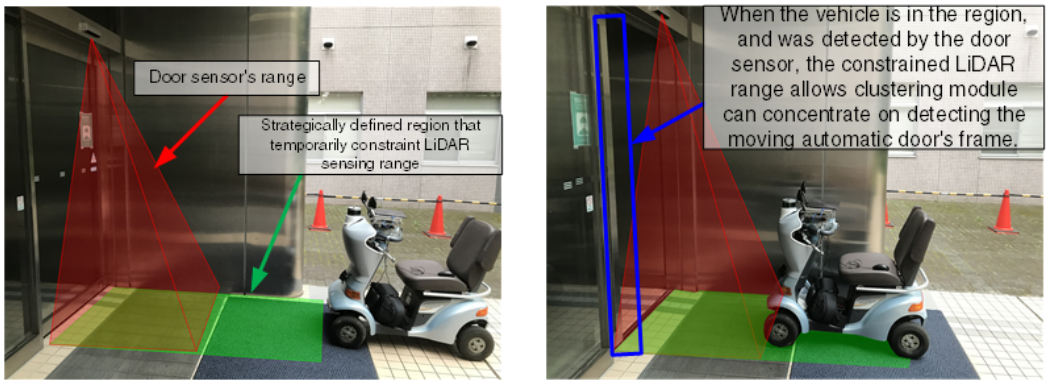


Figure 3.10. The flowchart of RBPS strategy.

the vehicle is in the region or not. Depending on the position and the module, it will return the configuration setting, $C(t)$, or virtual obstacle (VO). An example is as the following Figure 3.11.

In the figure, the ADS attempts to enter the building through the automatic door. Separate rules are explicitly defined to reconfigure the lidar's sensing configuration, $C(t)$. The rule only will be executed when the ADS is in the green region. In this case, LiDAR's Field of View (FoV) is allowed to detect the moving



(a) The behaviour of the SMAD-PMV outside of the detection range does not change.

(b) Inside the area, the region planning stategy allow changes the detection strategy to concentrate more on the door width situation.

Figure 3.11. Illustration of region in front of automatic door.

door frame.

3.2.4 Controller

The controller directs the ADS along the trajectory and translates path tracking output from raw steering angle and vehicle speed into hardware-specific commands. This phase is highly dependent on motion planning and is expressed in terms of physical output, the PMV's movement.

The controller consists of a few modules. First is the path tracking algorithm. When given a set of input(referenced) paths, the path tracking algorithm will compute the necessary steering angle and vehicle velocity. The pure pursuit path tracking algorithm is employed for the SMAD-PMV. Pure pursuit employs a look-ahead point, which is a fixed distance that falls along the referenced path. The steering angle is computed based on the current position, next position, and road curvature. The road curvature is obtained from the computation of the NDT

matching.

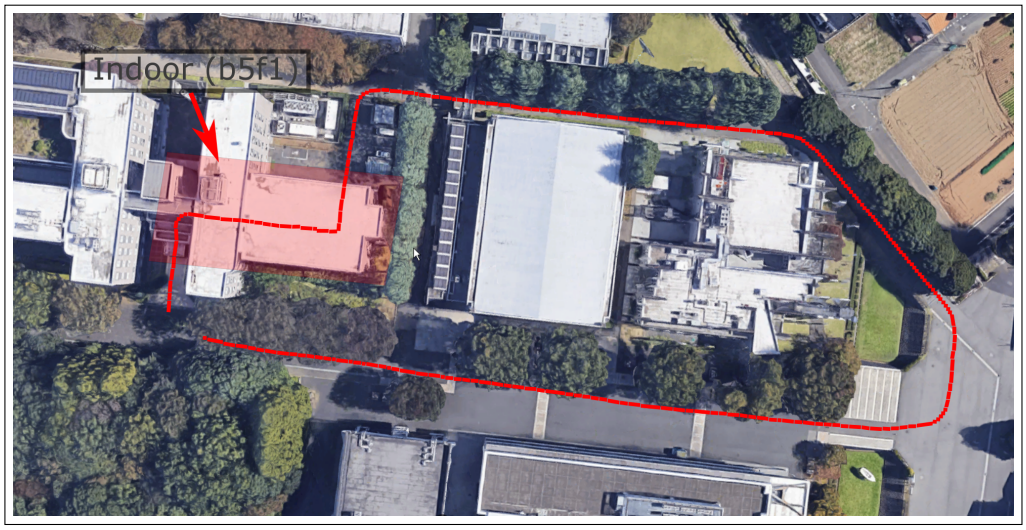
Other methods are commonly used in path tracking include Stanley and model predictive control (MPC). Both of these methods require steering feedback to track accurately, since for Stanley, the tracking error is considered from the front-axle, and for MPC, the model needs closed-loop feedback to perform. In this sense, the pure pursuit has the advantage since the look-ahead distance is computed from the rear axis. Therefore, it does not need the steering feedback.

The pure-pursuit algorithm provides raw steering angle and velocity value. The computed values are not suitable to be directly used since the servo motor takes an angle as the input. Therefore, the servo controller module is employed. The servo controller module takes value from pure pursuit and converts it into a steering servo and accel servo angle.

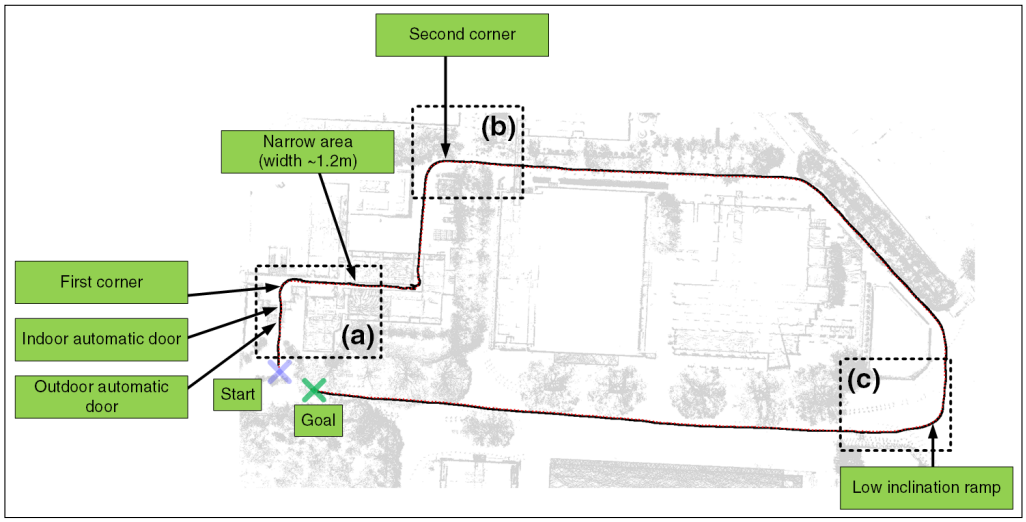
Once the necessary angle is computed, the final phase sends the data to the actual servo motor. Therefore, in this phase, the values are sent from the computer to the LLC. The Arduino in the LLC takes these values and distributes them to each servo.

3.3 Experimental Procedure

In this section, the ADS is equipped with the hardware discussed in the previous Section 3.1.1. The software framework that has been explained in the previous section was also tested in the SMAD-PMV prototype. The validation was done around Shibaura Institute of Technology (SIT), Omiya City, Japan.



(a) The top view of Omiya campus with red line shows the referenced trajectory.



(b) The point cloud map of the campus area with referenced trajectory (red line) and generated trajectory (black line). The region planner can be strategically defined on the automatic door area. The slightly enlarge version of area a, b, and c can be seen in Fig. 3.14(a), 3.14(b) and 3.14(c), respectively.

Figure 3.12. Illustration of deployment map with the top view from google map.

The image in Figure 3.12(a) (as seen from Google Maps) depicts the operational domain. The predefined global path shown in the red line is approximately 500 meter in distance and consists of indoor and outdoor operations.

The test is conducted during the daytime, after 16:00 hrs Japan Standard Time (JST). The moving speed is constant at 3.6 km/h, within the average speed range of human gait[56]. Based on several observations, the look-ahead distance value for pure-pursuit must be set to $\geq$1.5 m to get stable steering motion for the PMV in this study. It is found that the shorter the distance value, the more unstable the vehicle becomes, i.e., slalom motion. The more distance, then at the corner, the vehicle might take a turn too early, which might cause accidents if the cornering trajectory is near any landmarks. In the following experiment, the look-ahead distance is fixed to 2 m. As no pedestrians were blocking the vehicle during the test, the dynamic APF path planning result is not reflected.

A map of the campus is needed before the SMAD-PMV can be deployed around the area. Therefore it needs to be built first. Only LiDAR 3D point cloud data is required to build the map. The data is collected by driving the PMV manually around the campus area. Once completed, the 3D map is preprocessed offline using the NDT mapping method. Once the map is obtained, the map is tested by localizing the SMAD-PMV in it. Then the SMAD-PMV is manually operated a few rounds to ensure no large positional error (drift) occurs that might caused the vehicle to lose control later. At the same time, the necessary location for RBPS is manually marked on the map, and the desired configuration is added. It should be noted that the NDT algorithm can be configured in Autoware to allow some tolerance for the positional error. Setting it too high may cause the system to get inaccurate position information, while setting it too low might cause the system to have difficulty localizing itself. For the SMAD-PMV, the system is set to allow 1 m error tolerance.

Once the localization result is satisfied, the referenced trajectory will be recorded. This is done by manually operating the SMAD-PMV and drive accordingly along the desired path from the start point until the goal. The preprocessed 3D point cloud map used for localization with the recorded position of the ADS during deployment is shown in Figure 3.12(b). Based on the recorded path and the preprocessed trajectory, the SMAD-PMV is now ready to be deployed.

3.4 Result and Discussion

This section discusses the result of the SMAD-PMV deployment around SIT. The first section of the deployment requires the SMAD-PMV to enter the indoor area through automatic doors. These two doors are the main entrance of the building. For indication purposes, the author labeled them as the outdoor and the indoor automatic (doors). The vestibule that separates the two doors is approximately 1.5 meters in the distance. The doors and their frame's materials are made out of transparent glass and aluminum, respectively, as previously shown in Figure 3.11. Both doors' sensors have a detection range of 1 meter and take about 1 to 2 seconds for each of them to open completely.

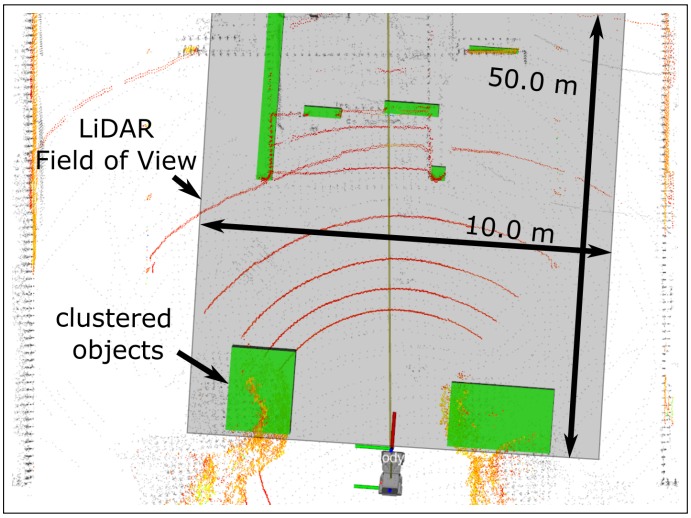
For the SMAD-PMV to enter the building, the system relies on the point cloud clustering module to detect and cluster the door's moving frame. As a result, the vehicle will have to make a complete stop while waiting for the door to be entirely opened. The stopping point is decided based on the relative distance between the door frame's clusters and the front part of SMAD-PMV. Detecting the door frame is crucial because LiDAR's laser beam will bypass any transparent

material. Therefore, it is critical to provide detailed information to the point cloud clustering module to have an accurate clustering result.

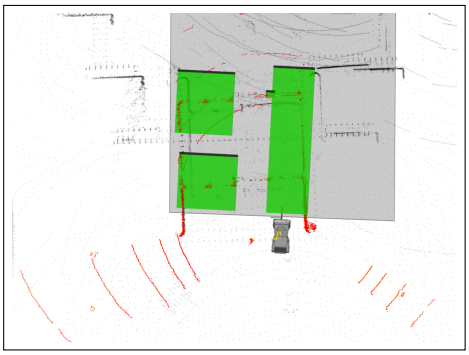
Under global behavior, LiDAR's clustering module will cluster the points located within 50 meters wide and 10 meters in width of LiDAR's Field of View (FoV), as portrayed in Figure. 3.13(a). A wide FoV enables more behavioral decision options in the general environment to be included, considering the spacious area. However, the range is not suitable when the SMAD-PMV is in proximity to the building's entrance; because the wider the FoV, the more points will be grouped, and some supposedly separated landmarks could be mistakenly clustered together. Such a scenario is depicted in Figure 3.13(b). Furthermore, since it is challenging to estimate LiDAR point's density, the author could not provide the optimum configuration of the minimum and the maximum number of points in the cluster.

The clustering issue is mitigated by narrowing down the LiDAR's FoV. First, a local region of 1.5 meters is placed in front of the outdoor automatic door that goes on and extends until 1.5 meters after the indoor automatic door. Then, a predetermined setting reconfigures the LiDAR's FoV while the SMAD-PMV is in this region. The reconfiguration sets the FoV limit to 1.5 meters in length and 0.6 in width, as shown in Figure 3.13(c). This constraint will allow the LiDAR clustering algorithm to cluster the door frame's points correctly, besides allowing some tolerance if any pedestrian is going out from the door. Once the outdoor automatic door is opened, the SMAD-PMV will proceed to the indoor automatic door. Since both doors are sharing a region, the local behavior remained unchanged. As the

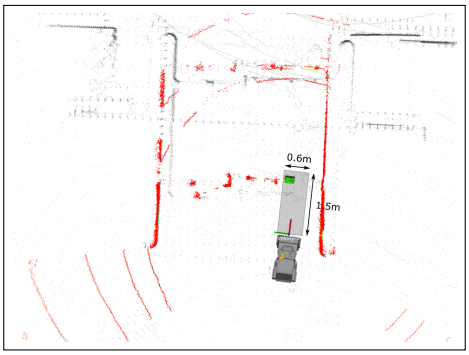
SMAD-PMV exits the shared region, the LiDAR's FoV configuration will return to normal, and any other region-specific behavior, if defined, will be reset.



(a) Outside of the region, the FoV is wider to allow more possible detection. LiDAR clustering FoV (black transparent square), clustered objects (green square), LiDAR raw scan (red/orange points), 3D map (black/grey points)



(b) Inside the automatic door region, the clustering result not able to search for the automatic door frame, due to too clustering clusters everything nearby as well.



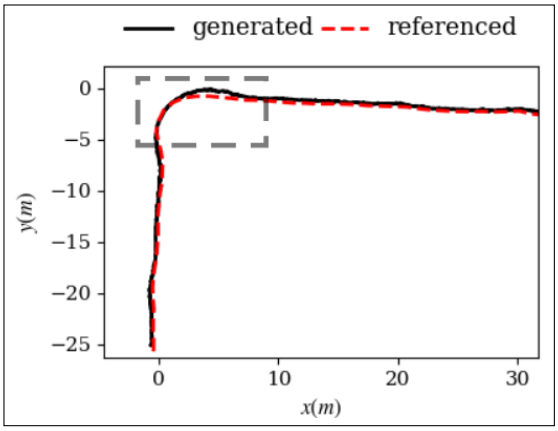
(c) When region is defined, the FoV is much more constrained, allowing the LiDAR clustering module to cluster more specific objects.

Figure 3.13. Result of modifying lidar FoV configuration when the vehicle is in the predefined region.

