Design and Development of an Autonomous Scaled Electric Combat Vehicle

by

Aaron Hao Tan

A thesis submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the requirements for the degree of

Master of Applied Science in Mechanical Engineering

Faculty of Engineering and Applied Science

University of Ontario Institute of Technology

Oshawa, Ontario, Canada

April 2019

THESIS EXAMINATION INFORMATION

Submitted by: **Aaron Hao Tan**

Master of Applied Science in **Mechanical Engineering**

Thesis title: Design and Development of an Autonomous Scaled Electric Combat Vehicle

An oral defense of this thesis took place on April 18, 2019 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Martin Agelin-Chaab
Research Supervisor	Moustafa El-Gindy
Research Co-supervisor	Haoxiang Lang
Examining Committee Member	Xianke Lin
External Examiner	Jaho Seo, FEAS

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

ABSTRACT

Current literature pertaining to multi-steerable mobile robots and the progression of military vehicles in the past few decades suggest a lack of effort in pursuing advanced technologies in this joint area. As a result, a novel scaled robotic platform that features independent wheel actuation and autonomous navigation capabilities is developed in this work to represent a potential future design of combat vehicles. The following thesis discusses the details of the mechanical systems in addition to the embedded electronics and software architecture. From there, previously developed mapping and path planning algorithms in addition to a developed localization algorithm are implemented to achieve autonomous navigation. Furthermore, a vision-based close quarter pose correction algorithm is designed and developed to improve upon the limitations imposed by current navigation methodologies. The result of this work is a proposed prototype capable of navigation and precise positioning.

Keywords: autonomous; combat vehicle; obstacle avoidance; path planning; ROS

AUTHOR'S DECLARATION

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Aaron Hao Tan

STATEMENT OF CONTRIBUTIONS

Parts of this thesis has been published or submitted for publication in the following:

A. H. Tan, H. Lang and M. El-Gindy, "The Design and Development of a Novel Autonomous Multi-Wheeled Vehicle," *International Journal of Robotics and Automation*, 2019. (Submitted)

A. H. Tan, H. Lang and M. El-Gindy, "A Novel Autonomous Scaled Electric Combat Vehicle," in ASME IDETC/CIE, Anaheim, 2019. (Submitted)

A. H. Tan, A. Al-Shanoon, H. Lang and M. El-Gindy, "Mobile Robot Regulation with Image Based Visual Servoing," in *ASME IDETC/CIE*, Quebec City, 2018.

A. Al-Shanoon, A. H. Tan, H. Lang and Y. Wang, "Mobile Robot Regulation with Position Based Visual Servoing," in *IEEE CCECE*, Ottawa, 2018.

The writing of the manuscripts, data collection and research conducted in this work and the mentioned publication(s) are completed by the author. Co-authors reviewed and provided technical support when required.

ACKNOWLEDGEMENTS

Firstly, I would like to acknowledge both my thesis supervisors, Dr. Moustafa El-Gindy and Dr. Haoxiang Lang for providing me with technical guidance, financial support and most importantly, the opportunity to explore an area that is of great interest to me.

In addition, I would like to express my great appreciation for my father, mother and brother for their assistance which enabled me to focus solely on the research presented in this work.

Finally, I would like to thank Miga for her continued encouragement and support which kept me motivated over the years. Without her and the people mentioned, the work completed here would not have been possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

SECV	Scaled Electric Combat Vehicle
ROS	Robot Operating System
MWMS	Multi-Wheeled and Multi-Steered
MW	Multi-Wheeled
MS	Multi-Steered
LAV	Light Armored Vehicles
4WD4WS	Four Wheel Drive and Four Wheel Steer
6WD6WS	Six Wheel Drive and Six Wheel Steer
8WD8WS	Eight Wheel Drive and Eight Wheel Steer
AWDAWS	All Wheel Drive and All Wheel Steer
SLAM	Simultaneous Localization and Mapping
GPS	Global Positioning System
RTK	Real Time Kinematic
LiDAR	Light Imaging, Detection and Ranging
GNSS	Global Navigation Satellite System
INS	Inertial Navigation System
IMU	Inertial Measurement Unit

IBVS	Image Based Visual Servoing
PBVS	Position Based Visual Servoing
RFID	Radio Frequency Identification
IR	Infrared
QR	Quick Response
PTZ	Pan Tilt Zoom
FWS	Front Wheel Steer
4AS	4^{th} Axle Steer
AWS	All Wheel Steer
USB	Universal Serial Bus
DOF	Degree of Freedom
CAN	Controller Area Network
CAD	Computer Aided Design
EB	Elastic Band
TEB	Timed Elastic Band
DWA	Dynamic Window Approach
FP	Feature Points
SIFT	Scale-Invariant Feature Transform

CG	Center of Gravity
M-PBVS	Modified Position Based Visual Servoing
FV	Forward Velocity
DH	Denavit Hartenberg
RTI	Ramp Travel Index

NOMENCLATURE

$x^{[j]}$	State hypothesis
$w^{[j]}$	Importance weight
$u_{1:t}$	Laser scan observations
$z_{1:t}$	Odometry information
δ_o	Outer steering angle
δ_i	Inner steering angle
δ_L	Left steering angle
δ_R	Right steering angle
O	Turning center
L	Wheelbase
B	Track width
δ	Steering angle
v	Total longitudinal velocity
$\dot{x}, \dot{y}$	Linear velocities along respective axis
θ	Heading angle
$\dot{\theta}$	Yaw rate
θ_1	Revolute joint angle

d_2	Prismatic joint displacement
φ	Angle of velocity with respect to vehicle longitudinal axis
σ	Path curvature
R	Turning radius
r	Wheel radius
ω_R, ω_L	Right and left wheel velocity
$e(t)$	Error with respect to time
k	proportional gain
J	Jacobian matrix

Chapter 1. Introduction

The world of automobiles has experienced several milestones in its development since its inception in 1885 by Karl Benz [1]. From inventions such as automatic transmissions, satellite navigation to sensor-based cruise control, automotive engineers have produced several commercialized and innovative solutions that made traveling easier, more affordable and accessible. Today, the automotive industry is seeing its latest revolution centered around the automation of transportation systems. This revolution entails the retiring of old manual gasoline vehicles with driver-less electric ones to create a more convenient and safer way of travel. This is accomplished by vehicles becoming more intelligent and instrumented with sensors which creates better vehicle accessibility for those who cannot drive, and car sharing features that lessen traffic congestion. The potential to impact the streets for road users is unmatched; not to mention, its ability to generate numerous job and business opportunities around the world. These advantages along with several unmentioned ones quickly made autonomous vehicle technologies an extremely sought-after research topic. As a result, substantial efforts are made by automakers, technology companies and academic institutions to collaboratively accelerate the automation progress.

Presently, autonomous navigation features for vehicles with traditional configurations such as two axles and front wheel steering are studied and documented extensively with real-world deployment. However, multi-wheeled vehicles, which is defined as any vehicle with more than four wheels in this work, have not received nearly as much attention due to its limited market. This type of vehicle finds its applications

primarily in off-road and military settings because of their ability to traverse through rough terrains.

As the push for autonomous navigation capabilities continue, a shared space with mobile robotics begins to emerge due to comparable fundamentals. Likewise, mobile robots and scaled vehicle platforms are often smaller than their life size counterpart which permits researchers to conduct experiments in indoor laboratories. This is tremendously convenient when focusing on the different subsystem algorithms pertaining to mapping, localization and path planning since a full-size vehicle model is not always necessary during development. Nevertheless, commercial mobile robots sold today are generally equipped with differential drive setups and often lack car like features such as steering and suspension. This problem in combination with a lack of effort in exploring alternative steering methods to improve maneuverability ultimately hindered technology advancements in military vehicles. To further explore this area, the following thesis presents a literature review of the current state of the art in terms of combat vehicle design along with the limitations imposed by the current navigation methodologies. Motivated by this, a novel physical prototype with an independently steerable system is proposed in this work in an effort to generate research attention for this field.

1.1 Scope and Objectives

The primary scope of this work is to design and develop a novel autonomous Scaled Electric Combat Vehicle (SECV). Completion of this work includes design and manufacturing of mechanical parts, instrumentation of electronics as well as development and implementation of software. More specifically, the mentioned software will include both high and low-level control algorithms to enable precise autonomous point-to-point

navigation. The goal is to create a physical prototype that improves upon the current combat vehicle design and ability. The expected outcome is a fully functional robotic platform loaded with both custom and current software for precise navigation as well as future development.

The detailed objectives of this research include:

- Design and manufacture of the SECV which include a custom chassis, multi-output drivetrain, independent steering and suspension system as well as a modular exterior
- Electronic system architecture design and sensor instrumentation to enable remote control and autonomous navigation features
- Development of low-level software for vehicle actuation such as driving, steering and localization features
- Create a modular software development environment within Robot Operating System (ROS) for the proposed SECV
- Integration of mapping, path planning and obstacle avoidance algorithms
- Propose a precise pose correction algorithm based on the traditional visual servo controller to improve upon the limitations constrained by current navigation methodologies
- Combine the proposed pose correction algorithm with currently available global and local path planners to achieve precise docking
- Integrate all software developed with the SECV prototype for physical experimentation and validation
- Conclude on vehicle operational performance, navigation ability and improvements made by the proposed docking algorithm

1.2 Outline of the Thesis

Chapter 1 introduces the research area by discussing the motivation, scope and objectives of the work. In addition, this chapter also covers the working foundations of the research presented in this thesis.

Chapter 2 provides a detailed literature survey of relevant work which includes current armored vehicle designs, multi-axle robotic platforms, autonomous navigation methodologies and current docking algorithms.

Chapter 3 discusses the SECV prototype in detail by covering all aspects of the mechanical, electronic hardware architecture design and making of a modular software environment for Robot Operating System realization.

Chapter 4 explains the autonomous navigation methodologies that are applied in this research which include algorithms pertaining to mapping, localization, path planning and control. This chapter also includes a derivation of the SECV's kinematics model.

Chapter 5 proposes a close quarter pose correction algorithm that capitalizes on the steering ability of the SECV's design. The complete algorithm that combines high-level path planning with low-level vision-based control is also discussed in this chapter.

Chapter 6 presents the results from physical experiments on all software algorithms implemented and developed in this work. Comparison with traditional methods is discussed here.

Chapter 7 concludes the thesis by describing the benefits and limitations of the proposed research as well as recommendations for future works.

1.3 Working Foundations

The working foundations for the research presented in this thesis is discussed in the following section. This includes an overview of the traditional Ackermann steering geometry which is extended for multi-axle steering in this work. From there, an in-depth description of the Robot Operating System is presented to provide readers with an understanding of the underlying environment responsible for all software developed and implemented in this work.

1.3.1 Ackermann Steering Geometry

Beginning with Ackermann steering geometry which describes the relationship between steering angles and vehicle turning radius during low-speed scenarios. This geometry is critical in minimizing the lateral sliding of tires during cornering maneuvers [2]. By doing so, the overall usage time of the tires is prolonged because of reduction in scrubbing. In a traditional vehicle configuration as shown in Figure 1-1, the front outer and inners wheels must steer at angles δ_o and δ_i, respectively, such that the wheel heading axis is perpendicular to a line that intersects with the turning center, O. Through similar triangles, the steering angles are calculated as follows where the wheelbase and track width are denoted as L and B, respectively.

$$\delta_o = \tan^{-1}\left(\frac{L}{R + \frac{B}{2}}\right) \tag{1-1}$$

$$\delta_i = \tan^{-1}\left(\frac{L}{R - \frac{B}{2}}\right) \tag{1-2}$$

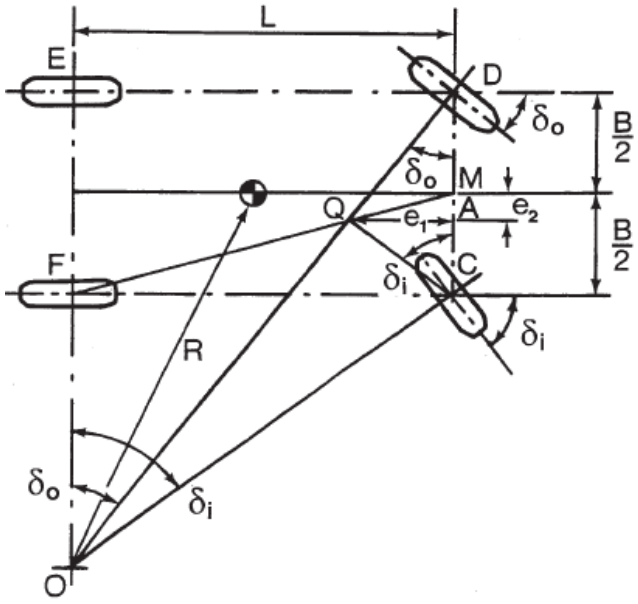

Figure 1-1 Ackermann Steering Geometry [2]

1.3.2 Robot Operating System

In 2007, a framework known as Robot Operating System (ROS) was released to the public to create a standard platform for researchers and enthusiasts to develop complete robotic systems [3]. Since then, the operating system has received several updates to include newer features and better stability. Currently, ROS offers a variety of useful tools and libraries that helps with hardware abstraction and low-level device control. One of the main tools is its node-based network that enables hardware and software communication. Other important tools include a frame transformation package along with simulation and data visualization applications known as Gazebo and RVIZ, respectively. These tools are described in detail in the following sections.

1.3.2.1 ROS Communication

As mentioned, one of the most imperative tools that ROS offers is its node-based communication method where hardware and software are represented as nodes and they are able to communicate with each other through topics. In a simple implementation as illustrated by Figure 1-2, Node A is able to publish a message to Topic 1 which Node B is subscribed to and in return, Node B is able to do the same with Node A via Topic 2. When comparing this setup to a generic motor control application, Node A and B may represent a motor controller and a motor with an encoder attached, respectively. In this case, the motor controller node can publish a desirable output speed to a topic called */motor_speed* which the motor node will subscribe to. On the other hand, the motor will receive feedback from its encoder and publish to an */encoder* topic which the motor controller is subscribed to. It is important to note that ROS provides different standardized message types that can be sent between components. When writing custom nodes, it is crucial to utilize these message types to ensure proper publication and subscription of data. By completing what has mentioned above, two different hardware components and their respective software codes can function in a closed loop manner. The advantage of such a node-based communication method is its modular nature which enables flexible additions of new hardware and software at any time given that the appropriate topics and message types are utilized.

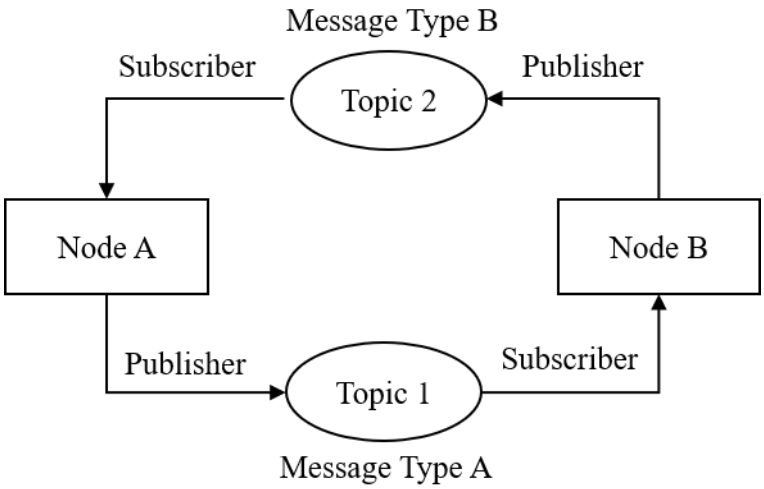

Figure 1-2 ROS Nodes and Topics

1.3.2.2 Coordinate System and Transformation Frames

To unify all development of tools and libraries, ROS employs a standard cartesian coordinate system and frame relationship to describe the relative position and orientations of a mobile robot in its workspace. To define this relationship, ROS implements a parent and child tree structure between frames as illustrated in Figure 1-3. At the highest level is the earth frame which serves as the parent for the map frame. The primary intent of this frame is to enable the use of multiple maps for robots that exist in different environments to coordinate with each other. If only a single robot is used, the earth frame is often not necessary as the map frame is able to serve as the absolute reference point. Moving forward, the map frame is a discrete and fixed frame that does not drift with respect to the earth frame. Since it is discrete, the position of mobile robots within this frame will experience sudden changes as new sensor data continues to update the robot pose estimation. On the contrary, the odom frame is a continuous frame that is based on the position at which a robot initializes in the map. The basis of this frame is highly dependent on the odometry

sensors of the robot; therefore, it experiences drift over time. Due to their respective pros and cons, the map and odom frames are only useful for long and short term navigations, respectively. Lastly, the frame that represents the mobile robot's position and orientation within the map and odom frame is denoted as the base_link. This frame is often attached to the center of the robot base component where the *x* axis signifies the heading direction. Although not shown in Figure 1-3, it is recommended to define further child frames for base_link for all additional hardware components and sensors to ensure that the mobile robot is aware of its own configuration. For example, if a robot features a laser scanner that is located one meter in front of the base_link frame, the robot would have to account for this difference when interpreting incoming range data. All coordinate frames discussed in this section are essential for proper implementation and development of mobile robot navigation algorithms.

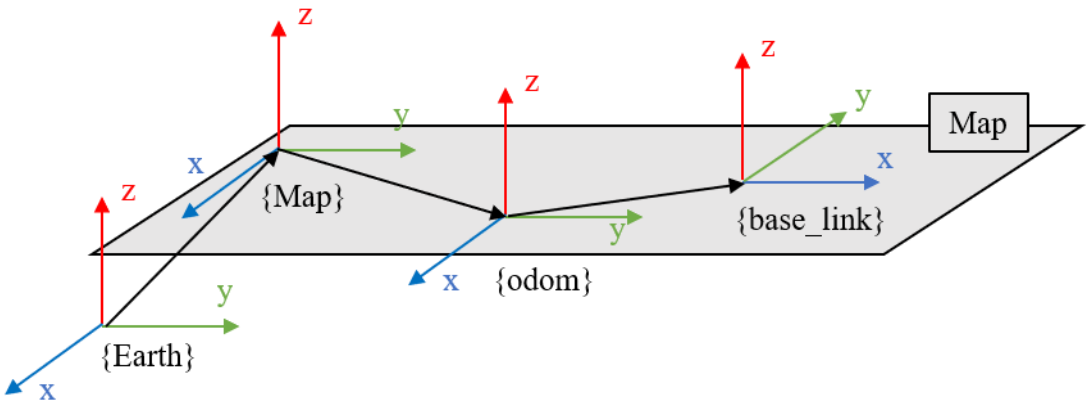

Figure 1-3 ROS Frames of Interest

1.3.2.3 Gazebo

Gazebo is an application within ROS that offers a simulation environment for mobile robot software development. In this environment, the physics of the robot and its surroundings are considered to mimic the real world. Often, researchers use Gazebo to test their software algorithms before physical integration to ensure that all working parts are functioning correctly. Besides, this application also allows people without access to physical robots to conduct research and contribute. The following figure shows an environment for a mobile robot that was custom created. There is a wide range of items that may be used to simulate different outdoor and indoor settings. It is also possible to create a custom model for the robot as well. In this work, Gazebo is utilized to ensure that all software is working before integrated with the SECV.

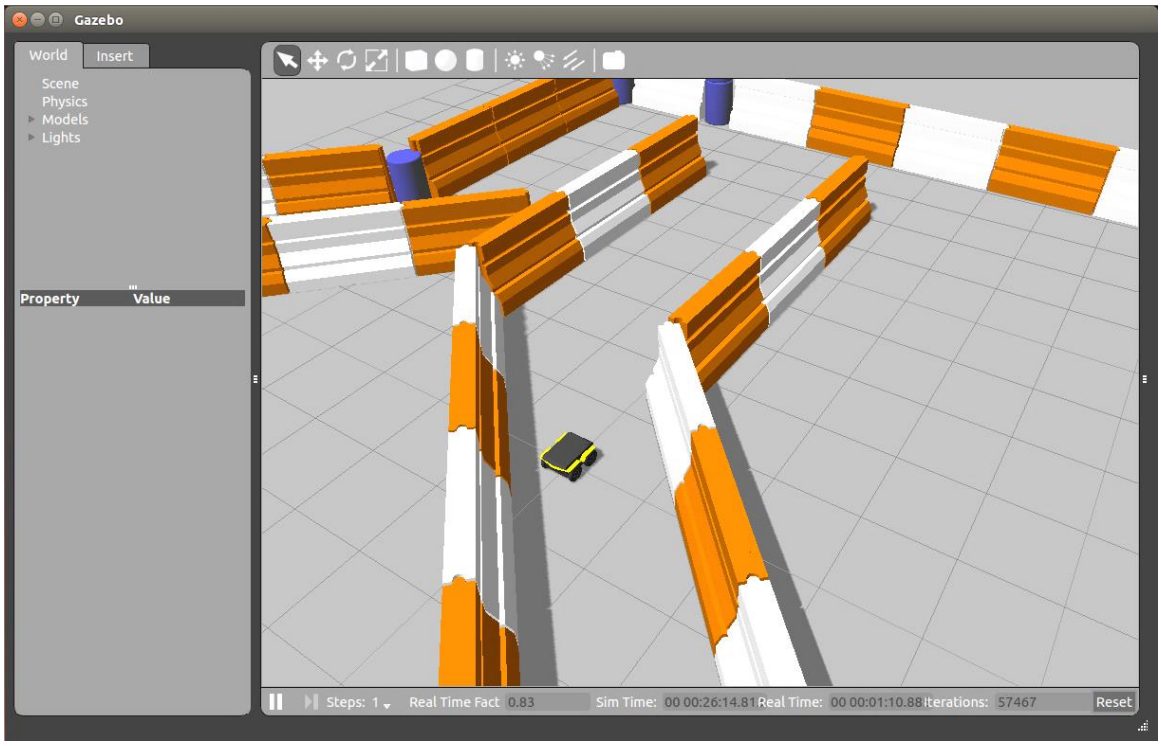

Figure 1-4 Gazebo Environment for Jackal Robot [4]

1.3.2.4 RVIZ

RVIZ is an application that is used for data visualization. In a typical workflow, the user would use either Gazebo to simulate a robot and its environment or launch a physical robot. From there, RVIZ subscribes to the different topics that are published and represent them visually within this application. For example, the different transformation frames as well as the incoming data from range sensors are seen in the figure below. This allows the researcher and operator to see what the robot percepts in both the simulated environment and real life. This application is referenced in the experimental section of this thesis.

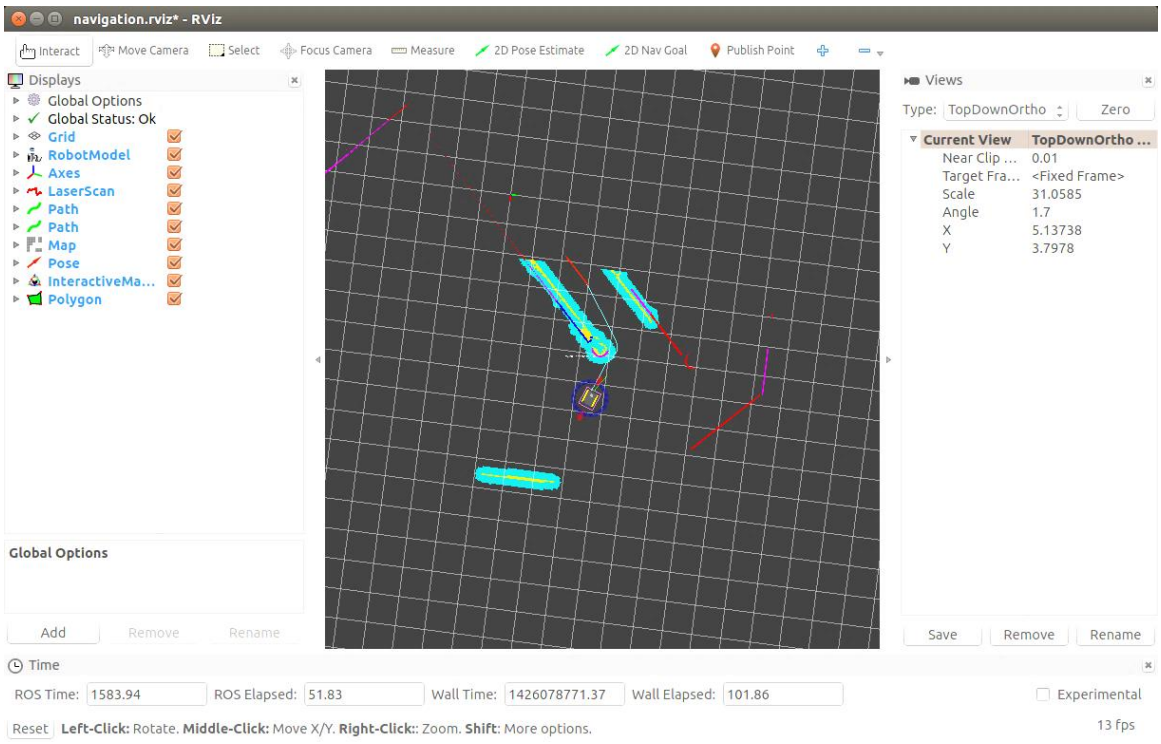

Figure 1-5 RVIZ for Jackal Robot [4]

Chapter 2. Literature Review

2.1 Introduction

In this chapter, a survey of all relevant topics covered in this research thesis is presented. These topics include the current state of the art in terms of military vehicle design, multi-axle vehicles and the implementation of multi-steered systems. Beyond the current mechanical design and features, a review of autonomous navigation methodologies is presented with a focus on the three pillars of navigation which are mapping, localization and navigation. This will lead in to a discussion on recent control and navigation approaches that are implemented with multi-wheeled and multi-steered (MWMS) platforms. Lastly, a review of close quarters pose correction algorithms for mobile robot docking is presented. By the end of this chapter, a summary of the current design and technologies is described with research gaps identified to further validate the motivation of this work.

2.2 Light Armored Vehicles

Because of its ability to maneuver in rough terrains, multi-wheeled vehicles often find its applications in off-road environments such as military and space exploration. In this work, the primary focus is military based with an emphasis on the Light Armoured Vehicles (LAV) designed by General Dynamics Land Systems in Canada [4]. This type of vehicles finds its principal usage in infantry fighting and personnel transportation. Throughout its history, there have been several installations to this family of vehicles which all share the eight-wheeled design. The first member of the LAV family dates to 1983 with the release of the LAV-25 [5] where it features an amphibious design with front wheel steer. In 1999, an updated design known as the LAV III was released with new features

such as a high management system, central tire inflation system and improvements in maximum payload capacity and protection [6] [7]. This vehicle was in service for the next 16 years until 2015 when the government of Canada decided to upgrade all LAV III's by implementing several new features such as an upgraded suspension system, higher horse power engine, better protection and a complete switch to digital electronic systems within the vehicle [8]. The improved LAVs, known as LAV 6.0, entered service in 2015 and are expected to remain in duty until 2035. Currently, a new LAV model known as the LAV 700 (as seen in Figure 2-1), is scheduled to entered production [9]. This vehicle features an improved control architecture when compared to LAV 6.0 and boasts other innovative features such as self-sealing fuel tanks, modular protection system and other advancements in engine horsepower.

Figure 2-1 Light Armoured Vehicle (LAV 700) [10]

2.3 Novel Scaled MWMS Robotic Platforms

Recent work in MWMS platforms is divided into implementations of either multi-wheeled or multi-steered with limited literature available on the combination of both. Starting with the design and development of novel multi-wheeled platforms, recent publications on six-wheeled robots are proposed in [11-13]. In these papers, the primary application is focused on space exploration robots. Agriculture applications can be found in [14] where a novel platform operates in a garden for plant identification and classification using neural networks. The goal is to recognize and determine the amount of water and fertilizer necessary to facilitate optimal growth. The above-mentioned papers are fixated on six-wheeled designs equipped with rocker-bogie suspension and differential drive trains. In terms of eight-wheeled setups, [15] proposes a platform with a passive planetary, overlapping wheel structure that is implemented to improve chassis orientation relative to the roughness of the terrain. Like the space rover designs, this vehicle also utilizes differential drive for steering. Another creative solution is proposed in [16] with a biomimicry-based solution to steer a multi-wheeled robot like a snake for rescue missions. This design enables the robot to traverse through tight and rough environments because of its elastic trunk that is capable of active bending and passive compliance.

Although the mentioned papers thus far feature multi-wheeled drive trains; none share similarities with traditional vehicles in terms of steering like the work proposed in [17]. In this paper, the authors describe a four-wheel drive, four-wheel steer (4WD4WS) robot that is embedded with a fuzzy logic controller for path tracking. Another group of researchers from [18] developed a similar platform with added features for lane following, reverse and parallel parking using machine vision and fuzzy controllers. For soil sample

collection and fertilizer dispensing, [19] proposes another 4WD4WS platform a new extended Ackermann steering principle introduced. Other efforts in this area include [20] where a single camera vision system is used for navigation. In this work, the authors considered image processing, control and path patterns. Since robots with independently steered wheels are theoretically capable of multiple steering modes, comprehensive analysis for a 4WD4WS platform is described in [21]. The steering modes studied here include front wheel, all wheel, crab and diamond steer.

2.4 Autonomous Navigation Methodology

One of the main research topics of this thesis pertains to the autonomous navigation of mobile robots, which can be broken down in to three categories; namely, mapping, localization and navigation. In mapping, the different types of maps and mapping algorithms are discussed in this section. Once the map is successfully acquired, different localization algorithms are explored to estimate the position of a mobile robot within the given map. When the robot is successfully localized within the given map, it is then ready to begin path planning. Visual servo control is an alternative method to navigate a robot without a map and it is a technique that is relevant to the proposed pose correction algorithm presented later in this work. All relevant literature published within the past decade regarding each of these navigation methodologies is presented in this section. Figure 2-2 provides a clear illustration of the different topics covered.

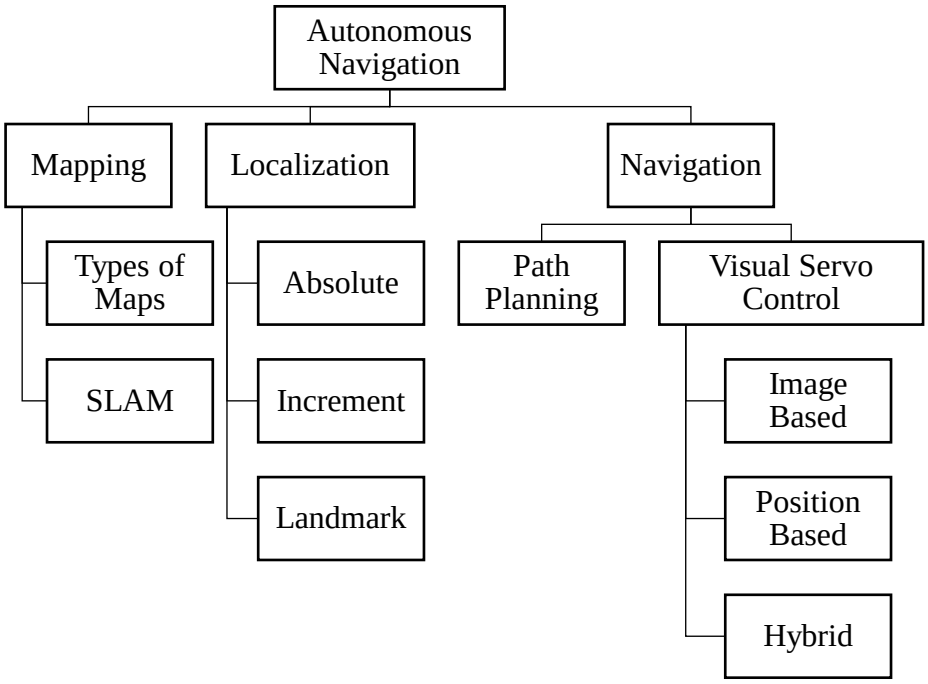

Figure 2-2 Autonomous Navigation Methodology

2.4.1 Mapping

Starting with recent literature involving mapping, this section will cover the different types of maps and mapping algorithms.

2.4.1.1 Types of Maps

The two primary types of maps are metric and topological; where the former includes accurate metric information between locations and the latter describes location relationships through nodes and connections. A combination of these two types of maps with the addition of semantic information for a more meaningful representation is possible [22].