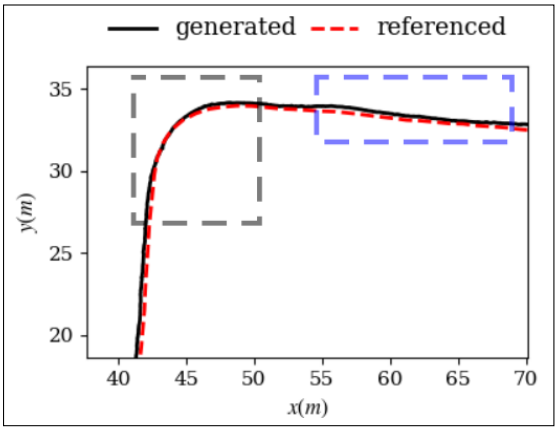
The developed system relies on detecting the moving glass frame during automatic door transition and cannot determine whether the door is currently opened. Therefore, the study needs to assume that the automatic door will always function properly. When the vehicle enters the door region, the frame will move instantaneously, consequently blocking the SMAD-PMV.

The point cloud map only provides the distance particular and not semantic information of the environment. Examples of such information are the location of the indoor and outdoor, automatic doors, the exact type of material of the landmark, etc. RBPS is an example of such a concept, albeit it only covers a specific area of interest. Therefore, if the complete semantic information is available, behavior planning can be more flexible. RBPS can use the information to generate a more specific planning solution locally. However, in the author's point of view, unless the connected system design is adopted, it is still not possible to accurately determine the state of the landmark. For instance, the semantic information does not inform whether the door is currently open or not.

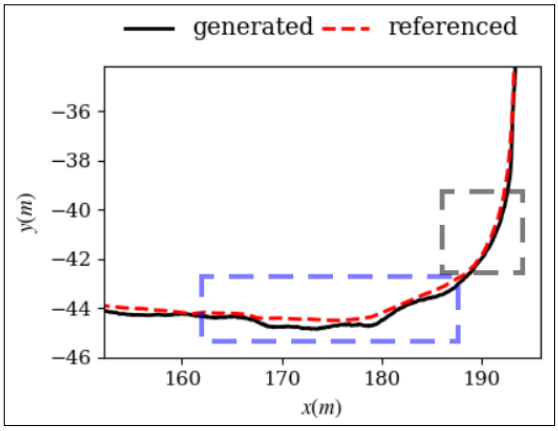
Figure 3.14 shows the operational data of the SMAD-PMV during deployment. It includes referenced and observed trajectory, steering and accel servo data, and relative distance and heading angle changes at every time step. Each time step is equal to 0.1 seconds, which is the default ROS's topic publishing rate. Since there is a massive amount of data, the areas are specifically selected as depicted in Figure 3.12. These areas are chosen to highlight the performance of SMAD-PMV in situations other than moving in a straight line.



(a) There is a large tracking error on the first corner. This is due to the steering mechanism limitation that didn't allow sharp turn.



(b) There are ample place to make a turn in this second corner area so the steering limitation is not that obvious.



(c) There are ample place to make a turn in this small ramp area. Therefore the steering limitation is not that obvious.

Figure 3.14. (a) covers automatic door, first corner and narrow area. (b) covers second corner. (c) covers small ramp.

In the Figures 3.14(a), 3.14(b) and 3.14(c), the red dotted line signify the referenced path while the black line shows the observed trajectory at each time step. Figure 3.14(a) covers the automatic door area, the first corner, and narrow area of the deployment map. It can be observed that the SMAD-PMV has successfully followed the path properly. Since the system relies heavily on LiDAR's

performance, there are slight differences between the referenced and observed trajectories. The differences are due to the localization error caused by the lack of encoder and inertial measurement unit (IMU) to increase path tracking accuracy.

Meanwhile, the rectangular-shaped dashed box shows the first cornering made by the SMAD-PMV. In this area also, there is a slightly larger deviation between both trajectories. The deviation is caused by the mechanism steering limit, which is capped at ± 10 degree. Therefore, it takes some time for the SMAD-PMV to complete the cornering motion. Figures 3.14(b) and 3.14(c) covers the second corner and the low-inclination ramp area of Figure 3.12(b), respectively. Since the referenced path's cornering radius is larger in both cases, the SMAD-PMV has ample time and space to complete the cornering motion. Here, the steering capability is deemed satisfactory, and no obvious limitation is observable.

Figure 3.15 shows the SMAD-PMV generated data at all aforementioned positions. The servo angle speed is constant. The green highlighted box shows the area where the SMAD-PMV makes a full stop to wait for the outdoor and indoor automatic door to open. In all cases, the steering angle displays a jerky response. Even though the computation of the steering angle is up to three floating-point, the servo resolution could not achieve that precision, giving rise to the jerky response and resulting in a jerky angle. However, since the angle change is small, the jerkiness is not felt by the passenger.

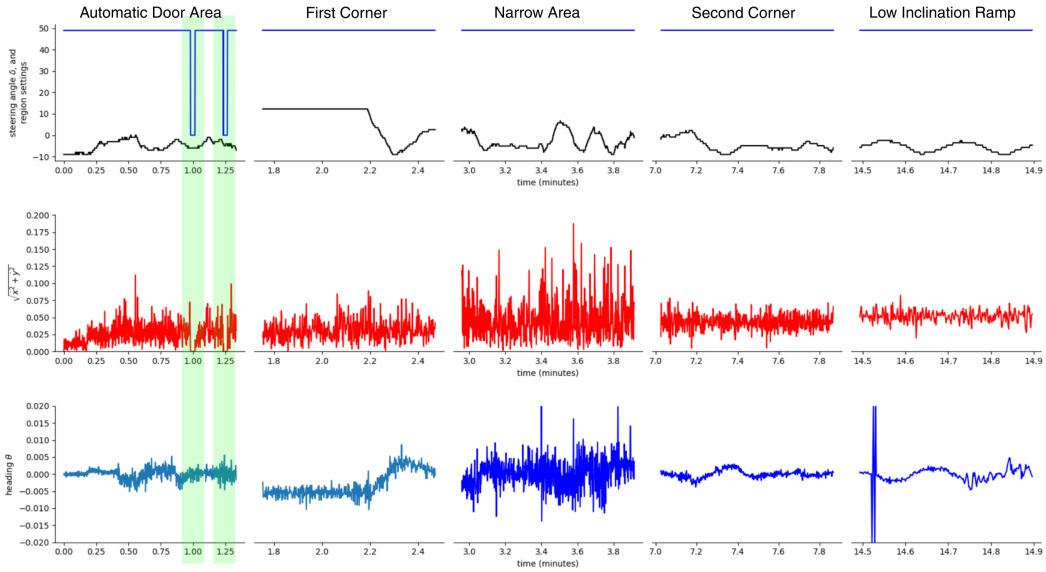


Figure 3.15. The point cloud map of the campus area with referenced trajectory (red line) and generated trajectory (black line). The region planner can be strategically defined on the automatic door area. The green box is where the SMAD-PMV halts while waiting for the automatic door to open.

3.5 Conclusion

In this chapter, an intelligent PMV prototype dubbed SMAD-PMV was developed. The hardware and software of the ADS mechanism were tested in real-time situations, namely a partial indoor and outdoor environment, including when the interface of an automatic glass door between the two environments is involved. Furthermore, the region-based planning strategy is employed whereby the vehicle's configuration changes depending on the vehicle's location.

The main contribution of the mechanism is solving the previously unaddressed problem of the previous studies, which is the modularity of the hardware. The design did not rely on modifying the actual vehicle. Any malfunction of the mechanism will not influence the actual vehicle hardware. For example, if the

motor is broken, the passenger can break the connection between the rod's end and the steering handle and resume traveling using a manual operation. Furthermore, since the steering mechanism only takes 6 minutes to be removed, conversely, the broken steering motor can easily be replaced.

The previous work also did not address the indoor to outdoor transition problem. With the RBPS method, although manual annotation is needed, the method is proven to solve the transition problem effectively.

Although the deployment of SMAD-PMV generally shows satisfactory results, there are some limitations that have been identified. For example, most of the time, the vehicle suffers from localization errors. Since the vehicle relies only on LiDAR for navigation, there is no other sensor available to compensate for correcting the error. However, from the author's point of view, the error is still within an acceptable range since the vehicle can navigate within a constrained environment such as indoor. Another limitation is the referenced path is manually generated before it is used in deployment. Therefore, a path planning method that is feasible needs to be investigated. It will be studied in the next two chapters.

CHAPTER 4
PROPOSED PATH PLANNING METHOD 1: AUGMENTED BY ROTATION MAP

4.1 Introduction

In the previous chapter, under the deployment of SMAD-PMV, the referenced path needs to be generated manually. In this chapter, a simulated study is conducted to find a feasible path planning algorithm for the SMAD-PMV. Since SMAD-PMV is vulnerable to localization error during deployment, it may cause the vehicle to drive very closely to any landmark. It is highly suggested to have a planner that can penalize the distance between the landmark and the SMAD-PMV. If the planner creates a path too near to the wall, accidents are guaranteed to occur. Two path planning method is chosen for the investigation, namely APF and A*. Each method is briefly introduced before one method is chosen with appropriate reasonings.

The contribution of this chapter towards completing this thesis and generally for advancing the research field is given as following. To the author's knowledge, there is no existing study that specifically relates the map's orientation with the result of the path planning. Since SMAD-PMV relies on preprocessing the map, extending this to a 2D map and onwards the obstacle map is possible. This chapter aims to show that it is important to standardize map orientation to get a consistent path planning result.

4.2 Path Planning

4.2.1 Artificial Potential field

The artificial potential field (APF) method is inspired by the electrical charges' concept [57], in which the objects in the configuration space where the vehicle is traveling are presumed to emit potential charges. The goal or target position is assumed to generate an attractive force that pulls the ego vehicle towards it. On the other hand, the obstacle creates a repulsive force the pushes the vehicle away. These forces are calculated in reference to the vehicle's position. These forces will define the vehicle's behavior as it moves towards the goal while avoiding any obstacle. The repulsive force will also prevent the vehicle from "hugging the wall", which is the behavior that some other path planning method suffers [13].

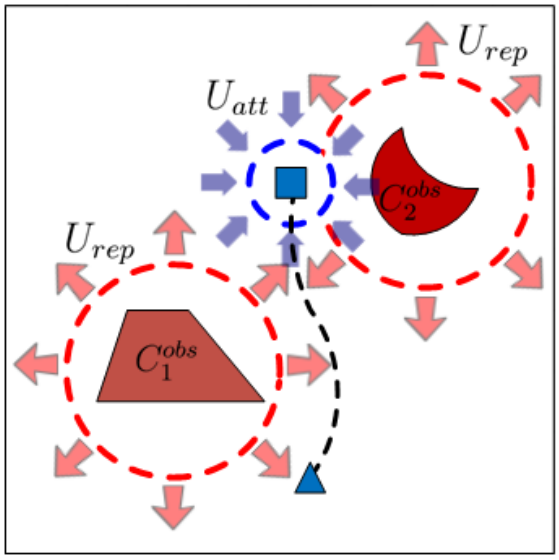


Figure 4.1. Attractive and Repulsive force field.

The total potential is defined as the sum of two potentials, i.e., attractive and repulsive, given as

$$U\left(q\right) \;=\; U_{att}\left(q\right) \;+\; U_{rep}\left(q\right) \tag{4.1}$$

where $U_{att}\left(q\right)$ is the attractive potential generated at the goal position, with respect to each cell in the configuration space, as follows:

$$U_{att}\left(q\right) = \frac{1}{2}\delta \cdot d\left(q, q_g\right) \tag{4.2}$$

where

- $\delta > 0$ is a positive constant that determines the strength of the attractive field, also known as gravitational constant.
- $d\left(q, q_g\right) = \left\|q - q_g\right\|$ is the Euclidean distance between each cell $q := (x, y)$ in the configuration space and the goal position, and, $q_g := \left(x_g, y_g\right)$.

Repulsive potential affects every cell in the configuration space as the following Equation 4.3:

$$U_{rep}\left(q\right) = \begin{cases} \frac{1}{2}\alpha \cdot \frac{1}{d(q,q_{obs})} - \frac{1}{Q^*} & \text{if} \quad d\left(q, q_{obs}\right) \leq Q^* \\ 0 & \text{if} \quad d\left(q, q_{obs}\right) > Q^* \end{cases} \tag{4.3}$$

where

- $\alpha > 0$ is a positive weighting constant of the repulsive field
- $d\left(q, q_{obs}\right) \;=\; \min\limits_{Q' \in Q_{obs}} \left\|Q - Q'\right\|$ represents the smallest distance between the vehicle's current position, $Q = (x_{start}, y_{start})$, and $q_{obs} := Q_{obs} = (x_{obs}, y_{obs})$ is the obstacle position

The repulsive potential is defined as zero outside the distance of the influence of the obstacle (Q^*) and positive when it is inside.

Based on the implementation of theoretical and Equations 4.1 — 4.3, the time taken to calculate the total APF map is given as Table 4.1, where the time to compute the path is excluded.

Table 4.1: Time taken to compute the total APF map.

Grid size	**Time taken to compute APF map**
79 × 76	0.35 s
312 × 299	0.70 s
275 × 140	0.70 s
2460 × 1062	72.0 s

4.2.2 A* Algorithm

A* algorithm (pronounced as A-star) is a graph-based algorithm where weights are assigned over each edge, and search heuristics are employed, i.e., the remaining distance between the current robot position and the goal, to find the feasible path. Examples of the heuristic are Manhattan distance and Euclidean distance. The cost function for the A* algorithm is given as

$$f(n) = h(n) + g(n) \tag{4.4}$$

where $g(n)$ describes the accumulated cost to get the shortest path from starting node to the current node, and $h(n)$, the admissible heuristic or estimation, showing the remaining distance between the current node and the goal. A* algorithm

is often employed to search for the best route during exploration [38] and the research society is constantly proposing its improvement. As A* is a greedy search algorithm, meaning it searches for the lowest weight at each time-step, it sometimes does not guarantee the shortest path.

4.2.2.1 A* as the global planner

A* by itself does not consider the actual distance between the vehicle and the obstacle during path planning. Therefore under normal circumstances, it suffers from "hugging the wall" issues. In order to use A* as the global planner, the map with distance penalty needs to be computed first[13]. The map can be computed by maximizing the penalty distance between empty cells and the occupied cells. Table 4.2 shows the time taken to compute the one-time obstacle map, where the computation time is depending on distance and number of occupied cells.

Table 4.2: Time taken to compute one-time obstacle map (with distance penalty) for A*.

Grid size	**Time taken to compute one- time obstacle map**
312 × 299	±700 s
275 × 140	±1220 s

4.3 Proposed Method for Map Augmentation

4.3.1 Non-augmented vs Augmented

The path planning problem involves solving the task of finding an optimal collision-free path between a starting point and the target point. It also considers the characteristics of the obstacle, for example, the size and shape of the object or

landmarks within a map.

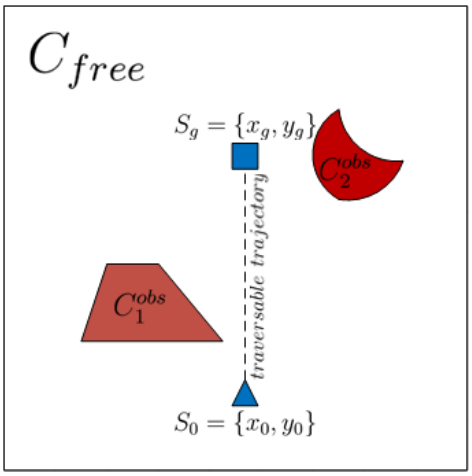


Figure 4.2. Landmarks within the map before being discretized.

Figure 4.2 shows the top view illustration of an $\mathbb{R}^3$ space. $\mathbb{R}^3$ contains two objects, C_1^{obs} and C_2^{obs} that occupy the free space, each assumed as a landmark. A collision-free path will allow the vehicle to traverse along with these landmarks along the traversable trajectory from point S_0 to S_g.