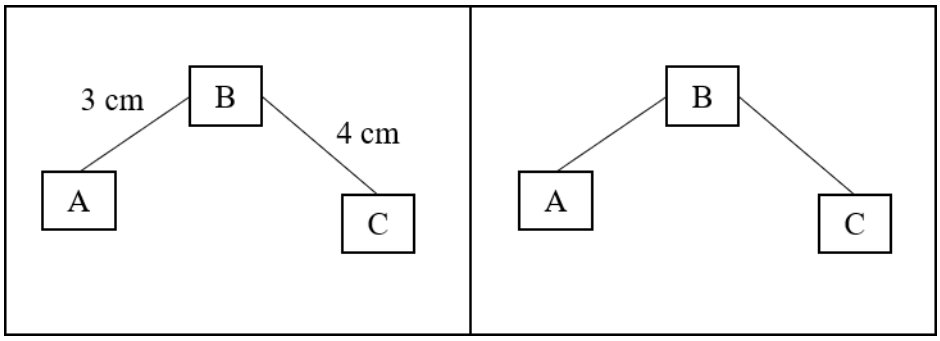

Figure 2-3 Metric Map (left), Topological Map (right)

Other forms of map representations exist in both two and three-dimensional forms. Researchers are constantly looking for efficient methods to represent the environment to achieve optimal performance based on the types of applications. Some examples include [23], where an efficient algorithm is introduced by the authors for a mobile robot to navigate indoors with a topological map. The proposed algorithm is lightweight which ensures adoptability for robots with onboard resource constraints. A different type of map is proposed in [24] where a laser range finder is used to generate a "travers-ability field histogram" based on irregularities in the ground plane to determine the robot's motion. In [25], the authors present a triangular cell-based map that offers greater navigation freedom when compared to regular rectangular cell-based maps.

2.4.1.2 Mapping Algorithms

For mapping, Simultaneous Localization and Mapping (SLAM) techniques are commonly used. This technique is a map-building navigation approach where a robot explores and builds a map of the unknown environment. It does this by simultaneously estimating its own pose relative to landmarks and mapping the environment by creating loop closures when revisiting old landmarks. SLAM is extremely common and heavily

studied in the field of mobile robotics and readers are recommended to read the literature in [26] for fundamentals while Durrant-Whyte and Bailey's two-part tutorial covers the theoretical background in detail [27], [28].

In some scenarios, the SLAM problem is considered solved while issues still exist in others. For example, using a laser range finder to build a two-dimensional map in a static environment with a mobile robot is considered a solved problem; however, SLAM with semantic information is still considered unsolved. The authors in [29] do a great job in introducing the readers to the past and present state of SLAM while providing a forecast to what will be the future. The following subsections will cover recent work in SLAM applications with Light Imaging, Ranging and Detection (LiDAR) sensors and laser scanners as well as multi-sensor fusion based approaches. Although SLAM methods cover a broad range, the two mentioned are what relates to the research presented in this thesis.

A. LiDAR: Laser-Based SLAM

In this section, SLAM is completed with either a LiDAR or laser sensor which can produce both two and three-dimensional maps. Although computationally more complex, three-dimensional point clouds are popular in recent literature due to the more complete knowledge of the surroundings when compared to two-dimensional maps. In [30], the authors developed an algorithm capable of determining ideal positions to scan a construction site to produce the most comprehensive map with the least amount of time. For rough terrain navigation, [31] proposes a system with a continuously rotating laser scanner that creates a three-dimensional map. The algorithm builds a local dense map of adjoining surroundings to enable six-dimensional robot pose tracking in real time. The experimental work was conducted in the DARPA Robotics Challenge.

One of the main issues with SLAM is the relocation problem sometimes referred to as the “kidnapping problem”. This happens when the robot is picked up and placed in a different location within the map during its process. This problem cannot be solved by basic SLAM implementations; however, a recent solution for this problem appears in [32] where a deep learning algorithm is presented to utilize LiDAR range data to predict the relocated position. This approach is computationally heavy; therefore, processing is offloaded to cloud servers. The results yielded a promising success rate.

Besides the kidnapping problem, laser range finders also suffer from faulty readings when the environment contains transparent or reflective materials. This issue is explored in [33] where a method of determining whether a laser reading is affected by reflective or transparent objects using a mirror detector and reflection classifier is introduced. By doing so, the affected readings are eliminated during the SLAM process to improve the accuracy. An additional solution to reduce faulty and noise readings generated from laser range finders can be found in [34].

Contrary to computationally demanding algorithms, an efficient SLAM algorithm is demonstrated in [35] where 3D point clouds are projected on to 2D planes to create planar surface segments. This enables the algorithm to reduce the amount of data processed and saved which consequently lessens operation time and memory usage. As most SLAM algorithms focus on low-level features such as points and lines, [36] focuses on higher level representations in the form of rectangles which also decreases the demand for onboard resources. Additionally, the authors in [37] proposed two algorithms that work in junction to reduce computational complexity. In one algorithm, the laser scanner data is used to estimate velocity at higher frequency while the second one runs at lower frequency to

produce a fine point cloud. Another low demand algorithm is presented in [38] where grid maps are used to interpret the range data.

B. Multi-Sensor Based SLAM

While the previous section utilizes either a laser scanner or LiDAR for SLAM, this section covers applications that fuse multiple sensors to achieve superior performance. This type of SLAM provides redundancy in the case of noisy or faulty readings that are present in any individual sensor. For example, the authors in [39] introduced a system that fuses laser with an omnidirectional camera. This system is accompanied by software that utilizes scan matching to ensure convergence in tracking with limited resources. In addition, scan matching is also completed in [40] where the authors combined data from an advanced sonar with a laser range finder to complete Voronoi graph-based exploration during SLAM. Next, the authors in [41] and [42] used a combination of laser and stereo cameras to identify tables instead of just the legs since laser scanners are two dimensional. Furthermore, cameras can also be used to detect visual features in the environment that may aid in adding semantic information to the generated map. To go one step further, wheel odometry is added to the laser and stereo camera fusion in [43]. This algorithm is capable of SLAM in addition to dynamic objects and people detection in the workspace. The authors in [44], [45], [46] all propose systems that combined laser scanners with RGB-D sensors to create non-computationally demanding algorithms. Although RGB-D cameras are already able to determine depth, the accuracy is improved with the addition of a laser sensor. On the other hand, [47] proposes a method that utilizes artificial beacons and a gyrocompass to reinforce the robot pose estimation accuracy during the SLAM process. Another example of SLAM using absolute localization includes [48] where information from Global Navigation

Satellite System (GNSS), Inertial Navigation System (INS), and LiDAR are combined to achieve an accurate localization. Currently, one of the most complete implementations of multi-sensor SLAM is presented in [49] where an event camera and a standard camera is combined with an Inertial Measurement Unit (IMU) to attain a robust and accurate state estimation during SLAM. The event camera is advantageous during high speed and dynamic range scenarios, whereas the standard camera provides accurate information during low speed and regular lighting scenarios. Furthermore, the IMU is used in the case of low light conditions. The combination of these three sensors enables the algorithm to perform accurately under a wide variety of environmental conditions.

2.4.2 Localization

A key problem to address in mobile robotics research is localization within the provided map. In localization, the three key methods are absolute, incremental and landmark localization. Each of these three methods is reviewed in the following sections.

2.4.2.1 Absolute Localization

In absolute localization, the position of the robot is determined explicitly using Global Positioning Systems (GPS), magnetic fields, beacons or similar methodologies. This type of localization requires additional equipment to be available in the workspace to communicate with an onboard receiver for robot position updates. One of the first implementations of absolute localization in mobile robots is described in [50]. When the signals are strong, absolute localization commonly yields the best results when compared to other localization methods; however, there is still room for improvements. In recent literature, this improvement frequently comes in the form of sensor fusion as there is not a single sensor that can provide robust performance in all scenarios. Outdoor and indoor

applications are studied separately for absolute localization as some sensors used for one are not functional for the other.

Starting with absolute localization in outdoors applications, the authors in [51] introduce an algorithm where a GPS and an inertial navigation system (INS) is combined to improve the localization robustness. In this work, a fault detection and isolation algorithm are also derived to verify GPS signals while the INS serve as feedback in the case of lost signals. In [52], GNSS is used in combination with 3D maps provided from vendors such as Google Earth to improve the localization performance in the presence of tall buildings that obscures signal strength. Another outdoor solution includes [53] where the fusion of IMU and a real-time kinematic GPS (RTK-GPS) is proposed to eliminate error accumulation over time. This algorithm showcases redundancy in design as it relies on the IMU in the case of lost signals.

Since GPS does not work indoors, an inertial sensor is combined with a LiDAR sensor and the received signal strength (RSS) in a wireless local area network is proposed to provide accurate localization [54]. In another work, a Cricket System based on distance measurements between transmitters and receivers developed by Massachusetts Institute of Technology is improved and implemented for an omnidirectional robot in [55]. Besides received signal strengths and distance measurements, other methods of indoor localization exist such as using magnetic fields [56] [57]. In these methods, different locations are identified by unique magnetic signatures.

2.4.2.2 Incremental Localization

Incremental localization is often referred to as dead reckoning in published literature. In this method, the initial position of the mobile robot in the given map must be known. As the robot starts to move, the sensor values are incrementally summed to predict the robot's latest pose. This method is prone to error accumulation. Some sensors used for this method include wheel encoders, cameras and laser range finders which are the primary drivers behind the wheel, visual and laser odometry, respectively.

Wheel odometry is popular due to its cost and ease of integration. It works by counting and tracking wheel revolution and speed in conjunction with the robot's kinematic model to estimate robot pose within the provided map. For example, the authors in [58] describe a method for vehicle positioning that is independent of GPS signals and only uses wheel speeds. In recent applications, the authors in [59] developed a wheel odometry algorithm for curved surfaces. Formulation of the solution for different curvature surfaces is derived in this work. To further improve the performance of wheel odometry, wheel revolutions are fused with INS by a Kalman filter to increase the localization accuracy in [60].

On the other hand, visual odometry works by comparing consecutive images from a camera feed to estimate the robots change in pose based on the change in image that is induced by motion. This technique has improved with advancements in camera sensors since its inception in 1991 [61] and now it is frequently considered to be a better alternative to wheel odometry in terms of accuracy. Recent work in visual odometry include [62] where one single camera is used to localize a vehicle globally. This algorithm implements a probability function to compensate for potential drift and scale ambiguity. In [63], a real-

time 6 degree-of-freedom (DOF) localization method is introduced with a single monocular camera. However, the localization process in this method happens in two separate stages with the first one garnering topological information followed by refinements using the derived metric information. Another visual odometry solution is proposed in [64] where the authors used two cameras to mitigate uncertainties with image noise and map ambiguities. The mentioned method can localize the robot in an unknown environment. This type of localization technique is extremely beneficial for omnidirectional robots as wheel odometry would be useless due to constant wheel slippage.

Besides wheel and visual odometry, a third type of odometry using laser range finders is explored in [65]. In this method, the authors match vertexes in the acquired range with that of a given map to localize the robot. The authors in [66] used laser to improve wheel odometry. The algorithm works by measuring the distance to a reference point at the beginning of the navigation. As the robot moves to its next checkpoint, the algorithm computes where the laser pointer should theoretically point to get the same reference reading. The difference is used to estimate its new pose. While laser scanners provide the most accurate metric information out of the three types of incremental localization, it is also the most expensive.

2.4.2.3 Landmark Localization

In this method, the map of the environment along with information regarding specific landmarks are provided for the mobile robot to accomplish localization. The sensing of the landmarks is generally done through visual features. For example, a set of appearance-based landmarks are provided along with their respective positions in a map in [67]. The mobile robot estimates its own pose relative to the landmarks and consequently, obtains

the global location. This idea is expanded further in [68], where landmarks with heterogenous features such as lines, points, planes and vanishing points are considered. To increase the field of view, a fisheye vision system is proposed in [69] where only two landmarks are necessary to localize the robot. A novel contribution from this work is its ability to solve the relocation problem which is when a robot is abruptly picked up and placed at a different location. In this implementation, relative position and orientation to the landmarks are used to approximate possible robot locations. A weighted distribution function is used to choose the right location. In another implementation, a hand-sketched map and a semantic map are combined in [70] and given to a mobile robot for navigation. Visual tags are used to enable robust recognition by the mobile robot. This algorithm further improves the localization accuracy by also considering wheel odometry. For autonomous vehicle applications, urban landmarks such as traffic lights, street signs and other street features are used in [71]. In this approach, the vehicle performs localization at intersections by matching the mentioned landmarks with a digital map. The main contribution from this work is the ability to determine the current lane that the vehicle is in based on visual and map information. While the previously mentioned articles cover the different applications of landmark localization, [72] illustrates the most effective way to place landmarks within a robot's workspace for top performance. The proposed algorithm can work for different robots in different environments by outputting the recommended number of landmarks and their locations.

2.4.3 Navigation Methods

Navigation is the last stage after a map is generated and localization is completed. In this section, different path planning approaches in current publications are presented and discussed. Furthermore, literature regarding a method known as visual servo control is also reviewed as it is an important part of the pose correction algorithm proposed later in this research.

2.4.3.1 Path Planning

Path planning is a task divided between the global and local planner where the former is responsible for generating consecutive waypoints that consists of position and orientations based on the acquired map and the latter is tasked with reaching the waypoints while considering sensor inputs. Literature regarding the two types of planners is discussed in the following two subsections.

A. Global Path Planning

The first global path planner known as the Dijkstra's Algorithm was developed in 1959 [73]. This path planner is based on representing a map with nodes that are assigned values based on the cost of arriving. The goal of the algorithm is to find the shortest path between the starting and finishing node. Approximately ten years after, researchers from SRI International improved Dijkstra's algorithm by implementing priority-based heuristics that dictates the search for the shortest path. This method is known as the A* Search Algorithm and it is more efficient than Dijkstra's algorithm because it does not search every node [74]. A few decades after A* was developed, an algorithm known as the Artificial Potential Field Method was invented [75]. In this algorithm, the main concept is to treat

the mobile robot and the goal destination as opposite polarities. By doing so, the robot would be attracted to move towards the intended goal. On the other hand, obstacles are assigned the same polarity as the robot which would subsequently generate a repulsive force between each other. The result of this is a real-time obstacle avoidance algorithm that generates robot motion from start to finish. Lastly, in 1994 another path planning algorithm that builds on top of the A* was developed [76]. This method is called the D* algorithm which got its name from "Dynamic A*" as the improvements made came in the form of changing costs. It is worth mentioning that there are other global path planning algorithms excluding the four mentioned here; however, these are the ones that are currently popular within the robotics area.

B. Local Path Planning

Once a global planner has finished generating a path between the start and finish, local path planners are responsible for considering real-time sensor data to avoid collision with obstacles that were not present during the mapping phase. A notable local planner known as the Elastic Band (EB) method was introduced in 1993 [77]. In this concept, an elastic band is imagined taking shape of the global plan at the start of the navigation. As the robot traverses, this elastic band is deformed and stretched to accommodate for new obstacles. A few years after the release of the Elastic Band planner, an algorithm known as the Dynamic Window Approach (DWA) was introduced. This algorithm is designed with dynamics of the robot in mind as it samples a search space around the robot for achievable velocities that does not result in a collision [78]. The search space in this case is limited by a time interval that may be set by the user. Fast forward another decade, a recent local path planner known as the Timed Elastic Band planner (TEB) was proposed in 2012 [79]. This

algorithm builds on top of the Elastic Band approach by considering an objective and cost function instead of forces that cause deformation to the bands. By considering the time element, this planner also considers dynamic constraints of the mobile robot, which the original Elastic Band planner did not. In addition, an advantage of this planner is its ability to plan for all mobile robot drivetrain configurations such as differential, omnidirectional and Ackermann systems. TEB is the current state of the art local planner for mobile robotics.

2.4.3.2 Visual Servo Control System

Aside from the path planners that take advantage of the acquired map and localization process, there is a category of navigation method that does not require the map. One of these methods is known as a visual servo controller which is presented in the following section.

In 1979, a type of vision-based control system known as a visual servo controller was introduced [80]. This type of controller is often referred to as “visual servoing”, which describes a closed loop system that controls the motion of a robot through visual feedback. This is a multi-disciplinary field of study that encompasses both computer vision and control theories with applications found in both manipulator and mobile robotic systems.

There are two fundamental approaches within visual servoing which are based on either 2D image or 3D position control. It is worth noting that a third category of visual servoing known as hybrid visual servoing exists by combining characteristics of the mentioned two methods. Besides where the control scheme happens, visual servoing is also characterized based on camera placements. The two types of camera placement are known

as hand-in-eye and eye-in-hand. Although the eye-in-hand configuration is the most popular in mobile robotics, each configuration exhibits its own pros and cons.