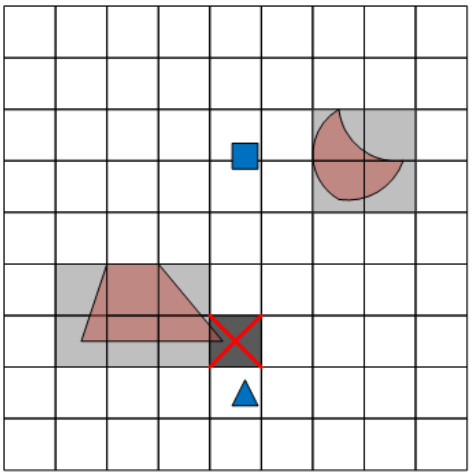


Figure 4.3. Discretized map with the original orientation.

Since the number of the point cloud is huge for every consecutive LiDAR scan, continuous space is discretized. However, this may result in the object's shape being constrained by the pixel that it occupies. For example, in Figure 4.3, the cell is marked as $\times$ has blocked the direct path between the two points.

By transforming the map and discretizing it again, the occupied cell will yield a different shape than the original orientation. Figure 4.4 illustrates a map rotated at an angle of 90° clockwise.

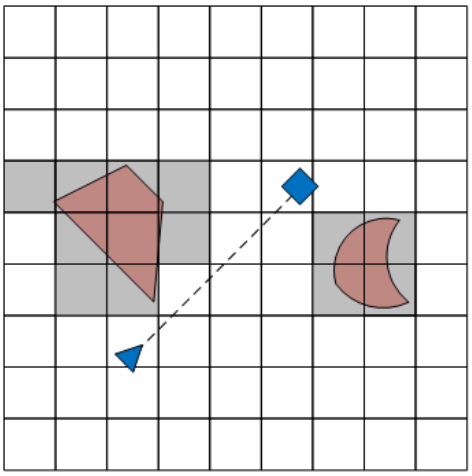


Figure 4.4. Discretized map with the transformation applied.

Note that object C_2^{obs} occupies a 2-by-2 cell, similar to its non-rotated counterpart in Figure 4.3, while the object C_1^{obs} has a different shape. The path between the point S_0 to S_g is also not blocked by any cells. In a discretized space, the generated path may not be smooth. However, it is common for path smoothing to be performed after a path is found [58]. The each points in the discretized map, the rotation is based on the following Eq 4.5

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{4.5}$$

where x and y are the original position, x' and y' are rotated position, $\boldsymbol{R}$ is the rotation 2×2 matrix, and $\boldsymbol{t}$ is the 2×1 translation matrix.

4.4 Experimental Setup

3D point cloud map is first projected into the occupancy grid. Occupancy grid is a 2D spatial representation of the robot's environment, where the traversable path is represented by empty cells while landmark or obstacle is represented by occupied cells (black clour). Although the grids don't have fine-grain information as a point cloud map, nevertheless, they still have the advantage of being computationally efficient [59]. After the map is projected, the total potential value is computed for every grid cells in the occupancy grid using APF.

By transforming the point cloud map into an occupancy grid, a static potential map is generated. The position of ADS is reflected in the 2D transformation. In the author's case, the 2D occupancy is beneficial as it allows us to vary the size of the potential boundary without sacrificing computation time.

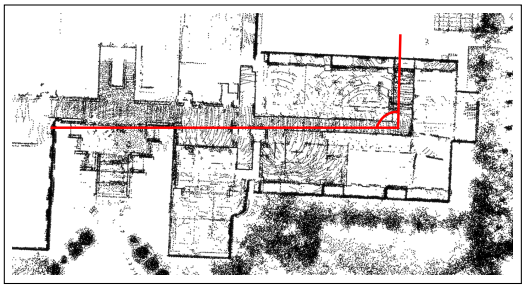
For A* planner, the same occupancy grid map is used to compute the map with distance penalty. For this case, all A* obstacle maps are precomputed, since it takes a huge amount of time to compute them.

4.5 Result and discussion

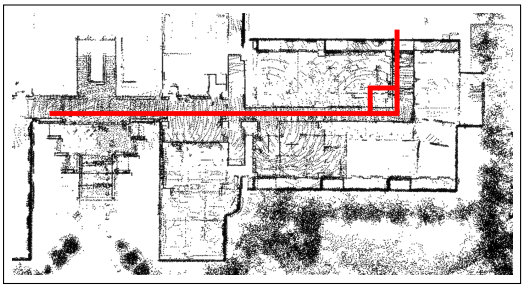
Firstly, since the final map orientation may not be in a right angle orientation, as shown in Figure 4.5(a), the result of the planning will vary depending on the orientation of map, and can be seen in Figure 4.5. The result slightly differs despite using the same setup. In the author's previous work[60], the result of the APF algorithm differs according to the orientation of the map. The test is slightly extended to the A* path planning algorithm on the augmentation case to add more validation. The augmentation is performed by manually rotating the point cloud map after it was preprocessed. The following Fig. 4.5(b) shows the map in its right-angled orientation after rotation.

Following that, Figure 4.6 shows the result of both A* and APF. In augmented map cases, APF and A* show similar trajectory characteristics, as opposed to the non augmented case. The dissimilarity is because, despite a similar map, the pixels are in different shapes. Since both planners attempt to search for the shortest path, they tend to trace the nearby landmark shape. The phenomena are observable in the not augmented map result, where the path is more skewed than the augmented case.

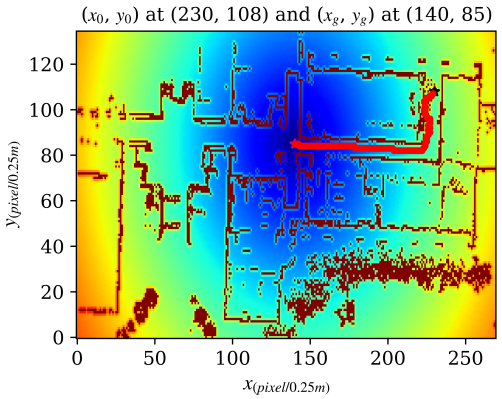
These result is reflected in their original orientation as shown in Figure 4.6(b). It can be seen that the APF under non augmented fails as it makes a curve the hits the nearby object. However, it performs satisfactorily in augmented case and has an almost similar line shape to A*. In A* cases, both paths are feasible.



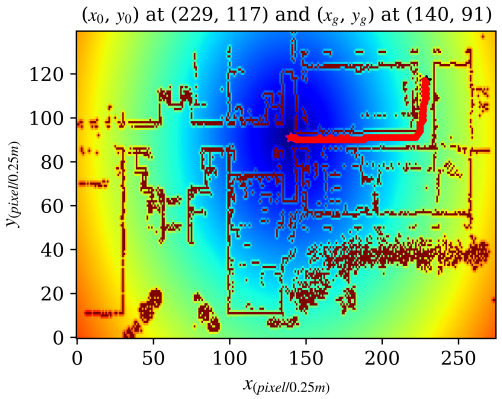
(a) b5f1 in its original orientation



(b) b5f1 after it was augmented



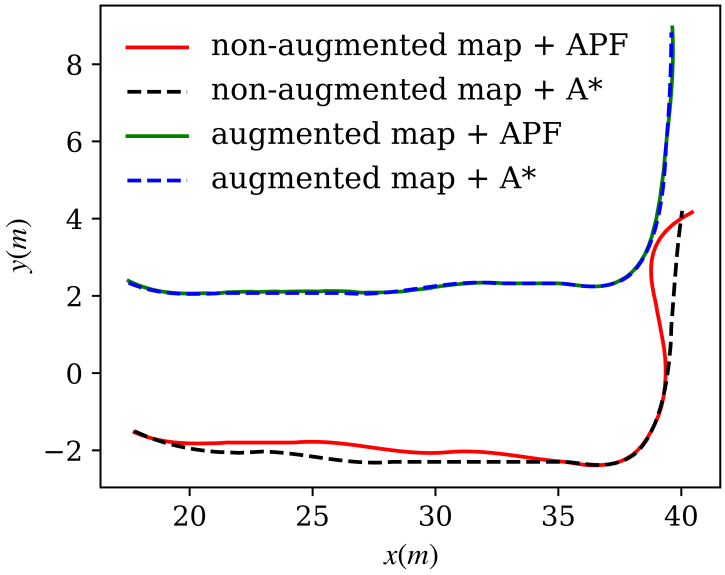
(c) b5f1 path planning result in its original orientation



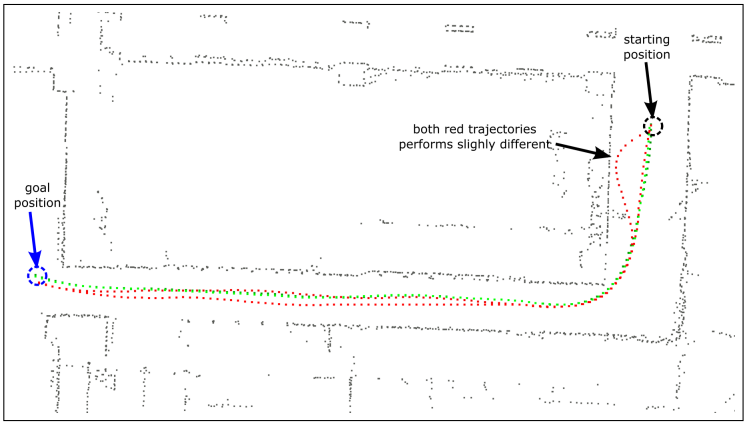
(d) b5f1 path planning result after it was augmented

Figure 4.5. Comparison between non-augmented and augmented map. During map preprocessing phase, it is better to adjust the orientation to allow standardize result.

The following Figure 4.7 shows another example of a non-augmented vs. augmented case. It can be seen in Figure 4.7(a) that the characteristic of the path differs in both dashed box areas. Their location in actual map is highlighted in Figure 4.7(b). Here, it is very obvious that the augmented results perform better compared to non-augmented ones since they didn't hug the wall. When compared between APF and A*, it seems that for the augmented case, APF started by making a curve directly and didn't go near the wall until the end.



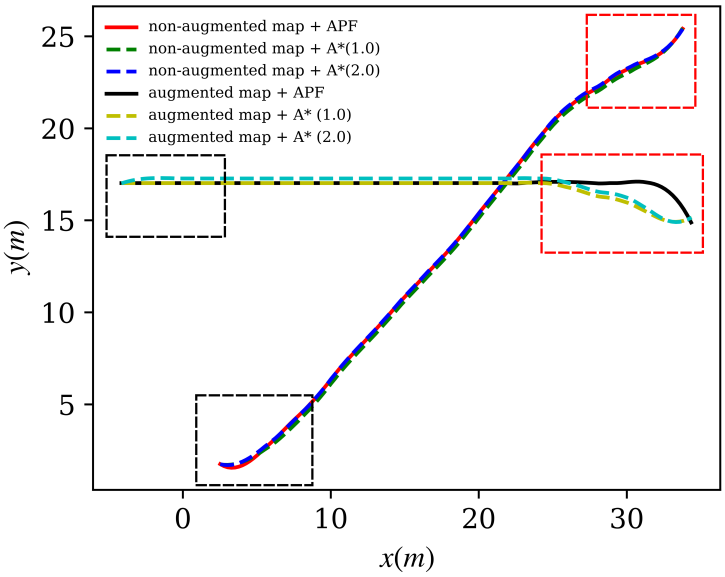
(a) The non augmented and augmented case are tested on A* and APF algorithm



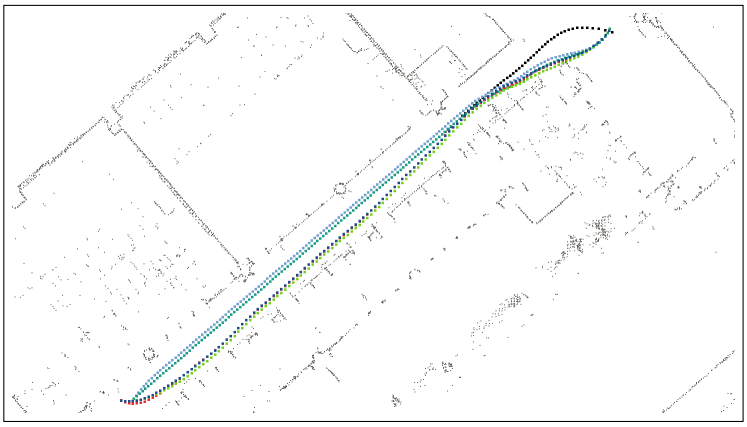
(b) The trajectories from 4.6(a) on point cloud map. Red(non augmented results of both A* and APF*) Green (augmented result of both APF and A*)

Figure 4.6. Comparing the characteristic of path in non-augmented and augmented map using A* and APF.

The augmented result can perform better since most indoor areas or roads are arrange structurally and designed as square or rectangle-shaped. The arrangement affects the final result of the obstacle map or potential map since it will reduce the



(a) The non augmented and augmented case are tested on A* and APF algorithm



(b) The trajectories from 4.7(a) on point cloud map. Red(non augmented results of both A* and APF*) Green (augmented result of both APF and A*)

Figure 4.7. Comparing the characteristic of path in non-augmented and augmented map using A* and APF.

artifacts caused by the discretization of the map. An example of such phenomena can be observed if one tries to draw a straight line on any paint app. When drawing a straight light on right-angled vs. oriented, the result of the oriented will look

like boxes connected by the edges. The right angle will look like boxes connected by the sides.

For unstructured indoor areas or trapezium such as areas, for example, shown in Figure 4.8, augmenting the map only by rotating it is insufficient. It is possible to employ other kinds of augmentation methods to handle these cases, for example, using curvilinear transformation to transform curves to linear equivalent[61]. Another example is distorting the trapezium-shaped area to get the box size equivalent, similar to the technique used to correct lens distortion.

Figure 4.8. Denenchofu areas (coordinate 35.597130232185606, 139.66728981970635), as shown in Google map.

4.6 Summary

In this chapter, the simulated effect of the map's orientation towards the path planning result is studied. Since there was no previous studies available that specifically tested the path planning result in this manner, the results obtained in this chapter's investigations are invaluable in the long run.

Firstly, the standardizing method for map orientation was proposed in all situations. Two indoor locations are chosen for the test. Their 3D point cloud map is divided into two: non-augmented and augmented. This process is performed manually. Then, these maps are projected into a 2D occupancy grid to decrease the computational time. The author arrive at a conclusion that the best orientation is the map that is in right angle orientation. This is because any significant rotation might cause the planner to head towards the nearest landmark due to both methods attempt to find the nearest distance. Based on these results, it is recommended that the 3D map is augmented to the right angle before it is converted into any 2D occupancy grid. In addition, the method may also be used in a 2D occupancy grid "as is".

Regarding selecting the path planning algorithm for SMAD-PMV, the author proposed using APF instead of A*. When the factor of smaller grid and computation time of the map are taken into consideration, the APF performs better than A* in the author's case study. However, further discussion is needed and the topic will be revisited after introducing the APF improvements. The next chapter discusses a modified version of the method, the dynamic APF.

CHAPTER 5
PROPOSED PATH PLANNING METHOD 2: DYNAMIC ARTIFICIAL POTENTIAL FIELD WITH VIRTUAL OBSTACLE

5.1 Introduction

In the previous chapter, the author discusses map orientation, augmentation, and how it affects path planning results. The aim of using APF is to leverage the distance penalty so that even under localization error, the ADS still has ample space for navigating. Furthermore, a map does not always provide accurate information due to the sensor's limitations, this point being briefly touched in the case of automatic doors detection, as explained in Section 3.2.3.

However, the APF method can be time-consuming, especially when dealing with the computation of a dynamically changing environment, because of the necessity to recompute the overall map. Moreover, despite augmenting the map into the right angle orientation helps in finding the feasible path, the planner is still vulnerable to the local minima trap. Therefore, this chapter proposes the dynamic APF map with the virtual obstacle (VO) method to overcome this issue.

As for the contribution point of this chapter, the proposed dynamic APF with VO method is considered much simpler compared to past studies. In the existing literatures, the the overall map needs to be recomputed once the VOs are added. The process of updating the whole map might consume more time. The study aim to eliminate this redundancy by exploiting the previously calculated static APF map, and only considering the affected region for the dynamic APF calculation.