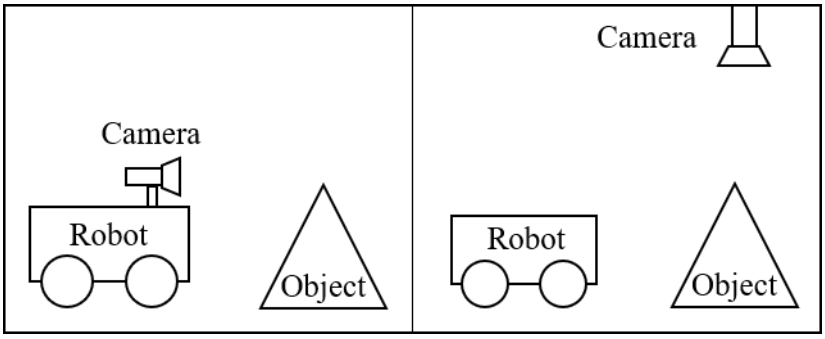

Figure 2-4 Eye-in-Hand (left), Hand-in-Eye (right)

Since depth is required in certain visual servoing applications; therefore, visual servoing can also be categorized based on the chosen method of depth estimation. In some cases, depth is determined explicitly using a range sensor whereas other cases include solutions that involve epipolar and trifocal tensor geometries [81], [82]. For background fundamentals regarding this topic, readers are recommended to read the two part comprehensive tutorials by Chaumette and Hutchinson in [83], [84]. The following section will review recent literature surrounding image based, position based and hybrid visual servoing.

A. Image-Based Visual Servoing

Recent literature in image-based visual servoing (IBVS) is focused around the regulation of robot pose and path following. Tracking is also common for visual servoing; however, supported literature using image-based techniques are limited. The following will cover each of these three types of applications.

Regulation is the task of controlling a robot's position and orientation in order to reach a desired location with reference to an appearance-based landmark. A basic implementation of IBVS is described in [85]. As mentioned earlier, there are different methods of estimating depth. For example, epipolar geometry is applied in [86] while trifocal tensor geometry is applied in [87]. Besides depth estimation, the regulation task is also subjected to limitations such as workspace and visibility constraints. These two types of constraints are both considered in the design of a predictive controller in [88]. In this case, the workspace constraint is caused by the nonholonomic nature of the robot while the visibility constraint is caused by the field of view of the camera. To tackle the visibility constraints, some have chosen to propose ways to keep image feature points (FP) within the image frame while others choose to increase the field of view of the camera. For example, a linear quadratic regulator control law is proposed for a mobile robot to keep FPs within the field of view in [89]. Alternatively, a method of increasing the field of view include utilizing a pan-able camera which is proposed in [90]. In this work, two control modes for the regulation and control of the pan camera angle are combined in a switching control algorithm. Another method of increasing the field of view is to use omnidirectional cameras. Recent works include [91] where three different homing algorithms are proposed based on different feature parameters. In this work, Scale-Invariant Feature Transform (SIFT) features are used to derive the control law. Another omnidirectional camera application is described in [92] where a radial model is implemented. Since every mobile robot and camera exhibit unique intrinsic and extrinsic parameters, an algorithm that is independent of these parameters would yield great freedom and flexibility across different setups. This issue is addressed recently in [93] where an algorithm that is independent of

camera parameters is proposed. Another contribution from this work is that the camera is placed in a hand-in-eye configuration on the ceiling to complete the regulation task. Conversely, [94] discusses an adaptive controller for an eye-in-hand configuration that operates without knowing the height difference between the camera and the visual landmarks. To improve the robustness, researchers have been proposing advanced controllers such as the one in [95] where a real-time inverse optimal neural control system is designed. Other contributions from this work include robot dynamics realization within the controller. Another advanced work published recently involves the development of a robust tube-based model predictive control scheme which is proposed in [96].

Besides regulation, path following is another common IBVS task studied in the past decade. Path following can be categorized into two separate types where one is the following of a path as drawn on the ground while the other is the following of a path as defined by reference images during a prior teaching phase. For the rest of this article, the first kind of path following will be referred to as physical path following while the second kind is referred to as visual path following. First, an algorithm for a nonholonomic mobile robot to reach and follow a path drawn on the ground is described in [97]. One of the novelties of this work is its simple requirement of two path features which improves the efficiency of the algorithm. Another path following work that exploits line features as oppose to point features is presented in [98] a few years later. This method avoids the difficulty in detecting and tracking geometric features in an image.

On the other hand, a visual path following is demonstrated in [99] where only a single feedback measurement from either epipolar or trifocal tensor geometry is necessary. In [100], the authors used line segments from visual memory as landmarks for visual path

following. This approach is limited by the performance of the feature extraction algorithms; however, the same authors improved this work a few years later when they combined line features with points to increase the robustness of the algorithm [101]. Like the regulation task, using an omnidirectional camera increases the field of view which enables greater navigation freedom. In [102], the authors propose a method that tracks photometric information of an image as oppose to visual features with an omnidirectional camera to accomplish the path following task. This method has the advantage of not depending on a feature extraction algorithm. Alternatively, omnidirectional cameras are also used for visual path following as shown in [103]. Since obstacle avoidance does not exist natively in visual servoing, several authors propose their methods of including this feature. For example, [104] discusses a road lane following algorithm that is capable of obstacle avoidance by using LiDAR. In this work, the control outputs are validated in an image-based dynamic window approach. For visual path following, a Lidar is also used to avoid both static and dynamic obstacles during the navigation process as presented in [105].

The last of the three major tasks for IBVS is tracking which is generally used for target following. As mentioned earlier, the available publications on this topic are limited within recent literature. An example of this is demonstrated in [106] where a controller for a mobile robot equipped with an active camera system is designed to track a dynamic target. This algorithm is based on a dual-Jacobian visual interaction model and it can work through partial occlusion of the target. Another work includes [107], where an online controller tracking controller is proposed. One of the main ideas of this work is that the authors focused on visual feature motion that happens because of the target object motion which enabled precise target following.

B. Position Based Visual Servoing

Similar to IBVS, the primary tasks for position based visual servoing (PBVS) is regulation, tracking and path following as well. Recent literature focuses on regulation, while papers for tracking and path following are limited. It is also worth mentioning that the amount of interest from researchers in PBVS for mobile robotics is considerably lower than that of IBVS across conferences and journals. Due to this, relevant works that include unmanned underwater and aerial vehicles are also included in this section's review. The following section will review the papers available from the past decade for each of these tasks.

As mentioned previously, regulation is the task of controlling the robot's position and orientation as it reaches the desired pose. When comparing to methods such as dead-reckoning, PBVS offers greater accuracy. In [108], the authors present a PBVS controller based on Lyapunov Functions that utilizes two points and a complete kinematics and dynamics model for the mobile robot. As mentioned in the visual servoing section, one of the methods of pose estimation is through epipolar geometry. The authors in [109] utilize this geometry in combination with an omnidirectional camera to develop an efficient controller based on a novel observability analysis. In [110], a PBVS controller designed for an autonomous underwater robot is used to automatically dock a charging station to increase operation time while submerged.

Since FPs are not controlled on the image plane, the field of view constraints are more prominent in PBVS than IBVS. In [111], the authors consider both camera motion and field of view constraints. In this work, an optimization problem is presented for global camera motion and a solution is provided. Another work that incorporates all constraints

along with the realization of both kinematics and dynamics of a mobile robot in a PBVS controller is developed recently in [112]. This work implements an RGB-D sensor for depth information and uses a model predictive controller that is improved by neural network to achieve effective regulation results. In the case that the reference landmark is occluded from the image, an algorithm that utilizes Extended Kalman Filter to combine data from odometry sensors and PBVS is proposed in [113] to solve this problem. For unmanned aerial vehicles, an algorithm that replaces the GPS in pose regulation is described in [114]. This method is beneficial in situations where GPS signals are weak or non-existent because it only utilizes landmarks placed at ground level as reference objects. To further improve the performance, a PBVS algorithm that exploits high-speed perception is presented in [115].

For tracking, the authors in [116] propose an algorithm to track a moving object by a mobile robot. Different geometric shapes such as circles, triangle, polygons and more are detected automatically and then a robust PBVS controller is applied to track the object. This work can be expanded to track humans. For path following, [117] demonstrates a mobile robot capable of reaching and then following a path as drawn on the floor. A noteworthy contribution from this work is its requirement of only a few visual features to ensure convergence which lessens the computational load. This work is expanded in [118] where the same authors improved the results while offering a comparison between image-based and pose based solutions to accomplish the same task.

C. Hybrid Visual Servoing

IBVS and PBVS both have their disadvantages. For example, the pose of the mobile robot is not considered explicitly in three-dimensional space which sometimes leads to

unstable commands in IBVS. For PBVS, the FPs are not controlled on the image plane; therefore, it is possible for them to leave the field of view which will cause system failure [119]. Because of this, hybrid visual servoing was first proposed in 1999 as "2 ½ D Visual Servoing" to combine the advantages of both methods to achieve a more robust solution [120]. In this work, the orientation and position are controlled by PBVS and IBVS, respectively. In hybrid visual servoing (HVS), the control law considers both three-dimensional coordinate space as well as the two-dimensional image space. In recent literature, hybrid visual servoing is often times referred to as homography based visual servoing [121] and its first application in mobile robots can be found in [122]. In this section, recent publications from the past decade are reviewed for the three different tasks; namely, regulation, path following and tracking.

In regulation, a basic approach is presented in [123] where a hybrid visual servoing algorithm is proposed to estimate the robot's motion. Since the algorithm works with low amounts of visual features, the field of view constraint is not a major concern. In addition, this algorithm uses a composite error vector to combine the advantages of IBVS and PBVS. For indoor navigation, [124] showcases an algorithm where an omnidirectional mobile robot is able to utilize PBVS and IBVS for global and local fine navigations, respectively. To improve the system's robustness, the authors in [125] introduce a complex HVS setup where a hand-in-eye camera is mounted on the ceiling of the workspace, and an eye-in-hand Pan-Tilt-Zoom (PTZ) camera is installed on the mobile robot. In this setup, the ceiling camera utilizes PBVS while the PTZ camera utilizes IBVS. Since the PTZ camera is capable of panning and tilting; therefore, the field of view constraint is decreased. Furthermore, the mobile robot is immune to image occlusion because of the hand-in-eye

camera on the ceiling. Alternatively, the authors in [126] developed an algorithm for systems with unknown camera intrinsic parameters. This is beneficial because it allows more flexibility when adopting the algorithm with different cameras. The same authors improved the work and published [127] where they describe a two-step controller that first rotates the robot to the correct orientation while estimating the intrinsic parameters in parallel. The second step drives the mobile robot to the desired position without changing the orientation. While this algorithm is functional without previous knowledge of the intrinsic parameters, [128] proposes an algorithm that functions without camera-to-robot parameters. This method also works as a two-step controller and features an adaptive control law. Most recently, the same authors published another paper that enables simultaneous depth identification [129]. This method works by using a concurrent learning strategy and an adaptive updating law to estimate the unknown depth information.

Recent publications in path following tasks are primarily focused with visual path following, where key images between the initial and desired pose are saved in the memory for the robot to follow. This is demonstrated in [130] where an algorithm that considers both motion and visibility constraints is proposed to achieve optimal path planning. Another advantage of visual path following is that the visual features from the target object is not necessarily required to remain within the field of view during navigation until the end [131]. To improve trajectory tracking, the authors in [132] propose a model predictive controller while an algorithm that represents rotation tracking error with quaternion formulation is presented in [133]. For tracking tasks, unified models are proposed in [134] and [135]. These unified controller's primary responsibilities are to track a desired trajectory and regulate the robot's position and orientation.

2.4.4 Implementation with MWMS Platforms

In this section, a survey of the literature published within the past decade regarding the control and autonomous navigation of MWMS platforms is presented. Starting with low-level kinematics and dynamics control systems for multi-wheeled vehicles, papers such as [136] [137] sets the basis for modeling of six and eight-wheeled platforms, respectively. A kinematics control law that considers wheel yaw, roll and suspension pitch for a 4WD4WS vehicle is proposed in [138]. The modeling and control of steering for a hydraulic-based system is examined in [139] where dynamic nonlinearities are compensated. Motion control with in-wheel motors are described for a six-wheel drive and six wheel steer (6WD6WS) vehicle in [140] where vehicle dynamics performance are improved implementing independent wheel torque and steering control; the results from this work are validated based on simulation. Vehicle stability and maneuverability is discussed in [141] where both an upper and lower controller work together to determine steering angles based on longitudinal forces, yaw moment and tire force information. To ensure smooth following of a given path, a bounded velocity motion controller with non-linear control techniques is described in [142]. Beyond control system development, the controllability of a similar vehicle for high-speed navigation in rough terrains is studied in [143]. All the mentioned work in control systems up to this point along with the others that are available generally focuses on four to six-wheeled vehicles with awfully limited literature for eight-wheeled vehicles.

In terms of high-level navigation algorithms of all-wheel drive and all wheel steer (AWDAWS) vehicles, recent publications have centered around either path planning or path following. For path planning, the authors in [144] describe an algorithm that features

two parts which are intended for steering and velocity planning. Optimal paths for an AWS vehicle are generated in [145] with motion splines that account for rough terrains and dynamics. Path planning using A* and the DWA algorithm is implemented in [146] for a 4WD4WS robot. This work is improved in [147] where pose estimation with RTK GPS and wheel encoders through an extended Kalman filter is applied. Most recently, a path planning technique that utilizes 7-order Bezier curves is developed to also provide velocity and acceleration profiles for a 4WD4WS vehicle in [148]. In this work, the vehicle is represented as a rigid body with previously determined characteristics such as mass and inertia. Conversely, recent path following algorithms include a basic approach that considers kinematic geometry are presented in [149]. A more advanced technique using sliding mode controllers for trajectory tracking is presented in [150] with the advantage that it can be applied to both front and all-wheel steering vehicles. Mamdani fuzzy logic controllers are implemented in [151] to follow waypoints that are generated based on the curvature derived point selection algorithm. Further development of this approach can be found in [152].

For eight-wheel drive and eight wheel steer (8WD8WS) vehicles specifically, research focused on dynamics control and path planning has been published over the last few years by members of the Crash Simulation and Vehicle Dynamics Lab at the University of Ontario Institute of Technology. The work in control systems started most notably with torque distribution in [153] for an 8WD8WS vehicle. This work was later improved by [154] with a feedforward zero side slip controller that is implemented to generate the rear axle steering angles. An optimal path planning algorithm based on the artificial potential field is proposed in [155] to drive the vehicle to a goal destination. Later, a robust heading

angle controller using h-infinity is introduced to overcome system disturbances such as noise [156]. All mentioned works are tested in simulation with promising results; however, physical experiments are required for further validation.

2.5 Mobile Robot Docking Methods

In this section, a review of recent work regarding mobile robot docking is presented. This is a type of maneuverer that requires precise pose correction of the mobile robot to arrive at a specified pose in a manner that is accurate enough for docking (often requires centimeter accuracy). The first work in autonomous docking dates to the beginning of mobile robot development in 1963 where robots used light to find the charging station [157]. Today, the concept of locating the charging dock via onboard sensors still holds with exponential improvements in both perception and computation performance. Currently used sensors for autonomous docking include Radio Frequency Identification (RFID), laser range finders, infrared and vision sensors. The primary challenge of such a task is to detect the charging station within the workspace before moving towards it in a precise fashion that enables charging. The following covers related work for each of the mentioned sensors.

Starting with RFID, the authors in [158] proposed a novel algorithm that utilizes a reader with dual directional antennas to estimate the arrival direction of docking station transponder signal. Through this, the robot calculates the direction it should travel for docking. To improve upon RFID technologies, laser range finders are used for charging station localization. A recent algorithm proposed by [159] uses a 2D laser rangefinder that is placed in the mobile robot workspace instead of onboard due to limited space. The laser sensor scans the workspace and detects the current position of the robot and the charging station to conduct closed-loop position control for docking. Alternative methods are

proposed in [160] and [161] where a laser rangefinder is placed onboard of the mobile robot to search for the charging station landmark.

A more popular solution when compared to RFID and laser rangefinders is the infrared sensor which is commercially available for mobile robot autonomous docking today. In [162], four infrared (IR) LED emitters and one IR receiver are used to estimate the angle and distance between the robot and the docking station by the angular intensity and the sensitivity distribution of the emitter and receiver, respectively. This work is improved by [163] which later saw physical integration in [164] where the authors proposed a system with IR sensors placed on either side of the docking station. Based on the readings from these sensors, the robot determines the relative positions to dock accordingly when the emitter signals overlap. An extension of this method is commercially available in autonomous vacuum cleaners such as the Roomba. Another implementation of IR sensor for automatic docking of home surveillance robot is described in [165]. Although the infrared method is popular, there is an issue with preciseness in long distances which is due to the assumption that the angle between emitter and receiver signals are close to zero. This assumption in addition to the sensor strength impacts the performance significantly; therefore, the authors of [166] proposed an Extended Kalman filters and Particle Filters method to fuse together readings from IR and encoder sensors to provide a more accurate estimation of distance and orientation.