5.2 Dynamic Map

For this matter, the dynamic APF map is introduced, of which its prerequisite is also the calculation of a static APF map, assuming that the goal is not changing. Hence, the time taken for generating the static APF map is crucial for accelerating the overall computation time of the dynamic APF algorithm. The time taken was recorded as 0.7 seconds for generating a static APF map of 321×293 cells by utilizing modern parallel computing vectorization methods. In comparison, the computation using a single processor's core takes 5.6 seconds. Fig. 5.1 illustrates the generation of dynamic APF map and its effect, where the original updated static APF map is updated to reflect the addition of the dynamic map.

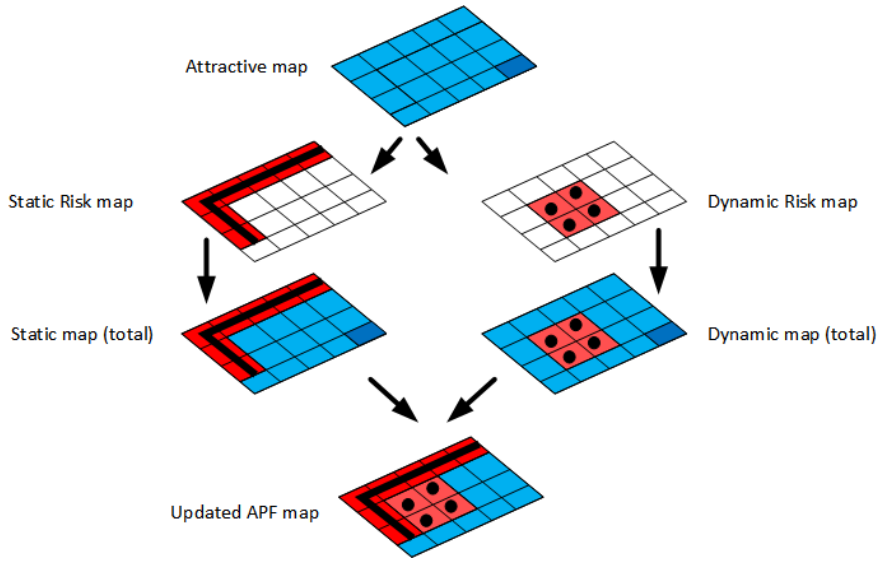


Figure 5.1. Generated dynamic potential map.

From the figure, to compute the dynamic potential map, first, the attractive force from the static obstacle computation is taken. Next, the information of any

obstacle present from the LiDAR current scan is extracted and projected into 2D cells. Then, their repulsive potential is computed and summed up with the already available attractive map as given by (5.1).

$$U_{dyn} \;=\; U_{att}\,(\mathbf{x}) \;+\; U_{rep}\,(\mathbf{x}(t)) \tag{5.1}$$

Finally, the potential map is updated by assuming the following equation (5.2).

$$U_{total}(\mathbf{x}(t)) = \frac{1}{2}(U_{total}(\mathbf{x}) + U_{dyn}(\mathbf{x}(t)) + \left|U_{total}(\mathbf{x}) - U_{dyn}(\mathbf{x}(t))\right|) \tag{5.2}$$

5.3 Virtual Obstacle

However, while searching for a traversable path, the APF with the dynamic map method is still prone to the local minima trap. Therefore, to solve the local minima problem, the dynamic virtual obstacle (VO) point[62, 63] is added by appending it to the dynamic map. The local minima trap is detected by observing the last $\lambda > 0$ number of points in the currently generated trajectory. If there are no differences between the previous λ number of points or if the value is oscillating, the planner will assume that it is now trapped in the local minima. When this scenario occurs, the planner will add a VO at the last location of the currently generated trajectory. After that, the planner will restart the path search again. The process of adding VO will keep on repeating until it is able to escape the local minima. These descriptions are implemented in the system as the following Algorithm 1.

Algorithm 1 Algorithm for using VO method

Input: x_0, x_g, $U_{dyn}(\mathbf{x}(t))$,$U_{total}(\mathbf{x}(t))$
Output: trajectory waypoint

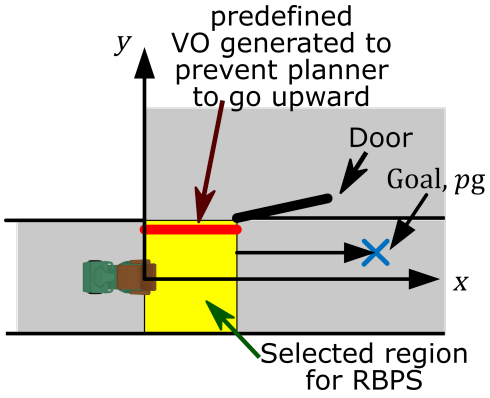
$i = 0$
$iter = 0$
$step = stepsize$
while $\left\|\mathbf{x}_i - \mathbf{x}_g\right\| > thres$ **do**
  if $\|\mathbf{x}_i - \mathbf{x}_{i-1:i-\lambda}\| \leq thres$ **then**
    ${\mathbf{x}_v}^{iter} = \mathbf{x}_i$
    $iter = iter + 1$
    Update $U_{dyn}(\mathbf{x}(t))$
    Update $U_{total}(\mathbf{x}(t))$
  end if
  $\mathbf{x}_{i+1} = \mathbf{x}_i + step \times U_{total}(\mathbf{x}_i(t))$
  $i = i + 1$
end while
return

5.4 Improving dynamic APF with RBPS

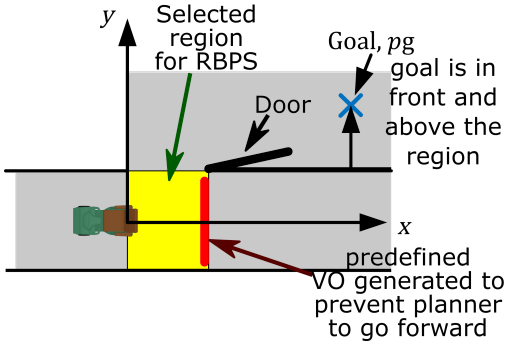
The next examination is to employ the RBPS for adding a predetermined VO. The criteria of adding VO, in this case, is the position of the goal. Figure 5.2 illustrates the example of the scenario. When the goal is located on the top, beyond the door, VOs that prevent the planner from proceeding past the door will be generated. This configuration helps in reducing the number of VOs necessary to prevent the planner from falling into the local minima trap.

The RBPS will also simplify the implementation of location-specific behavior, such as LiDAR's configuration and VO generation. Consequently, the number of global states that the planner needs to check can be reduced. This is because the planner will check on the location and not the states themselves. One example of the possible application is in urban areas, where the traffic light detection

could be enabled only in the traffic light predetermined-region. As a result of turning the detection on and off depending on the occasion, RBPS can reduce the computational burden on the computer. Finally, RBPS also supports modularity since the behavior for other modules can also be defined, such as object detection as previously shown in Chapter 3, Figure 3.10. This relates back to one of the objectives presented in this thesis.



(a) The behaviour of the SMAD-PMV outside of the detection range doesn't change.



(b) Inside the area, the region planning strategy allow changes the detection strategy to concentrate more on the door width situation.

Figure 5.2. Illustration of region generating predefined virtual obstacle point for indoor.

5.5 Experimental Setup

This section discusses the result of the proposed APF with Dynamic Map. It is generally known that a planner plans for a global path from point A to B, followed by planning the trajectory for avoidance. Then only the planner can execute the next task, which is calculating the dynamic APF. Here, two related scenarios are demonstrated; avoidance of a dynamic obstacle in the corridor, and computation

of a new trajectory given a starting position and a goal position. Both tests are independent of each other. The map used in this test is the same area discussed in Section 4.5. As previously mentioned, one of the challenges for this area is that some parts of the wall are made out of transparent glasses, rendering them "invisible" in the point cloud map. Consequently, the obstacle reflected in the 2D occupancy map will also be affected because it heavily relies on the information from the point cloud map.

Three experiments will be discussed. In all experiments, the start and goal are chosen arbitrarily. The description of the experiment set up is as follows

1. Experiment 1. In the experiment, the b2f2 map is utilized. Two scenarios are chosen, computing avoidance trajectory and finding a path when the map does not contain accurate information. The experiment aims to demonstrate dynamic APF and VO ability to search for a feasible path.

2. Experiment 2. The b2f2 and b5f1 maps are used. Two scenarios are discussed; with and without RBPS. After the goal is chosen, the APF will search for a feasible path. Then by using the same goal, position RBPS is placed in the selected locations. The experiment aims to demonstrate RBPS's ability to reduce the number of VO.

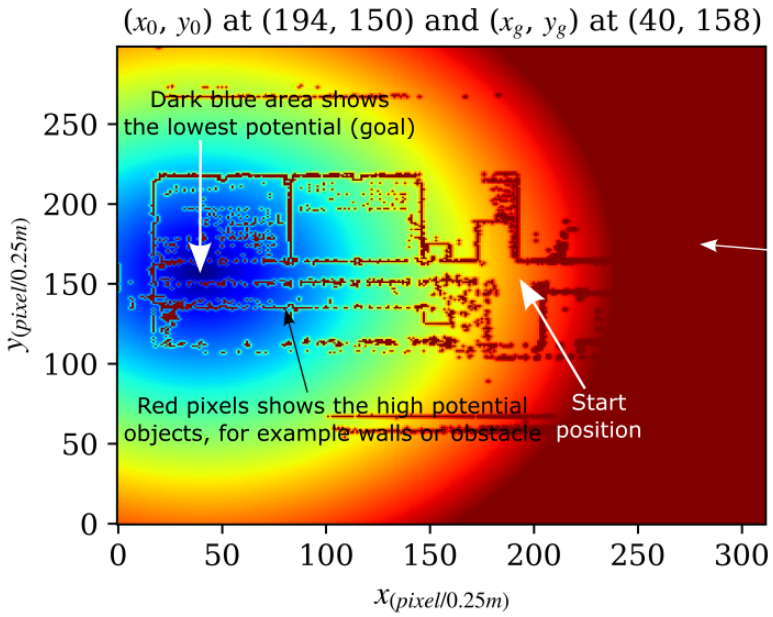
3. Experiment 3. Comparison between APF and A* is revisited. The proposed APF improvement is utilized in the study. The experiment aims to show dynamic APF ability in finding a suitable global path.

5.6 Result and Discussion

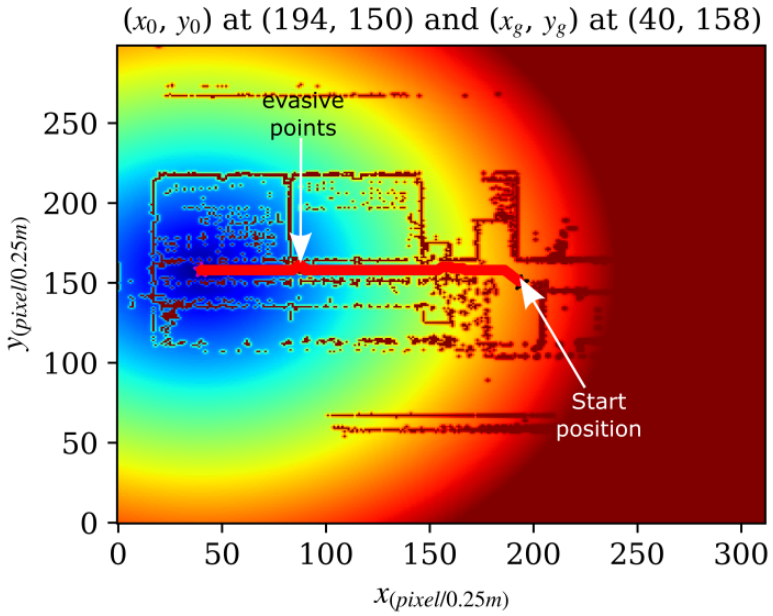
5.6.1 Experiment 1: Utilizing dynamic APF and VO is path planning problem

In the first scenario, the author would like to highlight the capability of the APF to plan a trajectory that avoids a dynamic obstacle while traveling from x_0, y_0 to x_g, y_g as shown in Fig. 5.3(a). The dark blue area shows the area with lowest potential that is the goal, while the red pixels indicates high-potential objects, such as walls and/or obstacles. As discussed in Section 5.4, the obstacle is placed in the dynamic map and the position of the nearby landmark from the static map, if any. For this map, consider three pixels in all directions surrounding the obstacle is considered. Once the position of the nearby obstacle is obtained, the potential of the dynamic field is then computed. The result is as depicted in Fig. 5.5(a). Following that, the static map is updated with the values that reflects the dynamic map's potential. Finally, the resultant trajectory needed to avoid the obstacle is calculated, as shown in Figure 5.3(b).

The second scenario also uses the same map, but presently the author attempts to search for a feasible path from a different x_0, y_0 to x_g, y_g. Accordingly, the blue area showing low potential indication is now shifted to $x_g, y_g = 182, 116$ as seen in Figure 5.4(a). This time, the gaps along the bottom side of the corridor's wall create some complications due to them, in reality, being glass walls. Therefore, the planner has to rely on the pillars' potential to avoid going through the wall. In addition, depending on the pixel size, a slightly lower potential gap in some areas may be observed, such as between a set of neighboring pillars. As a result, the



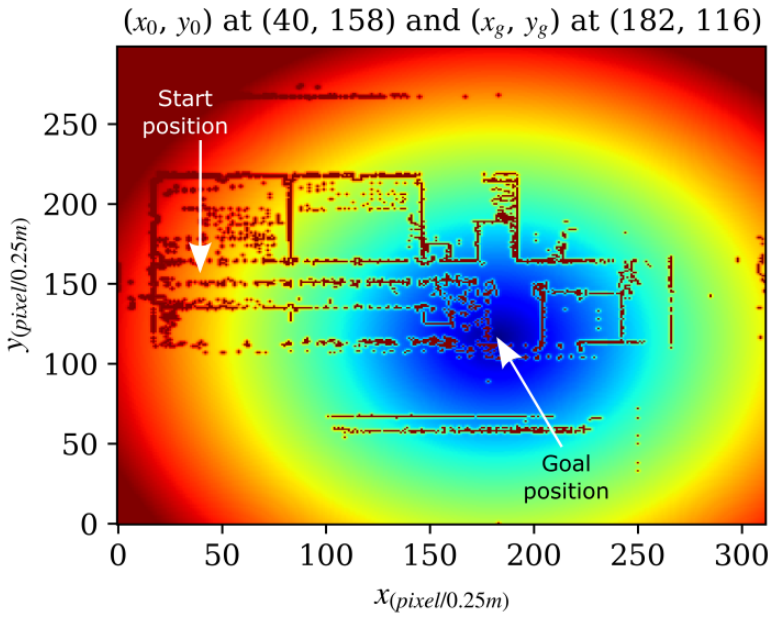
(a) Static map for first scenario. No dynamic obstacle here.



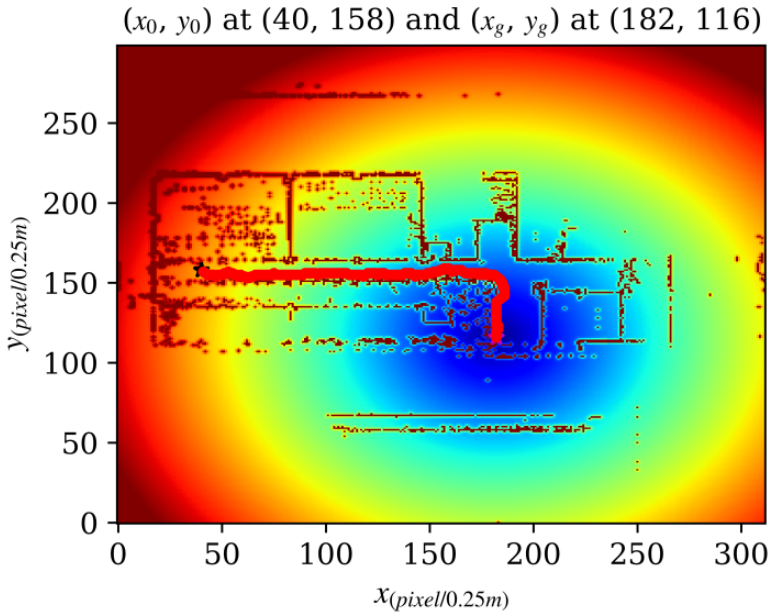
(b) Final map showing the resultant trajectory.