Finally, camera sensors for autonomous docking exhibit the highest potential due to its ability to mimic human perception. Notable work in this area began with simulation work by the authors of [167] who proposed a fuzzy controller for the heading angle of a mobile robot. A PTZ camera is used in parallel with laser beacons to locate the charging

station in [168]. In this implementation, the PTZ camera scans the robot's workspace for an orange sticker which brings the robot closer to the docking station. The beacons are then detected by the laser rangefinder to determine the difference in orientation for correction. Another approach using only a single webcam is proposed in [169] where an Otsu algorithm recognizes landmarks and calculate the distance and orientation between the mobile robot and docking station afterward. The two previously mentioned approaches utilize simple visual tags such as orange stickers and black boxes, whereas the authors in [170] adopt the use of Quick Response (QR) codes. In this implementation, an IR sensor is applied to first detect which zone the robot is currently in. These zones are labeled as "very close", "close" and "far" and the docking strategy is different depending on each zone. Position and orientation of the mobile robot are estimated through the relative size of the lateral edges of the QR code as perceived by the camera. Another visual pattern like QR codes known as the AprilTag is used in [171] where a mobile robot is able to autonomously dock in a warehouse environment based on two steps. The first step utilizes ORB-SLAM for self-localization to move into the vicinity of the AprilTag at which point position control takes place to dock accordingly. Beyond home and warehouse settings, some robots are tasked with operations in harsh environments which makes it infeasible for human intervention after deployment. In [172], a path planning algorithm is proposed for a robot working in a vineyard with steep slopes to autonomously dock. In this work, AprilTags are also used where its relative position and orientation are used to generate a cubic Bezier curve for the mobile robot to follow. Other contributions from this work include a performance benchmark of several visual tags. Additional path planning work is presented in [173] where Dubins curves and artificial potential fields are combined and

integrated with physical systems. A recent approach that does not need artificial visual tags is proposed in [174] where the robot recognizes power outlets with a single camera to estimate the three-dimensional position.

2.6 Summary

Based on the literature included in this chapter, it is clear research gaps exist within each of the topics reviewed. Beginning with the designs of LAVs which have been heavily focused on personnel protection and engine improvements for higher torque output in the past four decades. The focus on vehicle performance improvements has neglected research and development efforts in autonomous navigation, powertrain electrification and alternative steering systems for the LAV. To explore further on the idea of a MWMS design for the LAV, a review is presented on MWMS robotic setups currently available. From this, it was learned that physical prototypes developed are often either only MW or MS, and rarely the combination of both. In addition, these platforms regularly lack car like feature such as steering and suspension with an even more dominant case for eight-wheeled vehicles. Furthermore, docking approaches that enable precise pose correction for MWMS platforms to take advantage of its mechanical attributes are also not explored. Because of MWMS platform's poor availability in both the commercial space and published literature, physical implementations and development of navigation methodologies are extremely limited with only a few focused on simulation-based studies.

As a result of this, further efforts should be invested in multi-steered systems and autonomous features for the Light Armored Combat Vehicle family. The design and development of a SECV prototype will generate meaningful contribution for both the military vehicle and mobile robotics field. Beyond the mechanical and electrical design,

research efforts towards a close quarter pose correction algorithm that capitalizes on the design of the proposed vehicle will also yield beneficial results. A summary of the literature review presented for all relevant topics is shown in the table below with major research gaps identified.

Table 2-1 Literature Review Summary

Literature Review	Research Gaps
2.2 Light Armored Vehicles	• All improvements since the 80's have been focused primarily on engine output and passenger protection • Lack of research and development in exploring multi-steered systems and autonomous features
2.3 Novel Scaled MWMS Robotic Platforms	• Focused on four-wheel variations with a lack of eight-wheeled designs that mimic a typical light armored combat vehicle • Lack of car-like features (steering, suspension) • Novel prototypes are either MW or MS and rarely the combination of both which are usually simulation based rather than physical implementation
2.4 Autonomous Navigation	• Lack of development and implementation with MWMS platforms
2.5 Mobile Robot Docking Methods	• Lack of applications that utilize alternative steering modes and vision-based approaches for better maneuverability

Chapter 3. The Scaled Electric Combat Vehicle

3.1 Introduction

In this chapter, the mechanical design as well as the electronics and software architecture of the proposed SECV prototype is presented. Starting with the mechanical system of the SECV which is classified into four different subsystems; namely, chassis, suspension, driving and steering. Each of these subsystems is illustrated in this section to showcase its design details. Furthermore, an electronics hardware architecture including sensors, actuators and controllers is presented. This discussion will lead into the design of a modular software architecture within ROS that prepares the SECV for autonomous navigation development as presented in the next two chapters. The following figure shows the physical model of the SECV prototype that was designed and built over a period of two years.

Figure 3-1 Physical SECV Prototype

3.2 Mechanical Design

3.2.1 Chassis

The design of the SECV chassis resembles the letter *T* as the shape conforms to the suspension system while fitting the steering components in the desired location as seen in Figure 3-2. In addition, the chassis is also responsible to hold all required electronic components to enable vehicle actuation while maintaining its structural integrity to eliminate undesirable deflections. To accomplish this, the chassis is broken down into seven aluminum pieces which include: two identical left and right-side panels, two identical front and back pieces, a base which holds the motors and gearboxes, the middle which holds the steering actuators, and the top layer which hold motor controllers, batteries and the central computing unit. These seven pieces are water jet cut and bent with the side panels designed to be symmetrical to make the parts interchangeable; thereby improving cost and ease of assembly. Rivets are chosen as the method of connecting panels because of their ease of removal when necessary while maintaining adequate strength. The internal layers of the chassis are designed to be the same dimensions but are unique due to different mounting points required for each shelf. An additional support bolt is added to the steering layer as any movement to this shelf could cause damage to the steering components.

3.2.2 Suspension

The suspension design of the SECV is inspired by a double wishbone setup where the steering knuckle is mounted on two “control arms”; an upper arm and a lower arm. This system is made of mostly standard parts except for the lower control arm which is designed to be a specific length that provides the desired suspension travel and a width that fits the standard components of the wheel assembly. The lower control arms are water jet cut to

reduce the mass of the parts while retaining rigidity. A channel is milled along the top to accommodate the driveshaft and the holes for mounting the shocks are mirrored across the control arms to create an interchangeable part.

The advantage of using a double wishbone system is that it offers a large wheel travel, which is necessary in all-terrain vehicles. By mounting the upper and lower control arms almost parallel to each other, a four-bar mechanism is created. Furthermore, by keeping the lengths of the control arms the same, the four-bar linkage system keeps the remaining two sides parallel to each other while the angle of the arms changes. This means that the tires will remain at their set camber, which increases handling. Although the MacPherson strut is a simpler and more space efficient design, its travel distance is restricted to a fraction of the shock absorber. With the double wishbone design, the travel distance is a function related to the lengths of all its components, allowing for finer tuning. By keeping the upper and lower arm approximately the same length, the tire can move vertically without any camber change. A front view of the SECV prototype is displayed in Figure 3-2 where the T-shape chassis, internal layers, shock absorbers and control arms are shown.

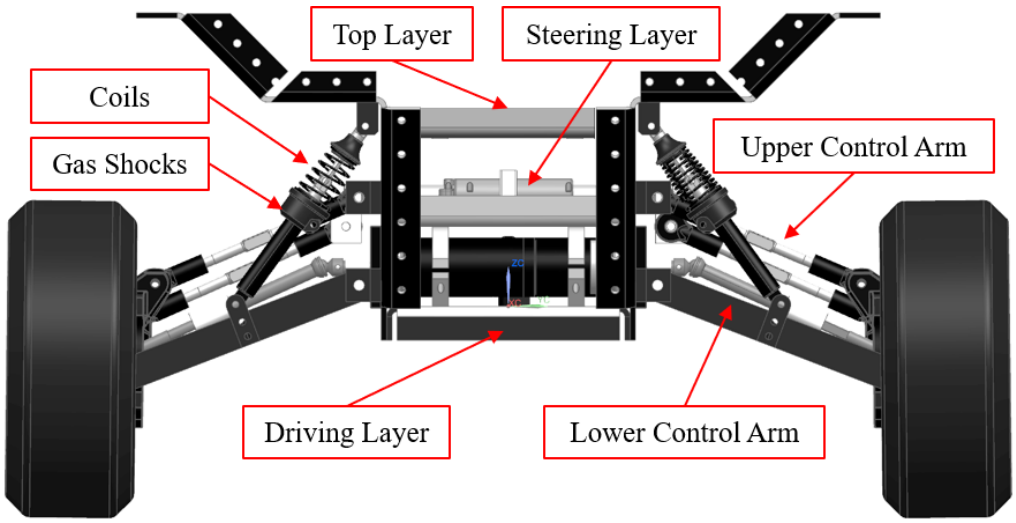

Figure 3-2 Front Section View of SECV Prototype

3.2.3 Driving Layer

Starting from the bottom with the Driving Layer, there are two DC motors per axle for a total of eight independently driven wheels. Each DC motor is attached to a 33:1 gearbox to reduce the total rotational speed while increasing the output torque. More specifically, the maximum rotational output speed is 217 rpm while the nominal output torque is 2.41 Nm. With these specifications, the vehicle is able to achieve a top speed of approximately 1.80 m/s. Due to the limitations imposed by the dimension of the chassis, it is not possible to place two motors back to back for a direct drive system; therefore, a pulley system with a 1:1 ratio per wheel is implemented instead. This setup enables the motors to be mounted in parallel with the axle axis as illustrated by Figure 3-3. In this figure, the top and bottom DC motors are driving the left and right wheels, respectively. Placed in between the side wall of the chassis and the output pulley is an encoder that is mounted on the output shaft.

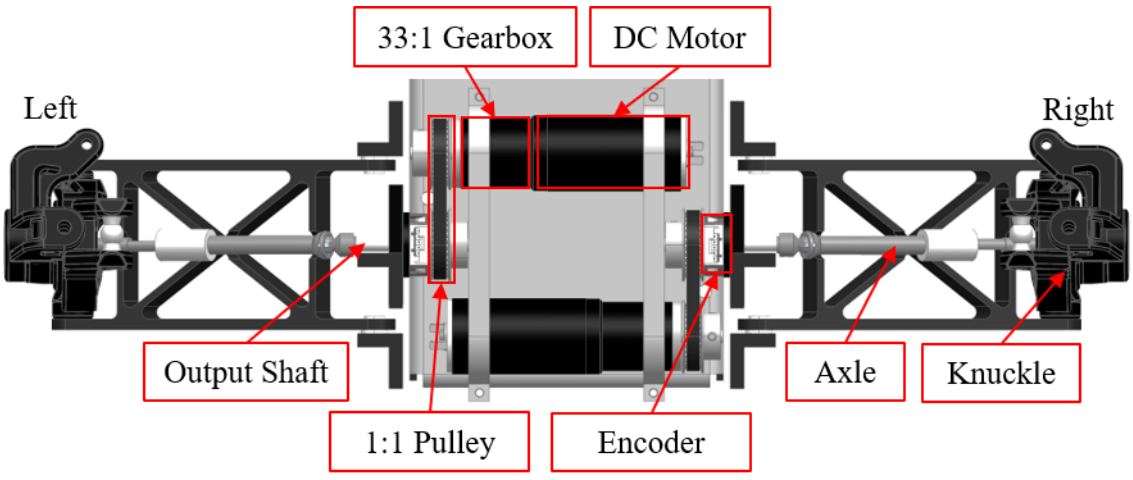

Figure 3-3 Top View of Driving Layer

The 1:1 pulley box is designed to attach to the end of the existing motor/gearbox combo. The use of a belt and pulley system on bearing supports have lower friction than the average gearbox and allow for looser tolerances during manufacturing. This solution allows an easy method of attaching the encoders. In addition, this 1:1 offset provides a high degree of design flexibility for future ratio changes. The current design utilizes a 1:1 ratio between the pulleys to maintain 180º of tooth engagement, which helps mitigate the risk of tooth skipping. The case element is a 2.5D based design that allows for simple manufacturing with either a 3-axis CNC router or 3D printer. Flush mount bearings in the support walls allow for smooth running operation and help to reduce frictional losses.

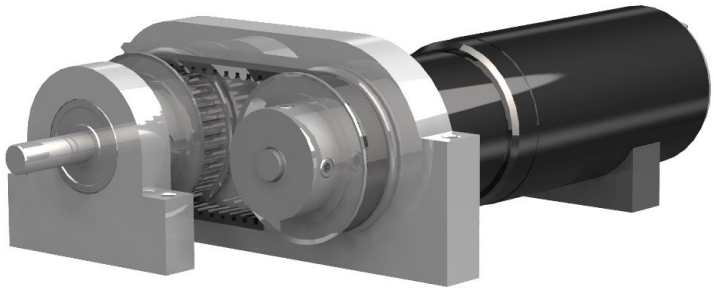

Figure 3-4 1:1 Pulley Box Design

3.2.4 Steering Layer

The Steering Layer sits approximately 6.35 cm on top of the Driving Layer where linear actuators are attached to each knuckle through a tie rod. The benefits of the linear servo system are that they offer structural rigidity and robustness in the steering system which ultimately reduces compliance. Each actuator has a total stroke of 50 mm with 25 mm being the neutral position. Steering of each wheel is accomplished through extending and retracting the actuators. In Figure 3-5, an extended left actuator with a retracted right actuator would steer both wheels to the right with respect to the steering axis in each

knuckle. The tightest turn would happen at full extension and retraction, respectively. Built into each actuator are potentiometers that provide stroke position feedback. The relationship between this feedback and the achieved steering angle is derived later in this thesis based on experimental results.

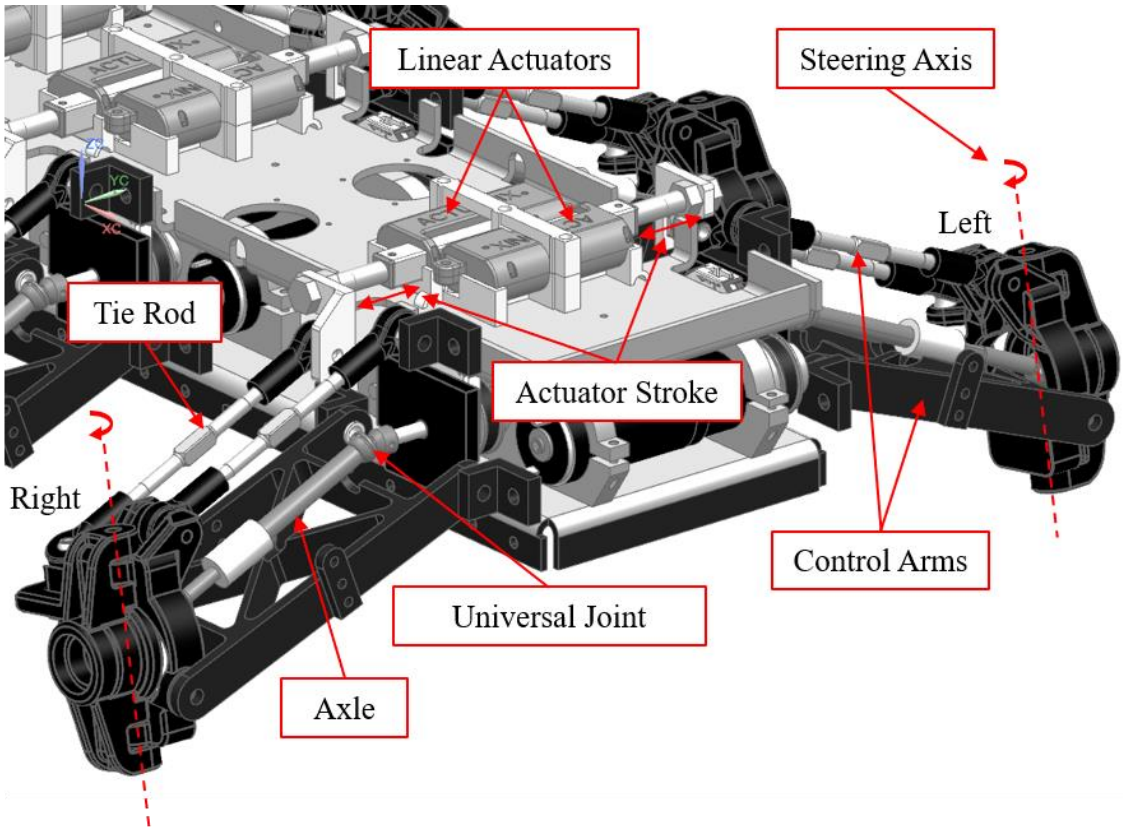

Figure 3-5 Isometric View of the Steering Layer

Since the proposed SECV features independently steerable wheels, it is capable of numerous steering configurations. However, this thesis focuses primarily on three modes which are front wheel steer (FWS), 4th axle steer (4AS) and all-wheel steer (AWS) as illustrated in Figure 3-6 for clarity.