Figure 5.3. Result of APF for dynamic obstacle case.

planner will be prone to be trapped in these areas when searching for a feasible path. This phenomenon is also known as the local minima trap. To solve this

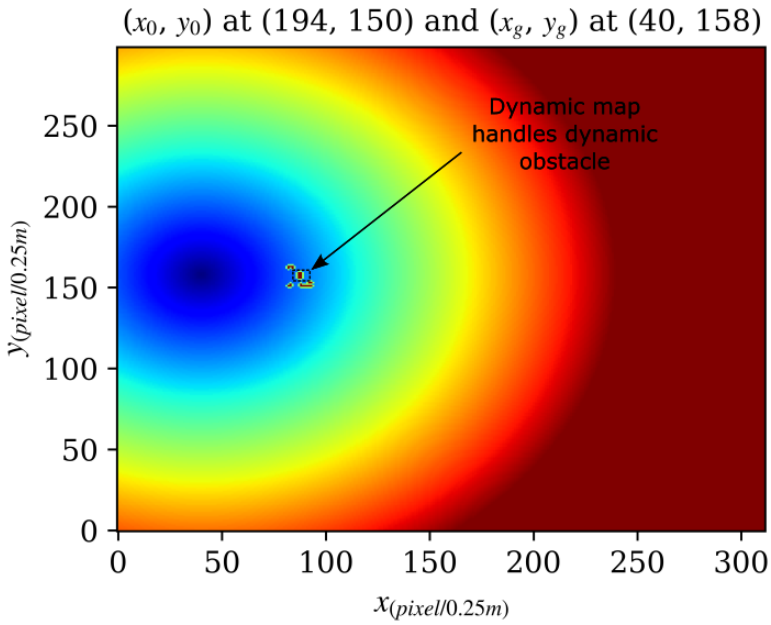


(a) Static map for second scenario.

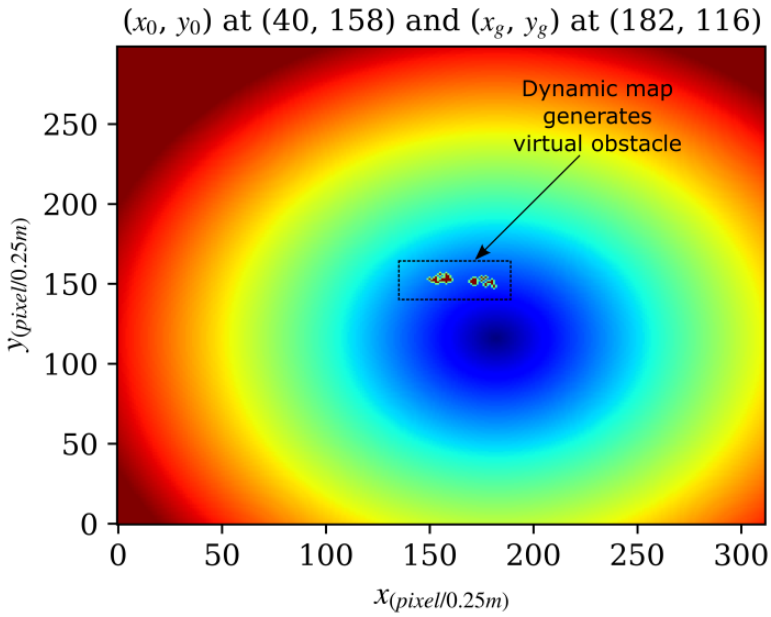


(b) The planner successfully arrived at goal with the aid of VO.

Figure 5.4. Result of APF with dynamic map for glass wall area case.



(a) Dynamic map with dynamic obstacle reflected in it.



(b) Dynamic map with VO.

Figure 5.5. Dynamic map visualized.

issue, the dynamic map is deployed to keep track of the local minima. Whenever the planner is stuck in the local minima, it will generate VO points at the involved

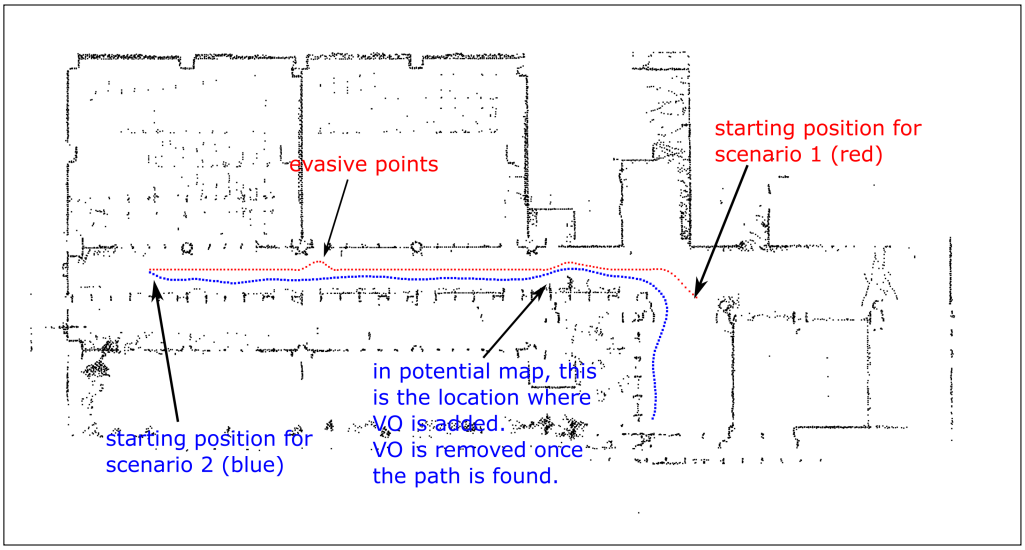


Figure 5.6. Result of Fig. 5.3(b) (red) and 5.4(b) (blue) in point cloud map

locations and reset the search as shown in Figure 5.5(b). As the search iteration repeats, the number of VO increases until the planner escapes the local minima. The final result of the search is shown in Fig. 5.4(b).

The results from both scenarios investigated are retranslated to the 3D point cloud format, as shown in Fig. 5.6. In actuality, transforming the map to 2D will inevitably reduce the resolution because the decimals are rounded to the closest whole number. However, the path calculated using the retranslated 3D map shows no sign of crashing and is within a safe range from the original landmarks, proving that the translation process does not affect the overall performance of the SMAD-PMV. Another point to note is that the second scenario's trajectory is slightly skewed at several points compared to the first scenario due to oscillation during the APF search. This can be solved by smoothing the path, which is out of the scope of this paper and hence will not be discussed.

5.6.2 Experiment 2: Utilizing RBPS in dynamic APF

The dynamic map eliminates the need to recompute the whole APF, and VO allows the planner to escape local minima. However, unlike graph-based planner that builds the knowledge of environment by connecting nodes and edges, APF relies on the values of the surrounding potentials and direct distance between the planner's current position and the goal position. Therefore, minimizing the distance difference causes the planner to miss some crucial marks, such as a door. Missing the door increases the possibility for the planner to be trapped in local minima. Consequently, more search iteration and VO is needed for the planner to escape them properly and finally enter the door. As shown in Figure 5.2, marking the door area as a region allows the local modification on the APF map. Figure 5.8 shows the results of the test dynamic APF with VO.

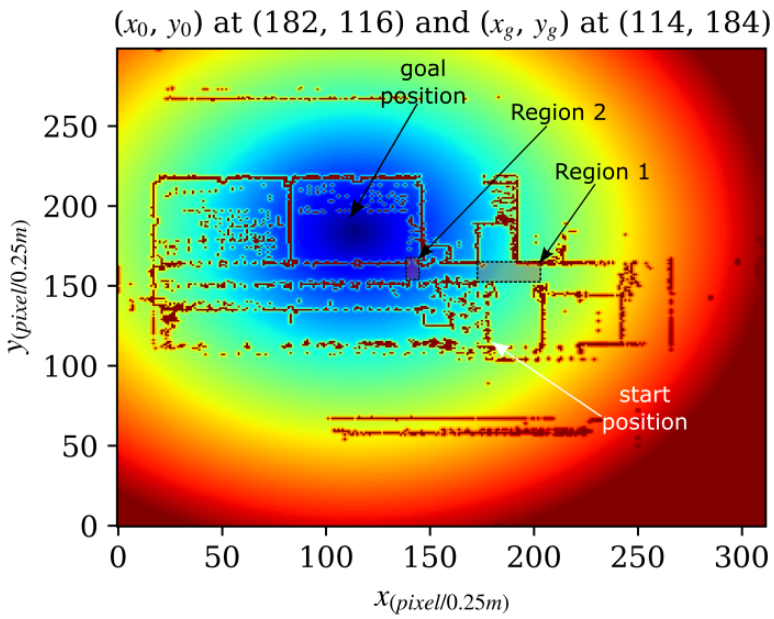
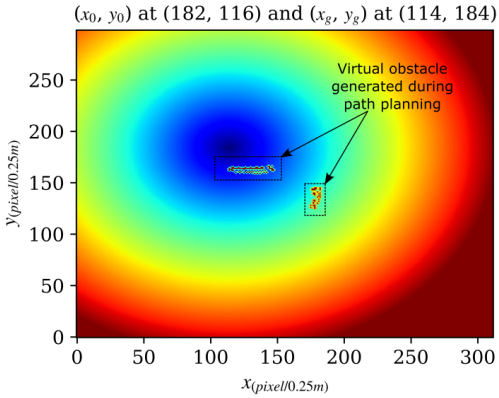
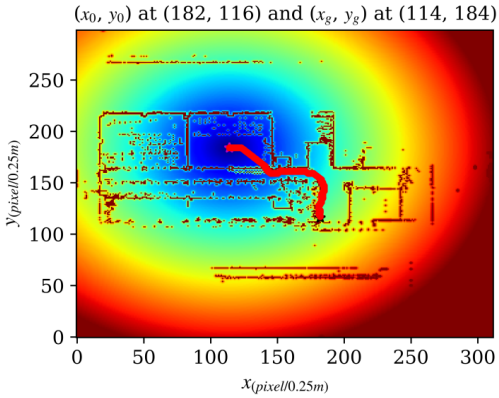


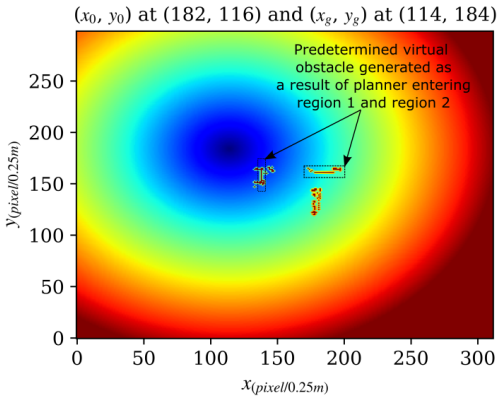
Figure 5.7. b2f2 static potentials and the region location



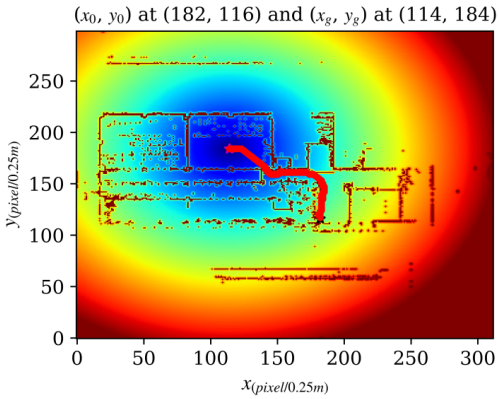
(a) Dynamic map without behavior modification.



(b) Final result without behaviour modification. Requires 30 iterations.



(c) Dynamic map with behavior modification when the planner searches within the region.



(d) Final result with behaviour modification. The generated trajectory is similar to Figure 5.8(b), however, only 14 iteration is needed to arrive to the goal.

Figure 5.8. APF searches for the goal, (x_g, y_g). The result of with and without region is compared.

To solve the described problem of SMAD-PMV disregarding the door, the RBPS is incorporated. For this purpose, the same map as that of Figure 5.7 of which was tested for static APF at b2f2, with the goal located in the mid-upper room (114, 184) is used. Region 1 and Region 2 mark the corridor and door's front (entrance side), respectively. Figure 5.8(a) shows the result of the planner without

RBPS. It can be observed that between $100 < x < 150$ and $150 < y < 170$, there are several VOs generated because the planner misses the door's entrance in Region 2 and becomes trapped in local minima at the nearby wall. Onwards, the planner takes 30 search iterations until it successfully arrives at the goal, as shown in Fig. 5.8(b). Next, the local behavior in the region is added, by defining a set of predefined VOs that block the vehicle from bypassing the door. As the planner enters the predefined region, it checks the position of the goal and creates a VO, if necessary. The outcome of adding a predetermined VO is shown in Fig. 5.8(c). The final result is shown in Fig. 5.8(d) where the planning now only takes 14 iterations to reach the goal. There is no difference between the characteristic of the path in Figures 5.8(b) and 5.8(d) since adding the blockage only reduces the needs to append VO at the local minima.

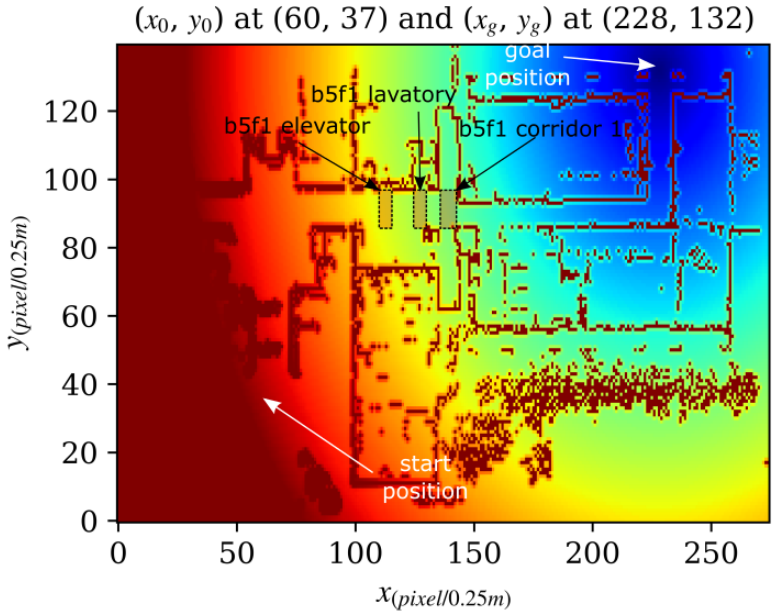
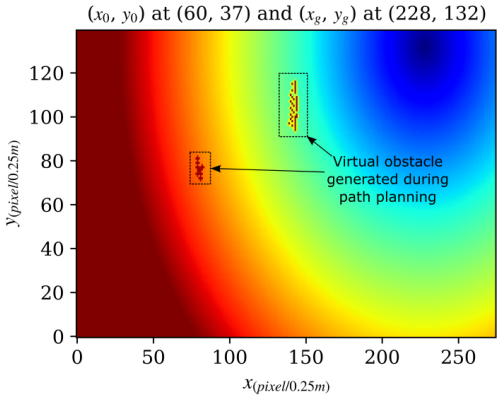
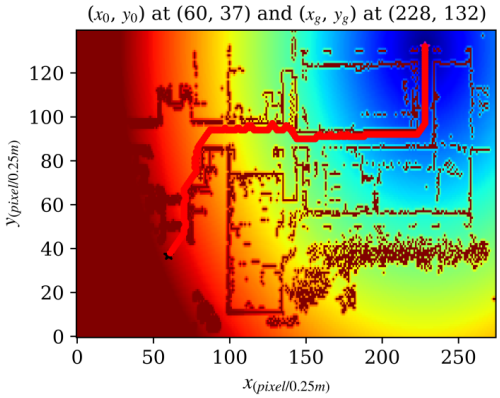


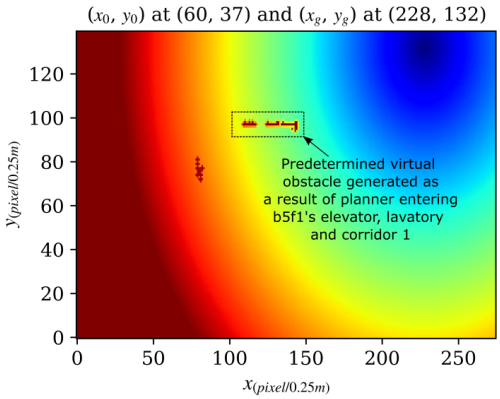
Figure 5.9. The b5f1 static potentials and the region location.