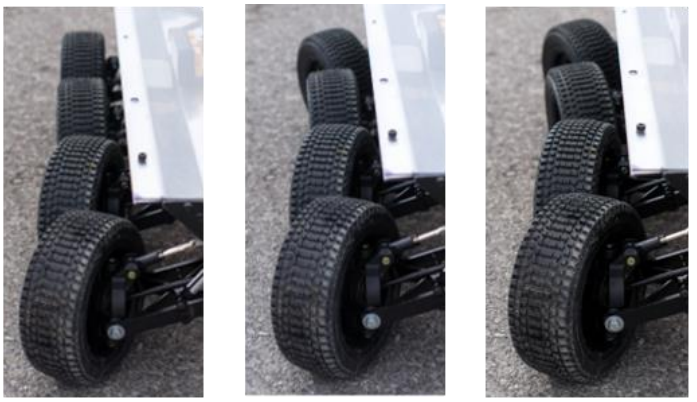

Figure 3-6 Multiple Steering Configurations

3.3 Electronic Hardware Architecture

The hardware architecture of the SECV prototype is described in this section with relationships between all electrical components and specifications listed. Starting with the full system hardware architecture as shown in Figure 3-7, the central processing unit of the SECV is an onboard laptop computer loaded with Ubuntu 14.04. This computer is interfaced with various controllers and sensors via Universal Serial Bus (USB). Beginning with the controllers, there are two types that are embedded within the vehicle. The first type is denoted as the Steering Controller which controls up to four linear actuators per unit. Since one of the novelties of the vehicle is its 8WS setup; therefore, two steering control units are necessary to control one linear actuator per wheel. Besides receiving and providing 12V from the onboard power supply, the steering controllers also receive feedback from a built-in potentiometer that enables closed loop stroke/steering control. On the other hand, the second type of controller is denoted as the Motor Controller where a single unit is implemented per axle to control two DC motors each. For simplicity, the driving controllers are set up in a way where only the master is controlled by the laptop via

USB and the remaining three are controlled via CAN as slave nodes. Attached to the end of every DC motors are encoders that provide feedback for closed loop speed control. As shown in the figure below, each driving controller receives 15V from the onboard power supply and provide them to the DC motors where it is then stepped down to 5V for the encoders. In terms of sensor instrumentation, a 9 DOF IMU, 360-degree laser scanner and a single lens camera are integrated along with a Bluetooth receiver for close range teleoperation. All components are realized within the ROS environment with their key specifications listed in Table 3-1.

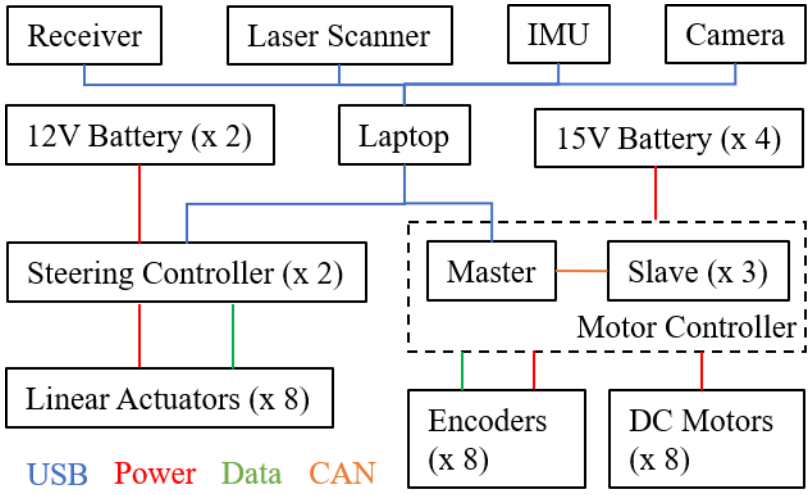

Figure 3-7 Full System Hardware Architecture

Table 3-1 Electronics Component Specifications

Component	Specification
Laptop	Intel Core i5-5300, 8gb RAM
Battery	14.8V LiPo 5000 mAh and 12V NiMh 2800 mAh
IMU	UM7-LT Orientation Sensor
Laser Scanner	RPLIDAR A2M8 360 Laser Scanner

Receiver	Logitech Gamepad F710
Camera	ELP USB Camera 2.1 mm Lens 1080p
Steering Controller	Arduino Uno + DFROBOT Quad Motor Driver
Driving Controller	RoboteQ SDC2130 Brushed Motor Controller
Linear Actuators	Actuonix P16 Micro Linear Actuator 50mm, 22:1
DC Motors	Maxon DC Brushed motors with 33:1 gearbox
Encoders	AMT10 incremental encoder

3.3.1 Sensor Bridge

Installed on top of the chassis is an aluminum bridge that provides mounting locations for the different sensors mentioned. These sensors include the laser scanner, IMU, camera and the laptop as seen in the figure below. In addition, the dimension of the bridge is designed to proportionally mimic the outer profile of a life-size combat vehicle.

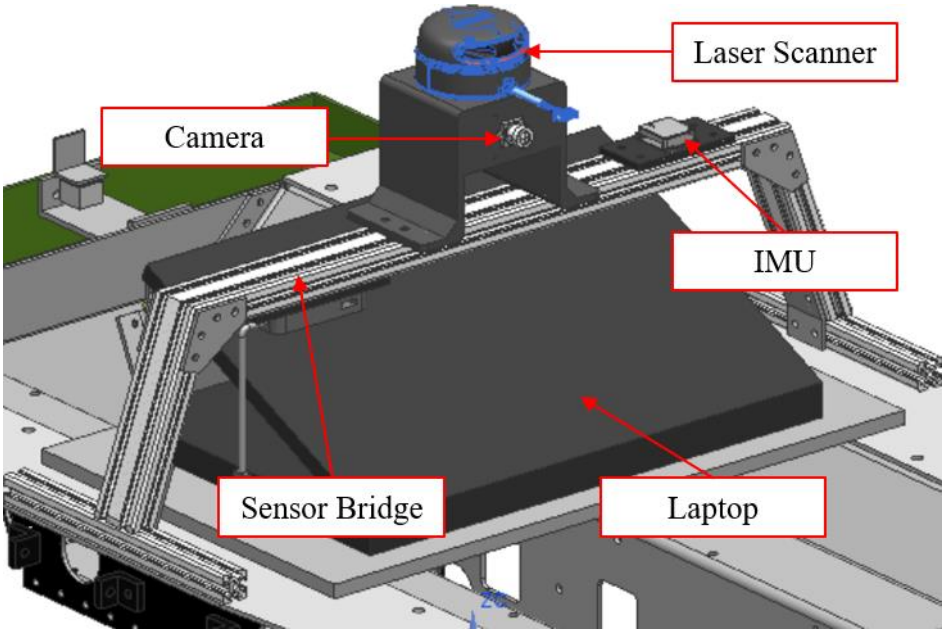

Figure 3-8 Sensor Bridge

3.4 ROS Nodes and Topics Structure

After the mechanical and electrical systems are completed, a modular software development environment is created within ROS by writing low-level software to generate nodes for publishing and subscribing as discussed in Section 1.3.2.1. Before any autonomous software is developed, teleoperation for the SECV is first created to allow for user control. To start, a */teleop* node is written to publish to a topic called *cmd_vel* which includes both a linear and an angular velocity message. This topic is critical in controlling the motion of the SECV as any additional software tasked with motion control are to publish to this topic. For example, other navigation based software can be represented by the placeholder node called, **other_nav_nodes** in Figure 3-9. As a safety measure, the */teleop* node is always prioritized over any other nodes publishing to the cmd_vel topic as user commands should override control of the SECV during emergency situations.

Moving forward, the linear velocity message within the *cmd_vel* topic is subscribed by the */motor_driver* node which subsequently dictates the motor output. A */convert_to_ackermann* node is created to subscribe to both the linear and angular velocity messages to calculate the appropriate steering angles based on Ackermann steering geometry from Section 1.3.1 . The steering angles are published and subscribed by the */steering_controller* node which then controls the linear actuators. Besides motion control, nodes and topics are also written for the onboard sensors. For example, the */usb_camera* node publishes to a topic called *rgb_image* which holds a matrix message representing the RGB values of each pixel to form an image. The laser, IMU and encoder nodes are responsible for publishing topics such as *laser_scan*, *yaw* and *lin_tran*, respectively. These topics are available for subscription by any future nodes that may include navigation within

ROS. The described architecture is illustrated in Figure 3-9 where squares and ovals symbolize nodes and topics, respectively. It is important to note that this figure shows a simplified network for clarity purposes as there are more nodes that exist within the vehicle which handles other aspects such as feedback control, power management and other background processes. With this architecture, additional software features covered in later sections such as mapping, localization and path planning are implemented by subscribing to the sensor nodes and publishing to the motion control nodes.

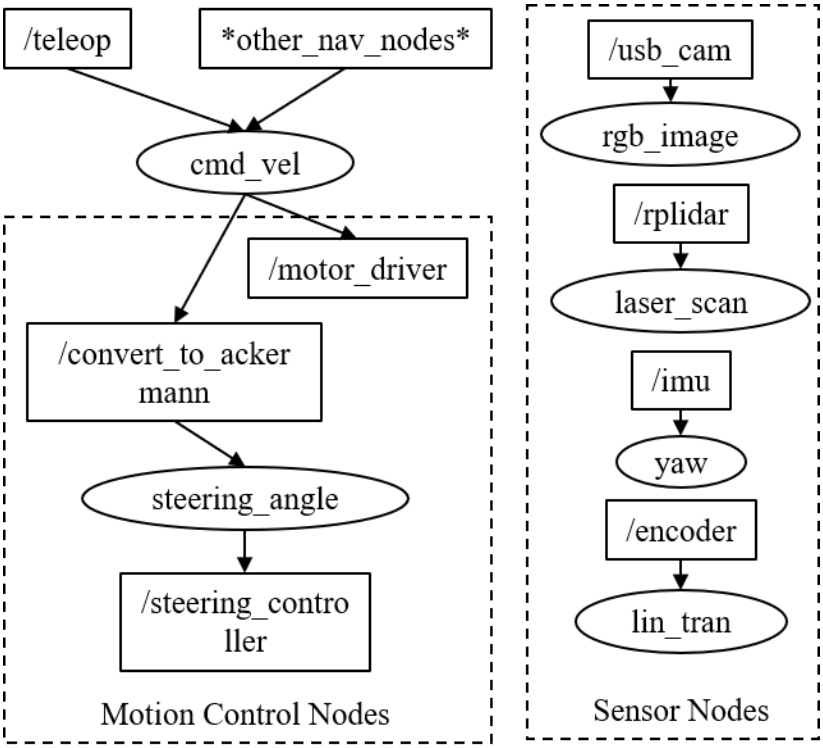

Figure 3-9 Fundamental Nodes and Topics for the SECV

3.5 Summary

In summary, this chapter proposes a physical SECV prototype that improves upon the current light armored combat vehicle design as well as mobile robotic platforms by creating a fully operational 8WD8WS robotic platform that is instrumented with sensors and computing hardware. To describe the prototype, detailed Computer Aided Design (CAD) models are presented for all mechanical systems of the vehicle. Next, the necessary electronic components are sourced and implemented to enable vehicle actuation. From there, a modular software architecture composed of low-level software nodes and topics is developed and integrated within ROS. As a result of the work presented in this chapter, the SECV is capable of teleoperation and able to accept additional software to expand its capabilities. With the mentioned design, the following tabulates all basic dimension and performance specifications of the SECV. The next two chapters will cover the algorithms developed and implemented to enable autonomous navigation.

Table 3-2 SECV Specifications

Category	Measurement
Chassis Dimension (mm)	869.65 x 438.90 x 201.55
Total Wheel Base (mm)	604.44
Track Width (mm)	460
Tire Radius (mm)	88.9
Suspension Rating	30 lb shocks + 10 lb coils
Total Mass (kg)	40
Max Speed	1.80 m/s
Max Steering Angle	30 degrees

Chapter 4. Autonomous Navigation Methodology

4.1 Introduction

In this section, the algorithms necessary to achieve autonomous point to point navigation for the SECV is discussed and implemented. The procedures to enable this feature is as mentioned in Section 2.4 where the first step is to map the workspace followed by localization and path planning with low-level vehicle control happening at the end. As a result, this chapter will progress in chronological order of these steps. More specifically, the discussion on the mapping algorithm will begin with an overview of particle filters and how it is used for mapping. This will be followed by the localization algorithm developed for the SECV which utilizes onboard odometry sensors. Correspondingly, the map and localization data are used by both the global and local path planners as introduced from Section 2.4.3.1 . Lastly, a model is derived to describe the kinematics and steering geometry of the vehicle with low-level vehicle control presented.

4.2 Mapping

4.2.1 Particle Filter

Recursive Bayesian estimation, or simply Bayes filter, is used in robotics to recursively estimate a robot's state and environment. More specifically, this technique estimates unknown probability distributions using real-time sensor measurements. All Bayes filters are based on the Markov assumption where the next state depends only on the current state and not the past. A discrete representation of Bayes filter known as the Particle Filter which is used to estimate the states of dynamic systems when only partial and noisy

observations are made. This is done through estimating a posterior distribution which is a prediction of possible un-observed values while considering the observed values.

More specifically, an arbitrary probability distribution based on sensor data represents an estimation of a mobile robot's pose. From there, samples made of a state hypothesis, x, and an importance weight, w, as shown in the equation below is used to represent the arbitrary distribution.

$$X = \{(x^{[j]}, w^{[j]})\}_{j=1,\ldots,j} \tag{4-1}$$

To obtain the samples, a method known as "Importance Sampling Principle" leverages sampling methods for known distributions to sample arbitrary distributions [175]. For example, to sample a target arbitrary distribution (red) in the figure below, which in this case represents the pose estimation of the mobile robot, it is possible to first sample the proposal distribution (Gaussian) and then account for the difference between the proposal and target. This account will come in the form of a re-weighting procedure that will enable the samples to better fit the target distribution. The two distributions and the samples are shown in the figure below.

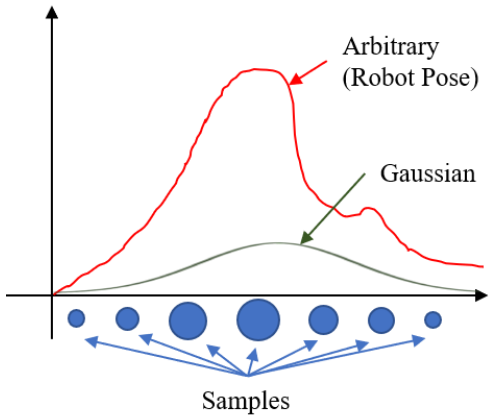

Figure 4-1 Robot Pose Distribution and Samples

The particle filter is described in a more formal fashion as shown in Figure 4-2. In line 1, the predicted set is initiated to the same as the corrected set which is all empty in the beginning. In line 2, the algorithm iterates over all samples. In line 3, new samples are obtained from the Gaussian distribution. Line 4 and 5 describe the re-weighting of the samples which uses the Importance Sampling Principle to calculate the difference between Gaussian and arbitrary distribution. The resultant is added to the sample set. Line 7-10 describes the resampling step to get rid of bad samples. This describes how a basic particle filter works while the next section expands on this to achieve mapping and localization.

$\mathbf{Particle\_filter}(\mathcal{X}_{t-1}, u_t, z_t)$**:**

1: $\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
2: *for* $j = 1$ *to* J *do*
3: *sample* $x_t^{[j]} \sim \pi(x_t)$
4: $w_t^{[j]} = \frac{p(x_t^{[j]})}{\pi(x_t^{[j]})}$
5: $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[j]}, w_t^{[j]} \rangle$
6: *endfor*
7: *for* $j = 1$ *to* J *do*
8: *draw* $i \in 1, \ldots, J$ *with probability* $\propto w_t^{[i]}$
9: *add* $x_t^{[i]}$ *to* $\mathcal{X}_t$
10: *endfor*
11: *return* $\mathcal{X}_t$

Figure 4-2 Particle Filter Pseudocode [176]

4.2.2 Simultaneous Localization and Mapping

As introduced in Section 2.4.1 , SLAM is an algorithm that simultaneously localizes a mobile robot within a map that it is also generating. In ROS navigation, the default SLAM algorithm used is known as GMapping which implements the FastSLAM 2.0 algorithm [177]. FastSLAM 2.0 builds on top of the concepts established by FastSLAM 1.0 (Rao

Blackwellization Filter SLAM [178]) by featuring two main improvements which are first by adding an improved proposal distribution and second, including an adaptive resampling method. The improved proposal distribution considers the sensor measurements during sampling which enhances the accuracy since a precise sensor can be used to reduce the impact of noisy motion estimation. The adaptive resampling method also ensures a proper number of samples are available to avoid particle depletion.

Starting with the first improvement, the new proposal distribution considers both the laser scan observation, $u_{1:t}$, as well as the robot odometry information, $z_{1:t}$, as shown in equation 4-2. From this, the two distributions are seen in Figure 4-3 where the pose distribution based on laser scan is peaked as it is accurate and able to provide good guesses of where the robot is. On the other hand, the distribution based on odometry sensors are often flatter because the data is noisy and inaccurate. By combining these two distributions, the accuracy of the overall estimation is much improved as laser scan matching and odometry are good with local and global localization, respectively. This behaviour is illustrated in Figure 4-3 for clarity.

$$X_t^{[k]} \sim p(x_t | x_{1:t-1}^{[k]}, u_{1:t}, z_{1:t}) \tag{4-2}$$

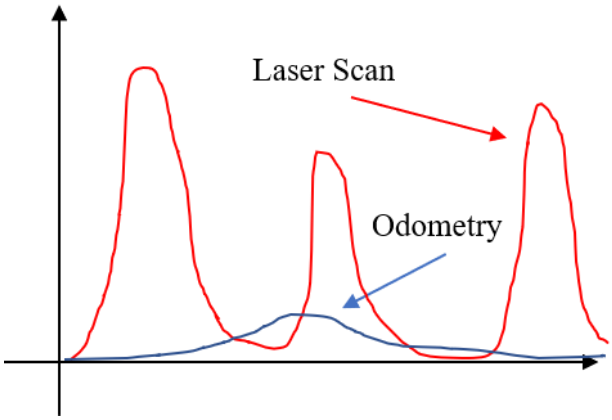

Figure 4-3 Distributions for Laser Scan Matching and Odometry

Moving on to the second improvement which is the implementation of a method to determine when to resample. This method utilizes n_{eff}, which is the inverse variance of the normalized particle weights. When this number is small, it means that resampling is necessary because most particles have a bad estimate. When this number is high, it becomes harder to tell which sample is better. Using the value, resampling can take place whenever n_{eff} drops below a given threshold. As a result of these two improvements, FastSLAM 2.0 achieves a better performance when compared to FastSLAM 1.0 due to the improved proposal distribution that considers scan matching. This ultimately led to a substantial reduction of the number of samples while the selective resampling method also reduces the risk of particle depletion.

4.3 Localization

To localize the SECV, an incremental localization algorithm that utilizes both wheel encoders and an IMU is implemented. In the proposed strategy, the wheel encoders and IMU are responsible for linear and angular displacement, respectively. The wheel encoders show acceptable performance in short range navigation; however, IMU drift issues are hard to ignore. From the experiment, it was determined that the yaw drift is extra prominent during long-distance navigation and when the vehicle is in an immobile state. For the former issue, a linear drift was deduced during physical trials; therefore, a compensator was integrated as a remedy. This compensator works by consistently subtracting the drift in a linear fashion. To alleviate the latter issue, a switching algorithm that stops updating the orientation of the vehicle when it is stationary and resumes when the vehicle becomes mobile is integrated. The results show promising capabilities for the application of this work.

4.4 Path Planning

After mapping and localization are complete, the next step is to autonomously navigate the SECV within the generated map by planning paths accordingly. In ROS, navigation is accomplished by two types of path planners; namely, global and local. The role of the global path planner is to find a path between the start and goal position while considering the static map that was previously acquired. The output of the global path planner is a series of waypoints that hold pose information. On the other hand, the local path planner's responsibility is to convert the pose waypoints to achievable velocities while considering sensor observations in real time to account for obstacles that were not present during the mapping phase. A cost map is updated throughout the navigation by adding and clearing obstacles. The following will provide further details in the usage of the cost map and the two types of planners that are implemented in the SECV.

4.4.1 Two-Dimensional Cost Map

In ROS, a two-dimensional cost map is a grid-based representation of a map where each cell holds a value between 0 and 255. On one end of the spectrum, 255 implies that there is no information available for the cell because it is out of the sensor's range. On the other end of the spectrum, a cell with 0 implies that it is free space and the robot can freely travel in these cells. The values 254 and 253 infer that an obstacle is detected in the cell and collision is expected. From 253, each obstacle is inflated by a user-specified radius to create a gradient bubble around the obstacle. In other words, this gradient is represented by cell values that decreases from 253 as the inflation expands to create an alert zone for the mobile robot as it navigates. In order for the global and local planners to work, there exists a global and local cost map. The global cost map is based on the static map that was

previously acquired in the mapping phase. Conversely, the local cost map represents a moving window of user specified size surrounding the mobile robot. This window is consistently updated with incoming sensor readings. Using these two cost maps, the respective planners can generate the appropriate commands for the SECV to arrive at the intended goal without collisions.

4.4.2 Global Path Planner: Dijkstra's Algorithm

The global path planner chosen for this work is the Dijkstra's Algorithm because of its open source availability and ease of implementation. This planner assumes a circular robot and it does not consider the robot's dynamics. Its sole purpose is to find the shortest path between the start and goal position without colliding with any obstacles that are observed during the mapping process. To understand how this algorithm works, one can imagine the aforementioned cost map broken down into a node-based representation. Each node begins with a temporary distance value which is zero and infinity for the starting and finishing node, respectively. Each node is categorized into one of two sections which are either visited or unvisited. At the beginning of the algorithm, only the initial node is labeled as part of the visited set. Connecting every node in this graph is a cost value associated with traveling between the nodes. In an iterative manner, the current node calculates the cost to visit all its neighbor nodes by adding the current cost with the traveling cost. If the value is less than that of the unvisited node, then the value of the unvisited node is updated, if not, then the value remains unchanged. During the first iteration, all values of the unvisited nodes will change since they are initialized to infinite to begin with. When all the neighboring nodes are visited, the neighbor with the smallest assigned value will be set as the new current node and subsequently added to the visited category. From there, the

algorithm repeats by calculating new values for all its neighbors. By the end of the algorithm, the values arriving at each node are stored in an array where the shortest series of costs between the initial and goal node is selected as the global plan. An example of a global plan based on the mentioned node graph is presented below. In this case, the green and red circles represent the initial and goal nodes. The blue represents the computed plan based on the cost of arriving at each node.

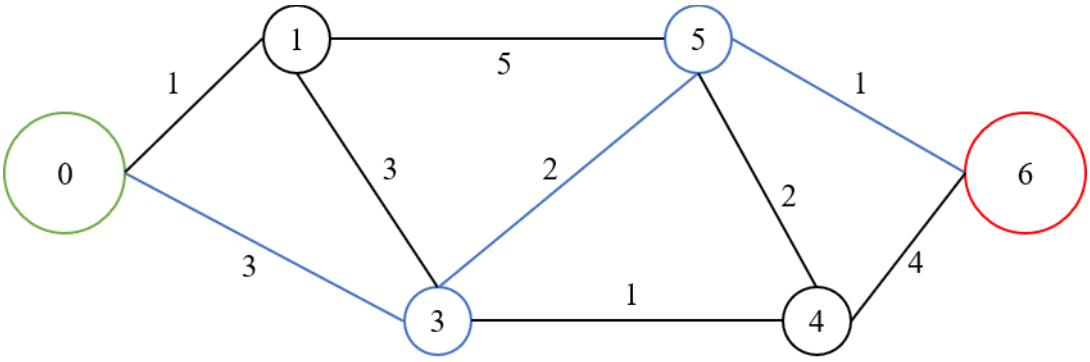

Figure 4-4 Dijkstra's Algorithm

4.4.3 Local Path Planner: Timed Elastic Band

To consider real-time sensor data and mobile robot dynamics, a local path planner known as the "Timed Elastic Band" is implemented. As mentioned in Section 2.4.3.1 , this algorithm builds on top of the "Elastic Band" planner which is defined by a series of robot position (x_i, y_i) and orientation, β_i, as seen below.

$$Q = \{X_i\}_{i=0\dots n}\ , where\ X_i = (x_i, y_i, \beta_i) \tag{4-3}$$

As the name suggests, the TEB algorithm extends beyond this by adding time intervals, ΔT_i, between two successive robot poses as shown below.

$$B := (Q, \tau), where\ \tau = \{\Delta T_i\}_{i=0\dots n-1} \tag{4-4}$$

The underlying goal is to find a path that optimizes the series of robot pose and time intervals based on a multi-objective optimization framework. The objective function is the weighted summation of the component, f_k, which considers topics such as the nonholonomic constraint, fastest path, as well as distance to waypoints and obstacles. This is illustrated with the equation below.

$$f(B) = \sum_{k} \gamma_k f_k(B) \tag{4-5}$$

The mentioned components are categorized into two types, which are either velocity and acceleration or trajectory based [79]. For the first category, the velocity and acceleration limits of the mobile robot are bounded by vehicle performance specification. For the second category, objective functions that consider the nonholonomic constraint as well as the fastest path are presented in equations 4-6 and 4-7, respectively, where $d_{i,i+1}$ denotes the direction vector between two consecutive way points.

$$f_k(x_i, x_{i+1}) = \left\| \left[\begin{pmatrix} cos\beta_i \\ sin\beta_i \\ 0 \end{pmatrix} + \begin{pmatrix} cos\beta_{i+1} \\ sin\beta_{i+1} \\ 0 \end{pmatrix} \right] \times d_{i,i+1} \right\|^2 \tag{4-6}$$

$$f_k = \left(\sum_{i=1}^{n} \Delta T_i \right)^2 \tag{4-7}$$

Next, distance to each waypoint on the generated global path as well as nearby obstacles are considered with equations 4-8 and 4-9. In these equations, the waypoints and obstacles attract and repel the elastic band, respectively. The minimal separation distance between TEB and waypoints or obstacles is symbolized by d_{min}, j which is either bounded from above by r_{pmax} or below by r_{omin} by waypoints and obstacles, respectively.

$$f_{path} = e\left(d_{min,j}, r_{pmax}, \in, S, n\right) \tag{4-8}$$

$$f_{obstacle} = e\left(-d_{min,j}, -r_{omin}, \in, S, n\right) \tag{4-9}$$

Equations 4-6 - 4-9 are combined with equation 4-5 to form the complete multi-objective optimization framework.

4.5 Low-Level Control

The last step of the navigation procedure is to convert the output of the path planners into meaningful commands for low-level controllers. To do so, the following section will first cover the kinematics model of the SECV as well as the applied Ackermann steering geometry for a MS system. Lastly, a discussion on the speed and steering controllers are presented with control block diagrams shown.

4.5.1 Kinematics Model

In this section, the kinematics model of the SECV is derived. As illustrated by Figure 4-5, a simplified bicycle model along the longitudinal axis is illustrated between the physical wheels of the vehicle. More specifically, $(\delta_{Li}, \delta_{Ri})$ denotes the steering angles of the wheels of the vehicle while (δ_1, δ_4) denotes the steering angle average of the first and fourth axle. Equation 4-10 represents the nonlinear continuous time relationships of the different velocities of the system where $(\dot{x}, \dot{y})$ are the linear velocities along the respective axis.

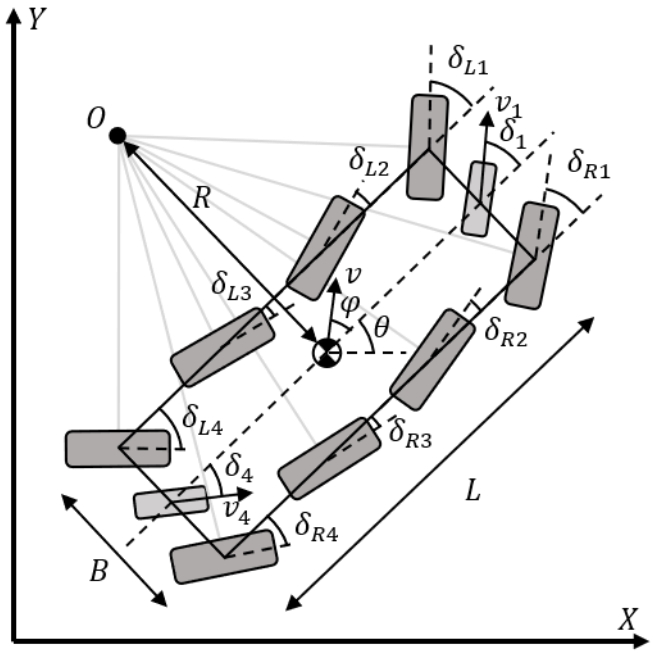

Figure 4-5 SECV Kinematics Model

These velocities are calculated by the product of the total longitudinal velocity, v, and the vehicle orientation with respect to the x-axis, θ, along with the angle between the direction of the velocity with respect to the longitudinal axis of the vehicle, φ. The rate of change of the heading angle is denoted by $\dot{\theta}$, which is calculated by considering the length to the center of gravity (CG) of the vehicle from the front and rear axles.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = v * \begin{bmatrix} \cos(\theta + \varphi) \\ \sin(\theta + \varphi) \\ \frac{\cos(\varphi)}{L}(tan\delta_1 - tan\delta_4) \end{bmatrix} \tag{4-10}$$

To find the velocity at the CG, the velocity average of the first and last axle of the vehicle, (v_1, v_4) , is calculated as shown below:

$$v = \frac{v_1 \cos(\delta_1) + v_4 \cos(\delta_4)}{2 \cos(\varphi)} \tag{4-11}$$

Moving forward, the angle between v and the longitudinal axis of the vehicle is calculated with the following equation:

$$\varphi = \tan^{-1}\left(\frac{\frac{L}{2}\tan(\delta_1) + \frac{L}{2}\tan(\delta_4)}{L}\right) \tag{4-12}$$

In order to simplify equations 4-10 to 4-12, the path curvature of the vehicle, σ, along with some assumptions are considered. Starting with the curvature equation which is calculated as the inverse of the turning radius, R^{-1}, which is better defined as the quotient of angular and linear velocities. This relationship is illustrated below:

$$\sigma = R^{-1} = \frac{\dot{\theta} + \dot{\varphi}}{v} \tag{4-13}$$

Besides the curvature equation, the necessary assumptions made to simplify equation 4-10 include setting the CG location to the middle of the vehicle body and assuming the velocities, (v_1, v_4), are equal in magnitude but opposite in direction. With these assumptions applied, equations 4-10 and the derivative of 4-12 is substituted in to equation 4-13 to form the kinematics model below:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = v * \begin{bmatrix}\cos(\theta)\\ \sin(\theta)\\ \sigma\end{bmatrix}, where\ \sigma = \frac{2\tan(\delta)}{l} \tag{4-14}$$

As shown in Figure 4-5, the SECV is designed to follow the Ackermann steering geometry to reduce tire degradation. Since the vehicle is in all-wheel steer mode, the instantaneous turning center is denoted by, O, which intersects the CG of the vehicle. From there, eight steering angles are calculated as shown below where L and B denotes the wheel base and the track width of the vehicle, respectively.

$$\delta_{L1} = \tan^{-1}\left(\frac{L/2}{R - B/2}\right), \delta_{R1} = \tan^{-1}\left(\frac{L/2}{R + B/2}\right) \quad (4\text{-}15)$$

$$\delta_{L2} = \tan^{-1}\left(\frac{L/4}{R - B/2}\right), \delta_{R2} = \tan^{-1}\left(\frac{L/4}{R + B/2}\right) \quad (4\text{-}16)$$

$$\delta_{L3} = \tan^{-1}\left(\frac{L/4}{R - B/2}\right), \delta_{R3} = \tan^{-1}\left(\frac{L/4}{R + B/2}\right) \quad (4\text{-}17)$$

$$\delta_{L4} = \tan^{-1}\left(\frac{L/2}{R - B/2}\right), \delta_{R4} = \tan^{-1}\left(\frac{L/2}{R + B/2}\right) \quad (4\text{-}18)$$

4.5.2 Driving Control

With the kinematics model established, a PID driving controller using incremental encoders mounted at the end of each DC motor for closed-loop control is employed. The goal is to ensure the error between the desired and actual vehicle velocity remains as minimal as possible during operation. In addition, the controller also calculates a velocity for the CG of the vehicle, $\boldsymbol{v}$, which is used by the software differential to generate inner and outer wheel speeds in the presence of non-zero yaw commands. The reason behind this is to improve steering maneuverability and decrease tire scrubbing imposed by different turning circle diameters between the left and right wheels. Equation 4-19 and 4-20 calculates the differential speed based on linear and angular velocity as well as the track width of the vehicle. Figure 4-6 illustrates the control block diagram with the software differential.

$$\omega_R = v - (\dot{\theta} * B/2) \quad (4\text{-}19)$$

$$\omega_L = v + \left(\dot{\theta} * B/2\right) \quad (4\text{-}20)$$

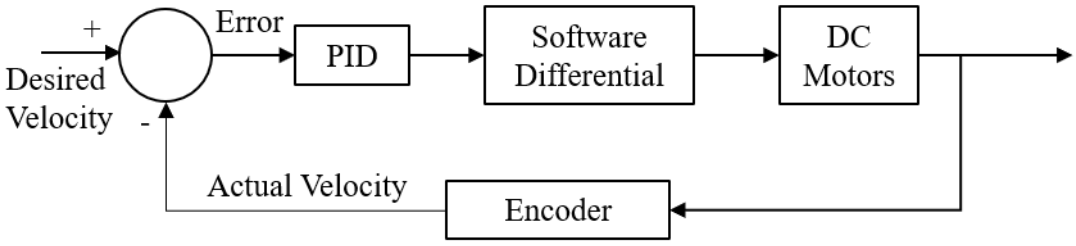

Figure 4-6 Closed Loop Speed Control with Software Differential

4.5.3 Steering Control

Next, desired steering angles are calculated based on Ackermann geometry from equation 4-15 - 4-18 which considers both the linear and angular velocity commands. With a desired steering angle, an actuator stroke position control algorithm is implemented to ensure satisfiable output. Since the vehicle features independent linear actuators for steering, built in potentiometers are used for stroke position estimation. The maximum stroke of each linear actuators and achievable steering angles are 50 mm and 30 degrees, respectively. Figure 4-7 illustrates the relationship between the steering angle and actuator stroke based on experimental data.