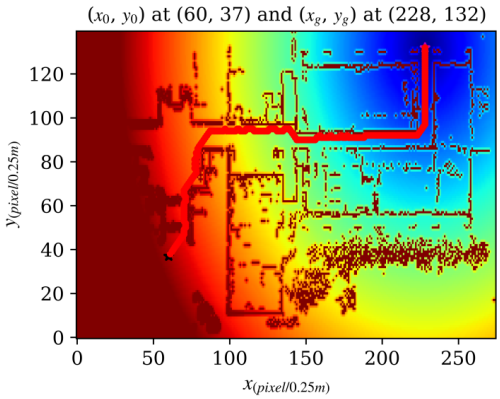
(a) Dynamic map without behavior modification.



(b) Final result without behaviour modification.It takes 24 iteration to search for the feasible path.



(c) Dynamic map with behavior modification when the planner searches within the region.



(d) Final result with behaviour modification. The generated trajectory is similar to Figure 5.10(b), however, only 8 iteration is needed to arrive to the goal.

Figure 5.10. APF searches for the goal, (x_g, y_g). The result of with and without region is compared.

The RBPS was tested in another map that is b5f1 of Fig. 4.5 and compared with non-RBPS. The results are shown in Fig. 5.9 and 5.10. The starting position is from the main entrance. Hence it needs to move towards the other end of the building. Here, the planners need to pass three areas; b5f1's elevator, lavatory, and corridor 1. The results in Fig. 5.10(a) show that the planner is stuck between

the wall area in the lobby ($70 < x < 80$, $70 < y < 85$) and the upper side of corridor 1 ($140 < x < 150$, $90 < y < 120$). Furthermore, the result in Fig. 5.10(b) shows that the planner was also slightly directed into the elevator and lavatory before continuing onwards. The planner finally found the feasible path after 24 iterations (0.65 seconds). Next, the region is defined in all three areas. The VOs are designated in all three areas, directly blocking the possibly misleading entrances. The inclusion of these VOs are shown in Fig. 5.10(c), whereby this configuration prevents the planner from overly building VOs at the top area of corridor 1. The planner successfully found the feasible trajectory after eight iterations (0.4 seconds), where the result is illustrated in Fig. 5.10(d).

Both results show the feasibility of the region-based strategy in relaxing the planner's pathfinding tasks.

5.6.3 Comparing improved APF and A*

In this subsection, the results pertaining APF's improvements is further discussed by comparing it to the A* algorithm. In this experiments, two obstacle penalty map for A* is precomputed, i.e: 1 pixels and 2 pixels penalty. These penalty is equivalent to 0.25 m and 0.5 meter respectively. For the APF, the map is used as it is, and the APF map is computed on-line. The dynamic APD amd Six different starting position is choosen in this experiment.

Fig. 5.11 shows the first three of the comparison between dynamic APF and A*. The results of Fig. 5.11(a)(a) and 5.11(b)(a) is similar to the previous Chapter

4's Fig. 4.7.

In Fig. 5.11(a)(b), 5.11(a)(c), 5.11(b)(b) and 5.11(b)(c) comes the interesting part. Previously the author mentioned about the augmented the map helps in finding a feasible path. In both 5.11(b)(b) and 5.11(b)(c), A* with 2 pixel penalty fails to find a feasible path. The reason is due to the pixels that is not evenly-placed in the door area in non augmented map, is now leveled. Therefore, during the computation of the obstacle map, the area is now not accessible as opposed to the 1 pixel penalty. However, when retransforming the points into point cloud map, under non augmented map and augmented map result's in Fig. 5.12(a) and 5.12(b), it is can be seen that Fig. 5.12(a)'s paths stays really near to the door despite the distance penalty. In actuality, transforming the map to 2D will inevitably reduce the resolution because the decimals are rounded to the closest whole number. Therefore in this case, the results suffers for the error. For the Fig. 5.11(a)(c) and 5.11(b)(c), the results shows some inconsistency in the non augmented map's case as opposed to the augmented map case. However, due to 1.0 pixel penalty is unable to close the transparent wall area properly, the A* planner fails in both the non-augmented and augmented scenario.

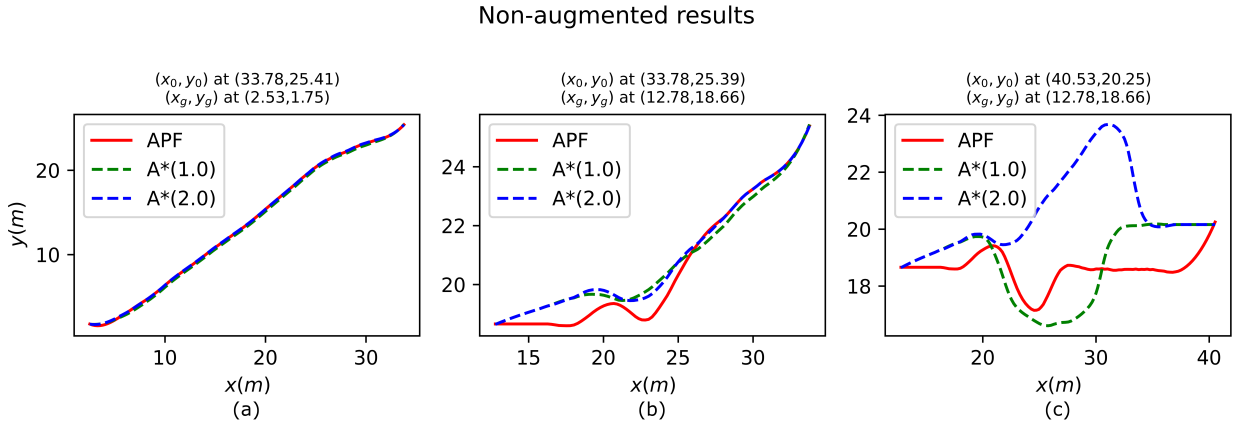
Fig. 5.13 shows the last three of the results. Directly in the Fig. 5.13(a))(a) and 5.13(b))(a), there are differences between the two algorithms. APF fails in the non augmented case despite using dynamic map and VO, but it successfully finds a path after the map orientation changes. A* fails in both cases. In many of the cases, A* with 1 pixel penalty goes through the glass area. The APF able to find the path in all augmented case. These results shows the effectiveness of

dynamic APF with VO in solving path planning problems.

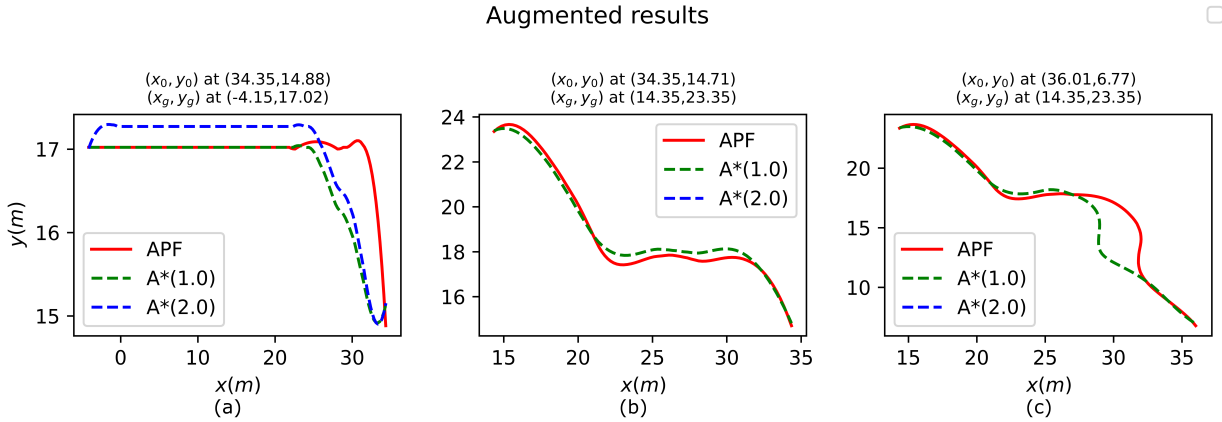
Fig. 5.15 shows the time taken to find the path for each of the scenarios. The computation of the path in APF takes a significantly higher time (with the highest time taken is 1.14 seconds) than the A*. However, the result of the paths needs to be taken into consideration as well, since, in all of the augmented results in Fig. 5.11 and Fig. 5.13, the APF is always able to find the suitable path. Furthermore, in Fig. 5.12 and Fig. 5.14, APF results, doesn't pass through any glass wall areas. In A* cases, the results are inconsistent despite finding the goal at a speedy computation time. Therefore, although improved APF is not fast compared to A*, considering the consistency of the result, for this experiment, APF is said to perform better than A*.

5.7 Conclusion

In this chapter, the dynamic APF with VO is proposed. The results show that by employing a dynamic map on top of the static map, it is possible to find a feasible path in less than 1 second. Furthermore, with the addition of VO, the APF is now able to escape the local minima problem. Furthermore, the inclusion of RBPS helps to reduce the time taken to find a path, proving its capability in solving behavior problems locally. The feasibility of the path is observed based on its characteristic after retransformed to the original orientation.

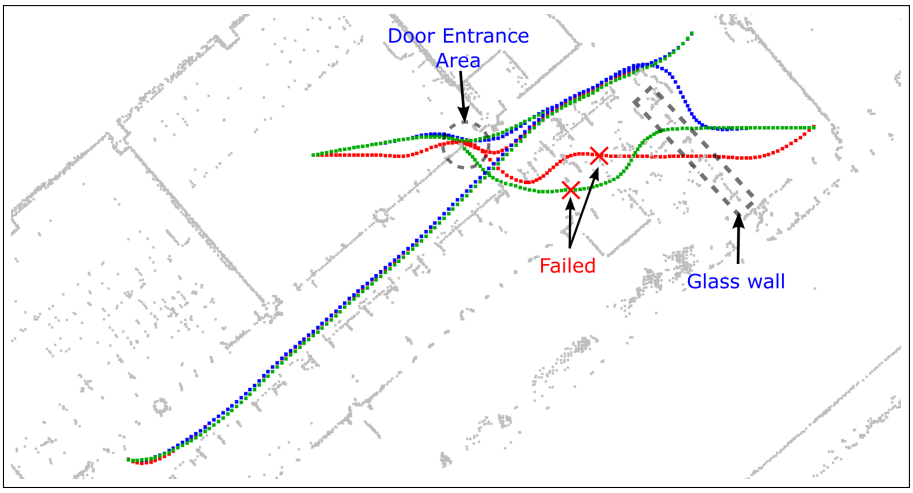


(a) The non augmented case are tested on APF and A* algorithm

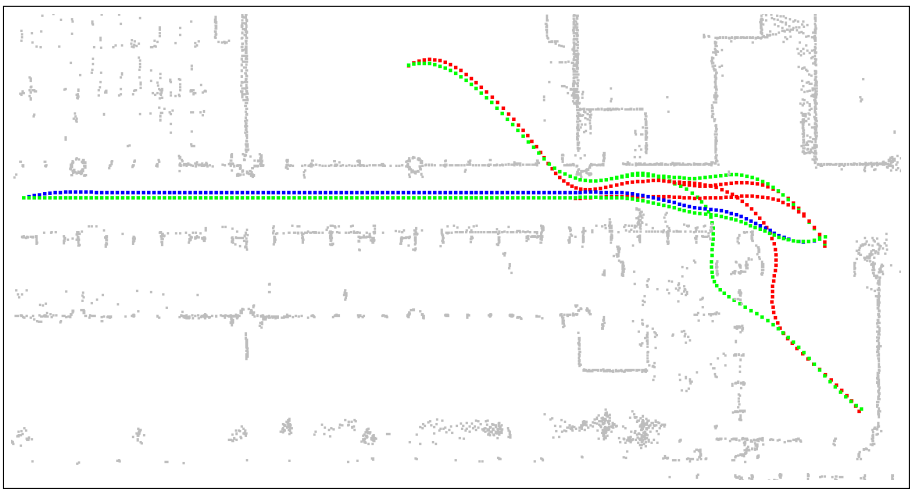


(b) The augmented case are tested on APF and A* algorithm

Figure 5.11. Comparing the characteristic of path in non-augmented and augmented map using A* and APF.

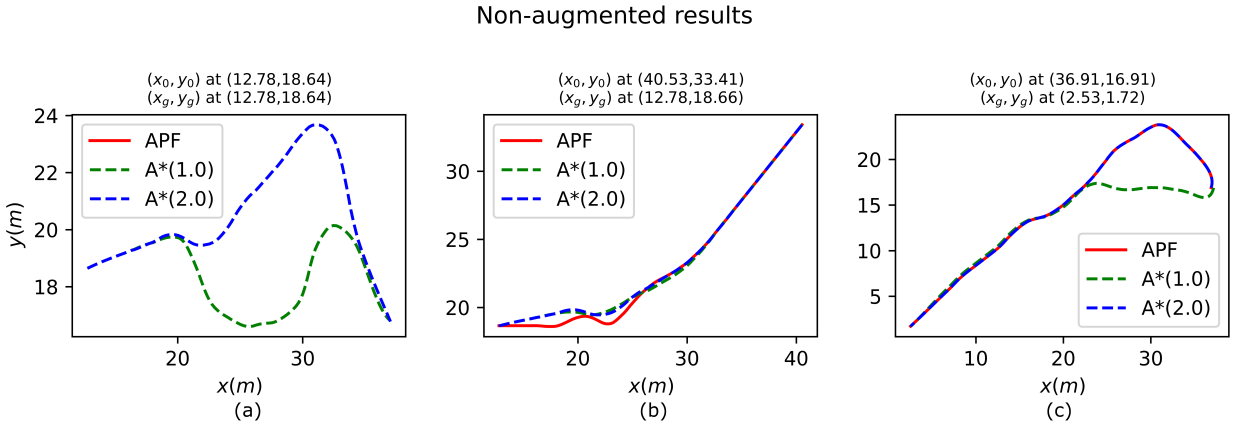


(a) Result in Fig. 5.11(a) reflected on point cloud map. The color of the paths correspond to the legend in Fig. 5.11(a)

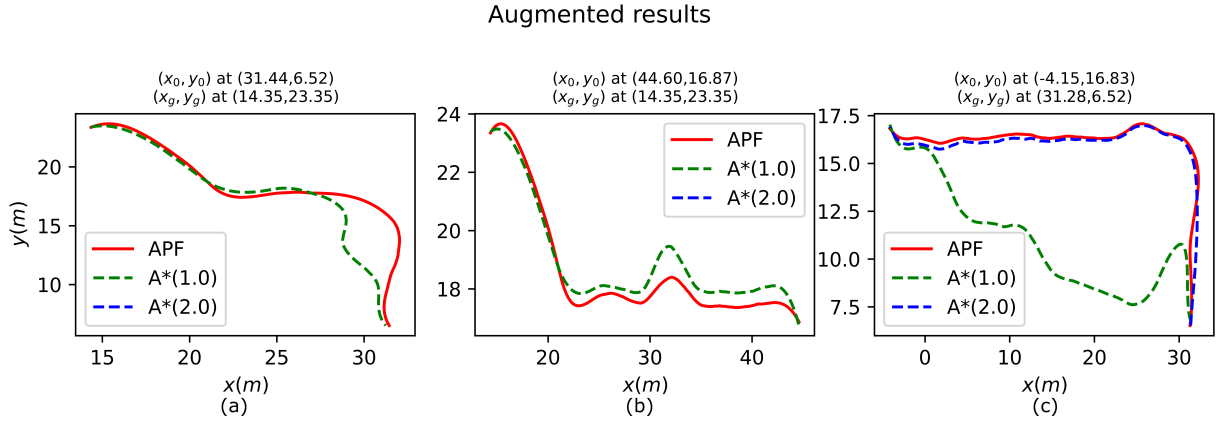


(b) Result in Fig. 5.11(b) reflected on point cloud map. The color of the paths correspond to the legend in Fig. 5.11(b)

Figure 5.12. The results of Fig. 5.11 shown in point cloud map.



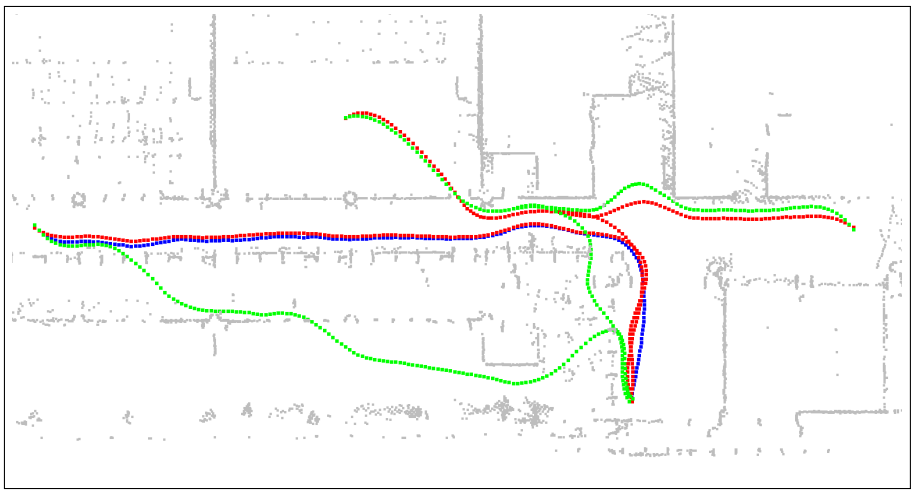
(a) The non augmented case are tested on APF and A* algorithm



(b) The augmented case are tested on APF and A* algorithm

Figure 5.13. Another Comparison of the characteristic of path in non-augmented and augmented map using A* and APF.

(a) Result in (a) reflected on point cloud map. The color of the paths correspond to the legend in (a)



(b) Result in Fig. 5.11(b) reflected on point cloud map. The color of the paths correspond to the legend in Fig. 5.13(b)

Figure 5.14. The results of Fig. 5.13 shown in point cloud map.

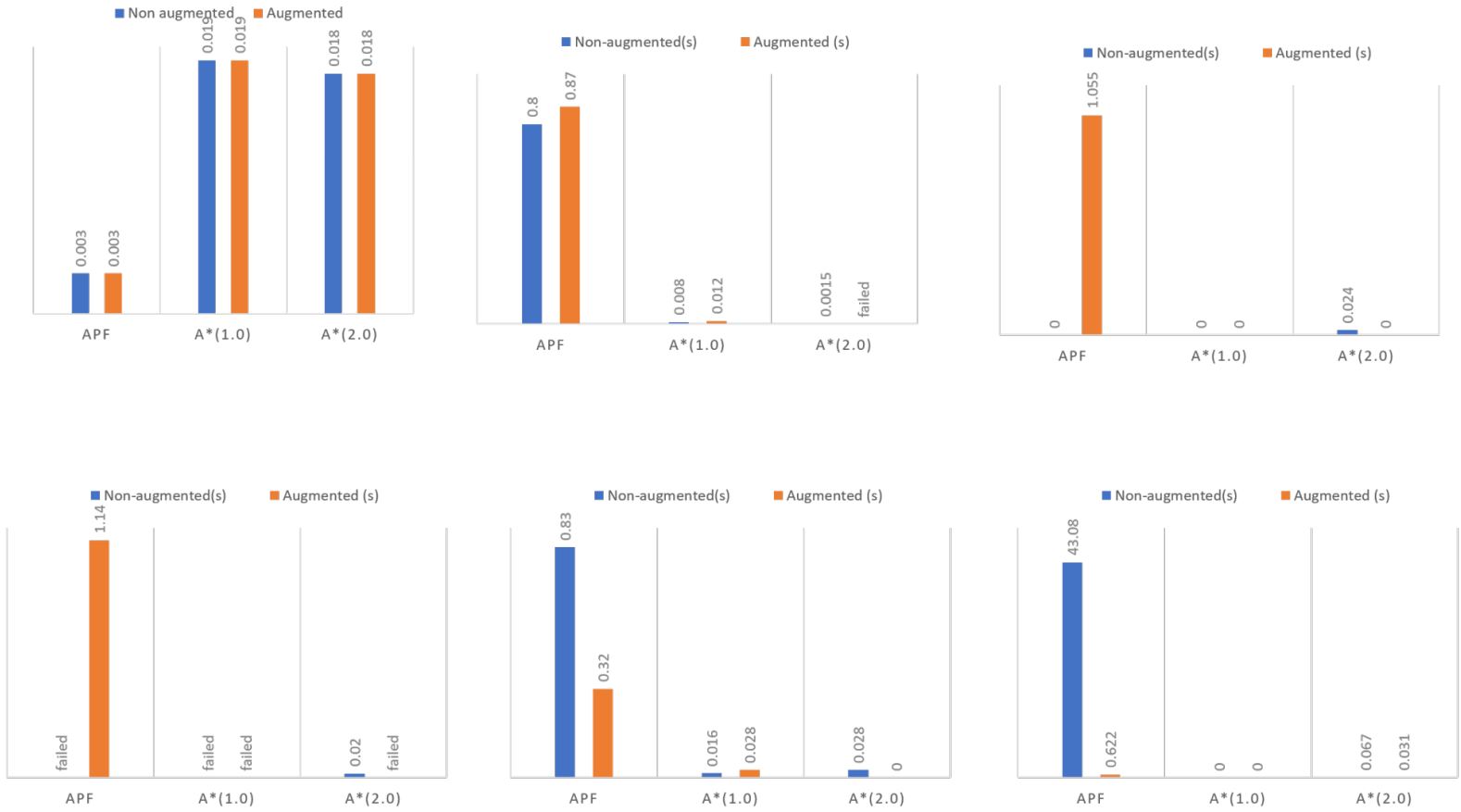


Figure 5.15. Time taken for all 6 scenarios

CHAPTER 6
CONCLUSION AND FUTURE WORKS

6.1 Summary of Research Contributions

The thesis presents the development of an autonomous, standalone, and modular-based personal mobility vehicle, the SMAD-PMV. Two varieties of path planning approach that revolves around the APF were also proposed for the SMAD-PMV prototype. In the following, the contributions from each chapters is summarized.

In Chapter 1: *"Introduction"*, the motivation behind the development of the SMAD-PMV is identified. The study aim to achieve system modularity both in terms of hardware and at least one part of software, and consequently develop the relevant path planning methods. To achieve hardware modularity, the installed components for ADS must be minimally invasive (or possibly, noninvasive). Meanwhile for software aspect, it is extremely arduous to design the whole pipleline modularly, hence at least one point in the pipeline should accomplish it.

In Chapter 2: *"Related Works"*, the summary of traditional and modern ADS and path planning algorithms are described. Their respective advantages and disadvantages are also discussed briefly. From the relevant comparisons and considering the practicality, standalone-modular approach is chosen. Particularly, the explanations of the two forms of ADS systems (*standalone* and *connected*) and architectures (*modular* and *end-to-end*), is the key to understand the reasoning for the selected approach.

In Chapter 3: *"Design and Deployment of Standalone-Modular Automated Driving Personal Mobility Vehicle"*, the SMAD-PMV prototype is described in detail. The ADS hardware includes an externally retrofitted, custom-made metal bracket with platforms, two sensors (LiDAR and omnidirectrional camera), a PC, and powered by an external battery. In addition to the capability of being attachable and removable easily, this original design is one of the novelty of this study. Following that, the mapping and localization along with path planning is explained. The author emphasize on RBPS as the modular aspect in achieving MDA for the software aspect. The SMAD-PMV was deployed as it is in real-world environment, and travels safely without crashing onto any obstacle or landmarks. Overall, objective (1) was achieved in this chapter.

In Chapter 4: *"Proposed Path Planning Method 1: Augmented by Rotation Map"*, the proposed concept and the experimentation of the APF with map augmentation is presented. The author found that augmenting the map and standardizing its orientation helps in getting a consistent result for path computation. Besides that, based on a preliminary result of comparison between A* and APF algorithm, the author select the APF due to its practicality in the selected case study. This chapter fulfills Objective (2) and also partially Objective (3).

In Chapter 5: *"Proposed Path Planning Method 2: Dynamic Artificial Potential Field with Virtual Obstacle"*, proposes a method to further improve the APF with augmented map, specifically to solve the local minima trap problem. This works by adding a predetermined behaviour, the VO that blocks the vehicle

from bypassing a defined object and/or landmark. Moreover, this computation is basically accelerated by taking advantage of the previously calculated static APF map to generate the dynamic APF map. This chapter signals the completion of achieving Objective (3).

As a conclusion, the a self-driving PMV with MDA was successfully developed and deployed in real-world environment. Then, the author also developed two path planning methods based on modified APF algorithm. Following that, the limitations of the modified APF method were identified, and solved using a dynamic APF with VOs. According to the above summarized discussions, all of the objectives stated in Chapter 1 were achieved accordingly.

6.2 Future Works

In this section, the author indicate the future direction of this study. First and foremost, further improvement of the SMAD-PMV is needed. In the present system, it is possible to perform the reverse operations by computing the distance between the LiDAR object detection and the obstacle, however, the system doesn't have a rear sensor installed, therefore currently it relies on the passenger's observation. The reverse motion is important to enter critical areas like an elevator (so that in the event of fatal emergencies, the passenger can just take over the control and leave the area).

The servo motors are recommended to be replaced with Brushless DC motors. Since the resolution of a servo is low, the vehicle is unable to perform a very smooth operation. Despite the heading is zero according to the calculated angles, there

might be a slight error in the connection of the mechanism and the angle of the servo. Thus, the vehicle will slightly divert from the original path whenever in a straight motion.

Presently, the LiDAR has two limitations. First, the data is not dense, therefore it is difficult to perform obstacle detection when the size of the obstacle or landmarks is small. This can be improved by fusing it with the available 360°. Furthermore, the cost of LiDAR is expensive, and waiting for the cost to lower when the ADS is mass-produced might take a long time, so the option is to replace it with a Camera. Both of the proposed future works are ongoing research by the students in Advanced Driving Systems Laboratory.

Regarding the RBPS, more validation especially in the urban area should be conducted. Furthermore, the strategy is expected to work well when combined with the state-transition model approach for more flexibility in deciding the proper behavior in the region.

BIBLIOGRAPHY

[1] Corporation, S. M. ET4D. https://www.suzuki.co.jp/welfare/et4d/, accessed on 2021 March 19.

[2] World Health Organization. Summary : World report on disability 2011 (WHO/NMH/VIP/11.01). https://apps.who.int/iris/handle/10665/70670, accessed on 2021 March 11.

[3] United Nations Department of Economic and Social Affairs Population Division. World Population Ageing 2017 - Highlights. https://www.un.org/en/development/desa/population/publications/pdf/ageing/WPA2017_Highlights.pdf, accessed on 2021 March 11.

[4] Shimchik, I., Sagitov, A., Afanasyev, I., Matsuno, F. and Magid, E. Developing an autonomous service car: golf-cart prototype modelling. *The Proceedings of JSME annual Conference on Robotics and Mechatronics (Robomec)*, 2016. 2016: 1–7. ISSN 2424-3124.

[5] Pendleton, S., Uthaicharoenpong, T., Chong, Z. J., Guo Ming James Fu, Qin, B., Wei Liu, Xiaotong Shen, Zhiyong Weng, Kamin, C., Ang, M. A., Kuwae, L. T., Marczuk, K. A., Andersen, H., Mengdan Feng, Butron, G., Chong, Z. Z., Ang, M. H., Frazzoli, E. and Rus, D. Autonomous golf cars for public trial of mobility-on-demand service. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015. ISBN 978-1-4799-9994-1. 1164–1171.

[6] Liu, Z., Jia, X. and Cheng, W. Solving the Last Mile Problem: Ensure the Success of Public Bicycle System in Beijing. *Procedia - Social and Behavioral Sciences*, 2012. 43: 73–78. ISSN 1877-0428.

[7] Somenahalli, S., Hayashi, Y., Taylor, M., Akiyama, T., Adair, T. and Sawada, D. Accessible transportation and mobility issues of elderly — how does Australia compare with Japan? *Journal of Sustainable Urbanization, Planning and Progress*, 2016. 1(1): 31–43. ISSN 2424-8053.

[8] Ulrich, K. T. Estimating the technology frontier for personal electric vehicles. *Transportation Research Part C: Emerging Technologies*, 2005. 13(5-6): 448–462. ISSN 0968090X.

[9] Simpson, R. C. Smart wheelchairs: A literature review. *The Journal of Rehabilitation Research and Development*, 2005. 42(4): 423.

[10] Andersen, H., You Hong Eng, Wei Kang Leong, Chen Zhang, Hai Xun Kong, Pendleton, S., Ang, M. H. and Rus, D. Autonomous personal mobility scooter for multi-class mobility-on-demand service. *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE. 2016. ISBN 978-1-5090-1889-5. 1753–1760.

[11] Hirai, M., Tomizawa, T., Muramatsu, S., Sato, M., Kudoh, S. and Suehiro, T. Development of an intelligent mobility scooter. *2012 IEEE Int. Conf. Mechatronics Autom.* Chengdu, China: IEEE. 2012. ISBN 978-1-4673-1278-3. ISSN 2152-7431. 46–52.

[12] Duchoň, F., Babinec, A., Kajan, M., Beňo, P., Florek, M., Fico, T. and Jurišica, L. Path Planning with Modified a Star Algorithm for a Mobile Robot. *Procedia Engineering*, 2014. 96: 59–69. ISSN 1877-7058.

[13] Dolgov, D., Thrun, S., Montemerlo, M. and Diebel, J. Path Planning for Autonomous Vehicles in Unknown Semi-structured Environments. *The International Journal of Robotics Research*, 2010. 29(5): 485–501. ISSN 0278-3649.

[14] Raksincharoensak, P., Hasegawa, T. and Nagai, M. Motion Planning and Control of Autonomous Driving Intelligence System Based on Risk Potential Optimization Framework. *International Journal of Automotive Engineering*, 2016. 7(AVEC14): 53–60.

[15] Hamid, U. Z. A., Zakuan, F. R. A., Zulkepli, K. A., Azmi, M. Z., Zamzuri, H., Rahman, M. A. A. and Zakaria, M. A. Autonomous emergency braking system with potential field risk assessment for frontal collision mitigation. *2017 IEEE Conference on Systems, Process and Control (ICSPC)*. 2017. 71–76.

[16] Toyoshima, A., Nishino, N., Chugo, D., Muramatsu, S., Yokota, S. and Hashimoto, H. Autonomous Mobile Robot Navigation: Consideration of the Pedestrian's Dynamic Personal Space. *2018 IEEE 27th International Symposium on Industrial Electronics (ISIE)*. IEEE. 2018. 1094–1099.

[17] Lazarowska, A. Discrete Artificial Potential Field Approach to Mobile Robot Path Planning. *IFAC-PapersOnLine*, 2019. 52(8): 277–282. ISSN 2405-8963.

[18] Amari, S. and Wu, S. Improving support vector machine classifiers by modifying kernel functions. *Neural Networks*, 1999. 12(6): 783–789. ISSN 0893-6080.

[19] Perez, L. and Wang, J. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *arXiv*, 2017.

[20] The British Standards Institution. *PAS 1880:2020 - Guidelines for developing and assessing control systems for automated vehicles*. Technical report. The British Standards Institution. 2020.

[21] Yurtsever, E., Lambert, J., Carballo, A. and Takeda, K. A Survey of Autonomous Driving: Common Practices and Emerging Technologies. *CoRR*, 2019. abs/1906.0.

[22] Tian, Y., Pei, K., Jana, S. and Ray, B. DeepTest: Automated Testing of Deep-Neural-Network-Driven Autonomous Cars. *Proc. 40th Int. Conf. Softw. Eng.* New York, NY, USA: Association for Computing Machinery. 2018, ICSE '18. ISBN 9781450356381. 303–314.

[23] Bojarski, M., Testa, D. D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J. and Zieba, K. End to End Learning for Self-Driving Cars, 2016.

[24] Russell, S. and Norvig, P. *Artificial Intelligence: A Modern Approach*. 3rd ed. USA: Prentice Hall Press. USA, 2009. ISBN 0136042597.

[25] Sallab, A., Abdou, M., Perot, E. and Yogamani, S. Deep Reinforcement Learning framework for Autonomous Driving. *Electron. Imaging*, 2017. 2017(19): 70–76. ISSN 2470-1173.

[26] Kendall, A., Hawke, J., Janz, D., Mazur, P., Reda, D., Allen, J.-M., Lam, V.-D., Bewley, A. and Shah, A. Learning to Drive in a Day, 2018.

[27] Katrakazas, C., Quddus, M., Chen, W.-H. and Deka, L. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transp. Res. Part C Emerg. Technol.*, 2015. 60: 416–442.

[28] Fernandes, L. C., Souza, J. R., Pessin, G., Shinzato, P. Y., Sales, D., Mendes, C., Prado, M., Klaser, R., Magalhães, A. C., Hata, A., Pigatto, D., Castelo Branco, K., Grassi, V., Osorio, F. S. and Wolf, D. F. CaRINA Intelligent Robotic Car: Architectural design and applications. *J. Syst. Archit.*, 2014. 60(4): 372–392.

[29] Sh. Levin, M. *Modular Systems, Combinatorial Engineering Frameworks*. Springer, Cham. 2015.

[30] Smith, S. M., An, P. E., Holappa, K., Whitney, J., Burns, A., Nelson, K., Heatzig, E., Kempfe, O., Kronen, D., Pantelakis, T., Henderson, E., Font, G., Dunn, R. and Dunn, S. E. The Morpheus ultramodular autonomous underwater vehicle. *IEEE J. Ocean. Eng.*, 2001. 26(4): 453–465.

[31] Liu, S. Affordable and Reliable Autonomous Driving Through Modular Design. In: *Eng. Auton. Veh. Robot. DragonFly Modul. Approach*. IEEE. 1–12. 2019.

[32] Wang, I., Factor, B., Fladung, S. and Stenson, R. Modular hardware infrastructure for autonomous underwater vehicles. *Proc. Ocean. 2005 MTS/IEEE*. Washington, DC, USA. 2005. 2652–2655.

[33] PerceptIn Incorporated. PerceptIn. `https://www.perceptin.io/`, accessed on 2021 June 20.

[34] Berger, C. and Dukaczewski, M. Comparison of architectural design decisions for resource-constrained self-driving cars - A multiple case-study. Plödereder, E., Grunske, L., Schneider, E. and Ull, D., eds. *Informatik 2014*. Bonn: Gesellschaft für Informatik e.V. 2014. 2157–2168.

[35] Sugahara, Y., Akiyama, H., Jong, J., Endo, M. and Okamoto, J. *Design and Control of a Human-Powered Robotic Personal Mobility Vehicle Prototype*. Springer, Cham. 2019.

[36] Claussmann, L., Revilloud, M., Gruyer, D. and Glaser, S. A Review of Motion Planning for Highway Autonomous Driving. *IEEE Trans. Intell. Transp. Syst.*, 2020. 21(5): 1826–1848.

[37] Latombe, J.-C. *Introduction and Overview*, Boston, MA: Springer, Boston, MA. 1st ed., 1991. ISBN 978-0-7923-9206-4, 1–57.

[38] Amato, N. M., Bayazit, O. B., Dale, L. K., Jones, C. and Vallejo, D. OBPRM: An obstacle-based PRM for 3D workspaces. *Proc. Int. Work. Algorithmic Found. Robot.* 1998. 155–168.

[39] Harabor, D. and Grastien, A. Online Graph Pruning for Pathfinding on Grid Maps. *Proc. Twenty-Fifth AAAI Conf. Artif. Intell.* AAAI Press. 2011, AAAI'11. 1114–1119.

[40] Yang, L., Qi, J., Song, D., Xiao, J., Han, J. and Xia, Y. Survey of Robot 3D Path Planning Algorithms. *J. Control Sci. Eng.*, 2016. 2016: 1–22. ISSN 1687-5249.

[41] Tibebu, H., Roche, J., De Silva, V. and Kondoz, A. LiDAR-Based Glass Detection for Improved Occupancy Grid Mapping. *Sensors*, 2021. 21(7): 2263.

[42] Wang, R., Bach, J. and Ferrie, F. P. Window detection from mobile LiDAR data. *2011 IEEE Workshop on Applications of Computer Vision (WACV)*. 2011. 58–65.

[43] Shi, W., Alawieh, M. B., Li, X. and Yu, H. Algorithm and hardware implementation for visual perception system in autonomous vehicle: A survey. *Integration*, 2017. 59: 148–156.

[44] ASUSTek Computer Inc. Asus ZenBook Flip UX461. `https://asus.com/us/Laptops/For-Home/ZenBook/ASUS-ZenBook-Flip-14-UX461/`, accessed on 2021 June 21.

[45] Yang, B., Cao, X., Li, X., Yuen, C. and Qian, L. Lessons Learned From Accident of Autonomous Vehicle Testing: An Edge Learning-Aided Offloading Framework. *IEEE Wirel. Commun. Lett.*, 2020. 9(8): 1182–1186.

[46] Zong, W., Zhang, C., Wang, Z., Zhu, J. and Chen, Q. Architecture Design and Implementation of an Autonomous Vehicle. *IEEE Access*, 2018. 6: 21956–21970.

[47] Velodyne Lidar. Velodyne Puck. `https://velodynelidar.com/products/puck/`, accessed on 2021 June 21.

[48] Company, E. K. Kodak PIXPRO SP360. `https://kodakpixpro.com/cameras/360-vr/sp360`, accessed on 2021 June 21.

[49] Wang, X., Mizukami, Y., Tada, M. and Matsuno, F. Navigation of a mobile robot in a dynamic environment using a point cloud map. *Artif. Life Robot.*, 2021. 26(1): 10–20.

[50] Magnusson, M., Lilienthal, A. and Duckett, T. Scan registration for autonomous mining vehicles using 3D-NDT. *J. F. Robot.*, 2007. 24(10): 803–827. ISSN 15564959.

[51] Kato, S., Takeuchi, E., Ishiguro, Y., Ninomiya, Y., Takeda, K. and Hamada, T. An Open Approach to Autonomous Vehicles. *IEEE Micro*, 2015. 35(6): 60–68.

[52] Yu, L., Kong, D., Shao, X. and Yan, X. A Path Planning and Navigation Control System Design for Driverless Electric Bus. *IEEE Access*, 2018. 6: 53960–53975.

[53] Zhu, Q. Hidden Markov model for dynamic obstacle avoidance of mobile robot navigation. *IEEE Trans. Robot. Autom.*, 1991. 7(3): 390–397.

[54] Zhang, J., Chen, H., Song, S. and Hu, F. Reinforcement Learning-Based Motion Planning for Automatic Parking System. *IEEE Access*, 2020. 8: 154485–154501.

[55] Lefèvre, S., Vasquez, D. and Laugier, C. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH J.*, 2014. 1(1): 1.

[56] Hornyak, V., VanSwearingen, J. M. and Brach, J. S. Measurement of Gait Speed. *Top. Geriatr. Rehabil.*, 2012. 28(1): 27–32.

[57] Kumar, P. B., Rawat, H. and Parhi, D. R. Path planning of humanoids based on artificial potential field method in unknown environments. *Expert Systems*, 2019. 36(2): e12360. ISSN 0266-4720.

[58] Klančar, G., Seder, M., Blažič, S., Škrjanc, I. and Petrović, I. Drivable Path Planning Using Hybrid Search Algorithm Based on E* and Bernstein-Bézier Motion Primitives. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019: 1–15. ISSN 2168-2232 VO -.

[59] Thrun, S. Learning Occupancy Grid Maps with Forward Sensor Models. *Auton. Robots*, 2003. 15(2): 111–127.

[60] Azmi, M. Z. and Ito, T. Artificial Potential Field with Discrete Map Transformation for Feasible Indoor Path Planning. *Appl. Sci.*, 2020. 10(24).

[61] Li, X., Sun, Z., Cao, D., He, Z. and Zhu, Q. Real-Time Trajectory Planning for Autonomous Urban Driving: Framework, Algorithms, and Verifications. *IEEE/ASME Transactions on Mechatronics*, 2016. 21(2): 740–753.

[62] Lee, M. C. and Park, M. G. Artificial potential field based path planning for mobile robots using a virtual obstacle concept. *Proc. 2003 IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM 2003).* 2003, vol. 2. 735–740.

[63] Zheng, Y., Shao, X., Chen, Z. and Zhang, J. Improvements on the virtual obstacle method. *Int. J. Adv. Robot. Syst.*, 2020. 17(2): 172988142091176.

APPENDIX A

Hardware Specifications

A.1 Mobile Battery

Figure A.1. Velodyne VLP-16.

Table A.1: Suaoki S270 150Wh Portable Mobile Battery

Specification	Details
Dimension (mm)	184.5 × 109.5 × 118.5
Weight (kg)	1.3
Power	150 watt-hours (11.1V 13,500 mAh)
Outputs	2 × AC outlet (max 100 W, peak power 150 W) 4 × DC port (15 A/ 180 W max) 3 × USB ports (10.5 W max) 1 × QC3.0 USB port (18 W max)

A.2 Velodyne VLP-16 LiDAR

Figure A.2. Velodyne VLP-16.

Table A.2: Velodyne VLP-16 specifications

Specification	Details
Dimension (mm)	103 × 72
Weight (kg)	0.83
Power consumption	8 W
Environment Protection	IP67
Rotation rate (Hz)	5 - 20
Accuracy	±3 cm
Measurement range	Up to 100 meters
Scan layer	16 layers
Outputs	UDP packets containing: – Distance – Calibrated reflectivities – Rotation angles – Synchronized time stamps in μs

A.3 Kodak Pixpro specification

Figure A.3. Kodak PixPro360.

Table A.3: Kodak Pixpro 360 Specification

Specification	Details
Dimension (mm)	41.1 × 52.1 × 50.0
Weight (kg)	0.103
Image Sensor Pixels	16.36 Megapixels
Field of View	Max. 214 Degree
Anti hanshake	Electronic Image Stabilization (EIS)
Power	Rechargeable Li-ion Battery LB-080 3.6 V 1,250 mAh

A.4 Laptop specification

Figure A.4. Asus UX461.

Table A.4: Asus UX461 Specification

Specification	Details
Dimension (mm)	139 × 327 × 226
Weight (kg)	1.5
Power	57 Wh lithium-polymer battery 65W power adapter
Operating System	Ubuntu 18.04
Robot Operating System Version	Melodic
Processor	Intel Core i7-8565U 1.8GHz Quad-core, with Turbo boost (up to 4.6GHz) and 8MB cache
Graphics	NVIDIA GeForce MX150
Memory	8GB 2133MHz LPDDR3

A.5 Voltage regulator specification

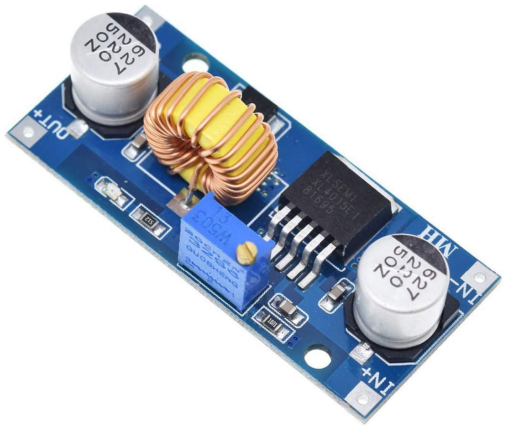


Figure A.5. HiLetgo DC-DC regulator.

Table A.5: Voltage regulator Specification

Specification	Details
Dimension (mm)	54 × 23 × 18
Input Voltage (V)	4 – 38
Output Voltage (V)	1.25 – 36
Output Current (A)	0 – 5
Output power (W)	75 W

A.6 Arduino Mega specification

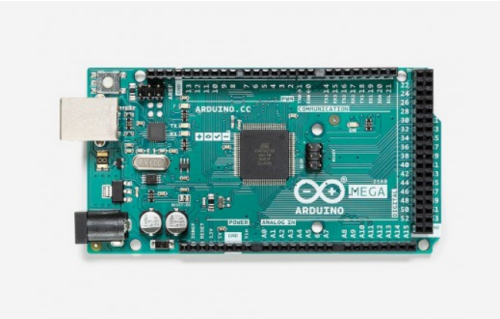


Figure A.6. Arduino Mega.

Table A.6: Arduino Mega (ATmega2560) specifications

Specification	Details
Dimension (mm)	101.52 × 53.5
Weight (grams)	37
Operating voltage	5V
Analog Input pins	16

A.7 Steering Servo specification

Figure A.7. Gearwurx Torxis Servo Motor.

Table A.7: Steering servo (Gearwurx Torxis)

Specification	Details
Dimension (inch)	6.5 × 5.5 × 4.5
Weight (lbs)	2.4
Speed	60 degrees in 500 ms
Voltage	12 VDC
Max angle	270 degrees

LIST OF PUBLICATIONS

The work presented in this thesis was obtained in close collaboration with Advanced Driver Assistant Systems Laboratory Senior Car team member and is documented in published articles, proceedings and presentation as listed below:

Journal Papers

1. **M.Z. Azmi**; T. Ito, "Development of Autonomous Four-wheel Personal Mobility Vehicle with Modular Design Architecture", *IEEE Access*; Submitted for review. [Reference Chapter: 3, 4, 5]

2. **M.Z. Azmi**; T. Ito, "Artificial Potential Field with Discrete Map Transformation for Feasible Indoor Path Planning", *MDPI Applied Sciences*; 2020; 10(24); pp 8987 [Reference Chapter: 3, 4]

Conference and Proceedings

1. **M.Z. Azmi**, R. Toya, M. Shikahama, Y. Nakayama, T. Ito, "Space Constraints Autonomous Navigation System using Artificial Potential Field and Model Predictive Control", In *Proceedings of the 5th International Symposium on Future Active Safety Technology towards Zero Accidents(FAST–ZERO 19)*, Blacksburg, USA, 9–11 September 2019 [Reference Chapter: 4, 5]

2. **M. Z. Azmi**, Y. Nakayama, R. Toya, M. Shikahama, T. Ito, "Modular Controller Box for Autonomous Personal Mobility", In *Proceeding of 26th*

World Congress on Intelligent Transport Systems (ITSWC19), Singapore, 21–25 October 2019 [Reference Chapter: 3]

3. **M.Z. Azmi**, Y. Nakayama, R. Toya, M. Shikahama, K. Kogane, T. Ito, "4–Wheel Senior Car Simple Vehicle Model", *JSAE Annual Congress Spring*, Yokohama, Japan, 22–24 May 2019

4. **M.Z. Azmi**, T. Ito et al, "Vehicle Model for 4-Wheeled Personal Mobility Scooter – an Experimental Study" Poster Presentation in *South East Asian Technical University Consortium (SEATUC)*, Hanoi, Vietnam, 14–15 March 2019