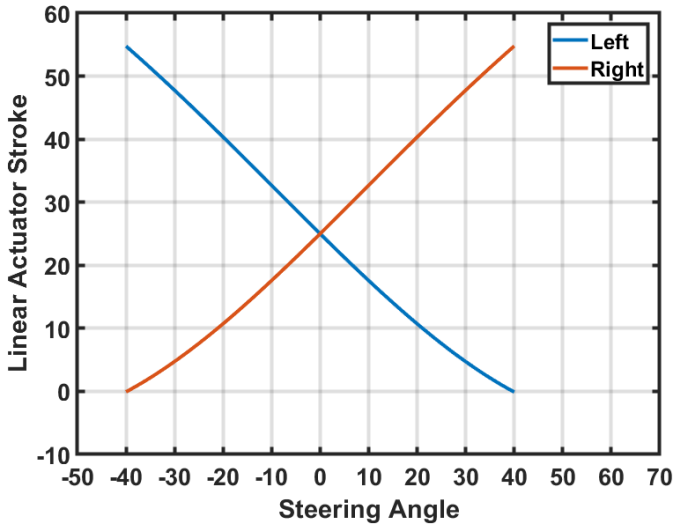

Figure 4-7 Steering Angle vs Actuator Stroke

From the experimental data, equations 4-21 and 4-22 shows 2 third order polynomials that models this relationship. A PID control law is subsequently derived based on the presented actuator model. Figure 4-8 shows the PID control block diagram for steering.

$$Stroke_{left} = (5 * 10^{-5})\delta_L^3 + 0.0014\delta_L^2 - 0.7589\delta_L + 24.974 \quad (4\text{-}21)$$

$$Stroke_{right} = -(5 * 10^{-5})\delta_R^3 + 0.0014\delta_R^2 + 0.7586\delta_R + 24.974 \quad (4\text{-}22)$$

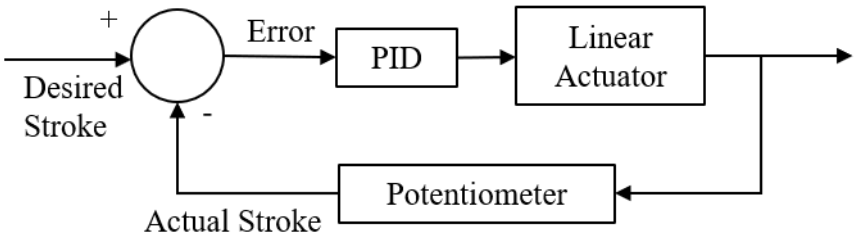

Figure 4-8 Closed Loop Steering Control

4.6 Summary

In summary, this chapter describes the different methodologies that are implemented with the SECV by taking advantage of the software environment developed in the previous chapter to achieve autonomous point to point navigation. These navigation algorithms consist of primarily four steps which are mapping, localization, path planning and low-level control. For mapping, a SLAM algorithm based on particle filters known as FastSLAM 2.0 was described and applied. Next, an incremental localization algorithm using wheel encoders and the IMU was developed to estimate the SECV's pose within the previously acquired map. With both mapping and localization complete, the global map and real-time sensor readings are utilized to generate two-dimensional cost maps. These cost maps show the obstacles from both the acquired map and new sensor readings in a grid-based representation while also inflating the obstacles by a user-specified radius to create an alert zone. The global and local cost maps are subsequently employed by the global and local path planners, respectively. In this work, the global planner is the Dijkstra's Algorithm while the local planner employs the Timed Elastic Band algorithm. These planners are considered as high-level planners and they are converted into achievable commands for the low-level hardware such as the DC motors and linear actuators through PID controllers. In addition, a software differential was implemented to improve cornering maneuverability for DC motors while the actuator stroke to steering angle model was acquired experimentally for optimal performance. All presented methodologies in this chapter are installed with the physical prototype for testing in Chapter 6.

Chapter 5. Pose Correction Algorithm Design

5.1 Introduction

Docking is a maneuver that requires a mobile robot to accurately arrive at a specific pose for reasons such as battery charging, loading/unloading of software and hardware … etc. Depending on the docking scenario, the arrival pose accuracy are often required to be within a few centimeters and degrees of the desired pose which creates a different type of navigation challenge. Although the methodologies presented in the previous chapter can command a mobile platform from point to point without colliding with any obstacles, the result is often accompanied with a high tolerance of error that is attributed to the non-holonomic kinematics constraint mixed with mechanical system imperfections and sensor inaccuracy. Consequently, mobile platforms navigating with these types of methodologies arrive at the desired pose with position and orientation errors that are not accurate enough for docking. Not to mention, the pose error begins to accumulate and magnify when the mobile robot travels long distances or for a long time. In this chapter, a docking algorithm is proposed for the SECV to achieve precise pose correction in an effort to alleviate the problems imposed by traditional navigation methodologies. To do this, visual servo control systems are explored to provide a more accurate and precise close quarters pose correction solution (in this work, close quarters is defined as distances less than 5 meters). The advantage of this technique is its dependence on a visual landmark placed in the world frame to decrease the error tolerance. The following chapter will present the working background of traditional PBVS and propose a modified version that takes advantage of the SECV's multi-steerable design. Lastly, a complete algorithm that combines the

methodologies mentioned in Chapter 4 and the presented modified approach is proposed to achieve an effective and efficient docking maneuver.

5.2 Traditional Position Based Visual Servoing

As briefly reviewed in Section 2.4.3.2 , PBVS is a type of vision-based control system that takes place in the three-dimensional cartesian space [178]. In this technique, the desired pose of a camera relative to a visual landmark is compared to that of the current camera pose to generate an error that is eliminated through robot motion. Traditionally, Lyapunov's proportional control scheme is applied to regulate the robot's position. Likewise, a priori knowledge of a three-dimensional model of the visual landmark is necessary in PBVS; however, homography and epipolar geometry are feasible alternatives for pose estimation in the case that the model is not available. An illustration of the control block diagram is shown below.

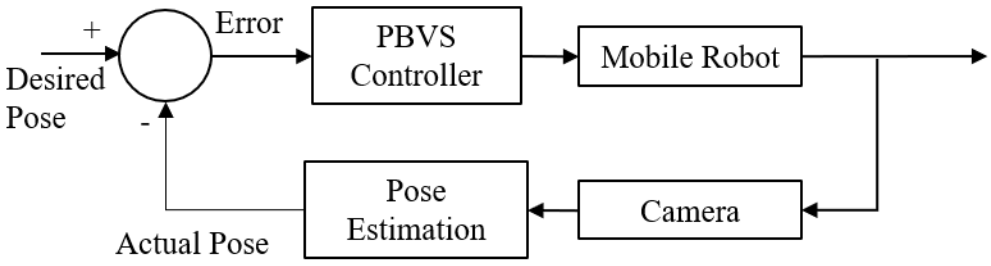

Figure 5-1 Traditional PBVS Control Block Diagram

5.3 Modified Position Based Visual Servoing

To expand on the traditional PBVS controller to capitalize on the SECV's multi-steerable design, a Modified Position Based Visual Servoing (M-PBVS) algorithm is proposed in this section. Instead of the Ackermann steering model described in Section 4.5.1 , the M-PBVS algorithm focuses on two alternative steering configurations which are

Diamond and Synchronous Steer as shown in Figure 5-2. Diamond Steer provides the SECV with the ability to pivot about its center by creating a diamond formation with all its wheels. On the other hand, the Synchronous Steer configuration steers all its wheels in the same direction allowing the vehicle to move at an angle that is constrained by maximum steering ability, δ_{max}. The following sub sections cover the kinematics model of both the steering configurations followed by the design of the M-PBVS controller and algorithm.

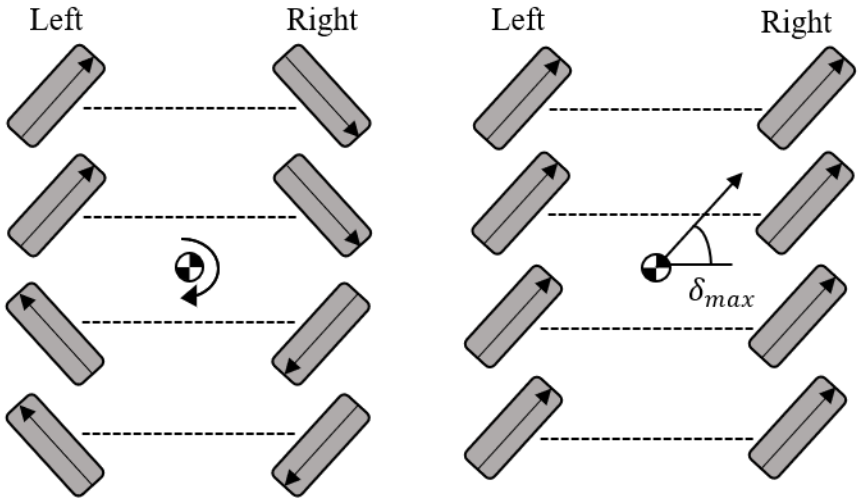

Figure 5-2 Diamond Steer (left), Synchronous Steer (right)

5.3.1 Kinematics Model

The kinematics model of each steering configuration is presented in this section to describe the SECV's motion with a full systems model derived at the end to consider the camera's frame for visual servoing applications.

5.3.1.1 Diamond Steer

The kinematics model of the SECV with Diamond Steer is analogous to that of a mobile robot equipped with differential steer. Traditionally, differential steer systems

consist of a single axle placed at the center of the robot's base with yaw generated by varying the magnitude and/or direction of wheel velocities on either side of the robot. To reduce lateral forces experienced by each wheel, the differential actuated wheels are placed as close to the center of the robot's base as possible. In order to achieve the same for the SECV, the wheels are steered to face towards the center of the chassis. The reason is to reduce tire scrubbing for the first and fourth axle of the SECV as they are placed relatively far apart from each other due to the rectangular footprint of the chassis. In other words, differential steer without the diamond formation would be inefficient, especially on non-smooth surfaces for the SECV.

In Diamond Steer, the angular velocity of the SECV is calculated by finding the product of the quotient between the wheel radius, r, and track width, B, and the difference between the left and right velocities of the vehicle, (ω_L, ω_R). It is important to note that linear velocity is not possible when equipped with this steering configuration.

$$\dot{\theta} = \frac{r}{L}\,(\omega_R - \omega_L) \tag{5-1}$$

The steering angles of each wheel are calculated based on similar triangles as shown below. Since the vehicle is symmetrical between both sides, the steering angles of $(\delta_{L1}, \delta_{R1}, \delta_{L2}, \delta_{R2})$ are equivallent to $(\delta_{R4}, \delta_{L4}, \delta_{R3}, \delta_{L3})$, respectively.

$$\delta_{L1} = \delta_{R4} = -\tan^{-1}(\frac{L}{B}) \tag{5-2}$$

$$\delta_{R1} = \delta_{L4} = \tan^{-1}\left(\frac{L}{B}\right) \tag{5-3}$$

$$\delta_{L2} = \delta_{R3} = -\tan^{-1}(\frac{L}{2B}) \tag{5-4}$$

$$\delta_{R2} = \delta_{L3} = \tan^{-1}(\frac{L}{2B}) \quad (5\text{-}5)$$

The following figure illustrates the kinematics model of the SECV with Diamond Steer.

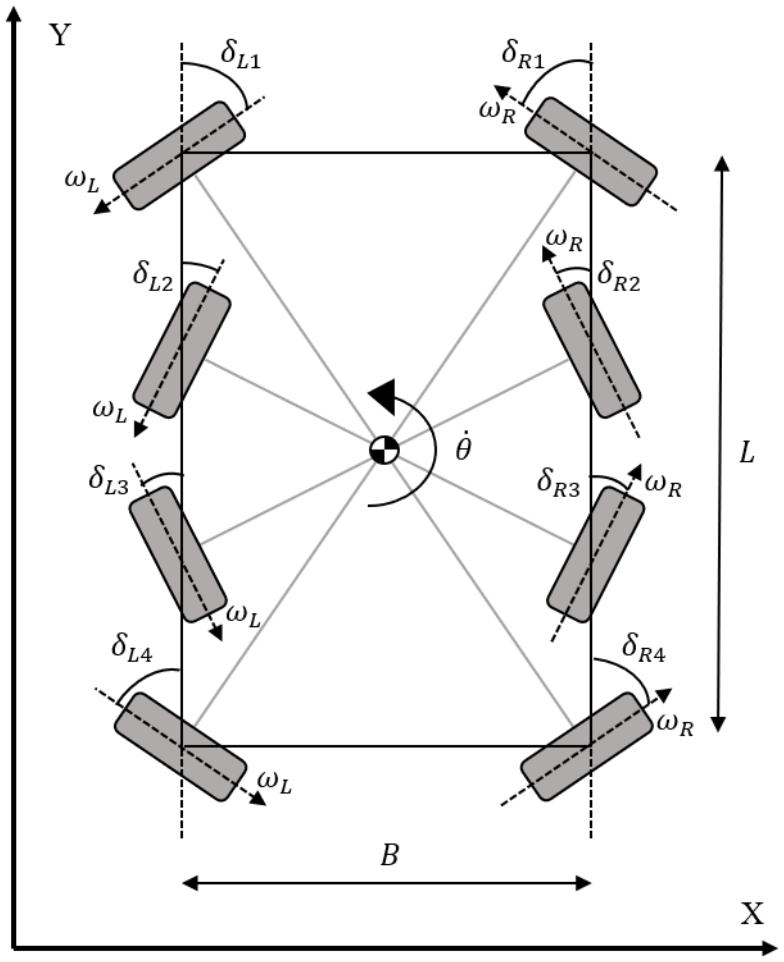

Figure 5-3 Kinematics Model of Diamond Steer

5.3.1.2 Synchronous Steer

Contrarily, the Synchronous Steer configuration allows the SECV to translate in both the *x* and *y* direction without changing the heading angle. This is made possible by steering all eight wheels parallel to each other in the same direction. Differing from what was described in Section 4.5.2, there is no need for a software differential here because all

wheels are traveling the same distance when steered to the same angle. Assuming the wheel speeds, ω, and steering angles, δ, are the joint velocities of a manipulator and SECV's CG velocity, $(\dot{x}, \dot{y})$, are that of the end effector, then the forward kinematics solution is presented below. This manipulator analogy is elaborated further in the next section. It is worth noting that the angular velocity is equivalent to zero because the heading angle is constant with respect to the x axis.

$$\dot{x} = r\,\omega\cos(\delta) \tag{5-6}$$

$$\dot{y} = r\,\omega\sin(\delta) \tag{5-7}$$

$$\dot{\theta} = 0 \tag{5-8}$$

Conversely, the inverse kinematics of the described configuration is as shown below where $sign()$ represent the signum function which outputs either positive or negative 1 based on the value of $\dot{x}$.

$$V = \frac{sign(\dot{x})}{r}\sqrt{\dot{x}^2 + \dot{y}^2} \tag{5-9}$$

$$\delta = \tan^{-1}\left(\frac{\dot{x}}{\dot{y}}\right) \tag{5-10}$$

The kinematics model of the SECV equipped with Synchronous Steer is illustrated in the figure below.

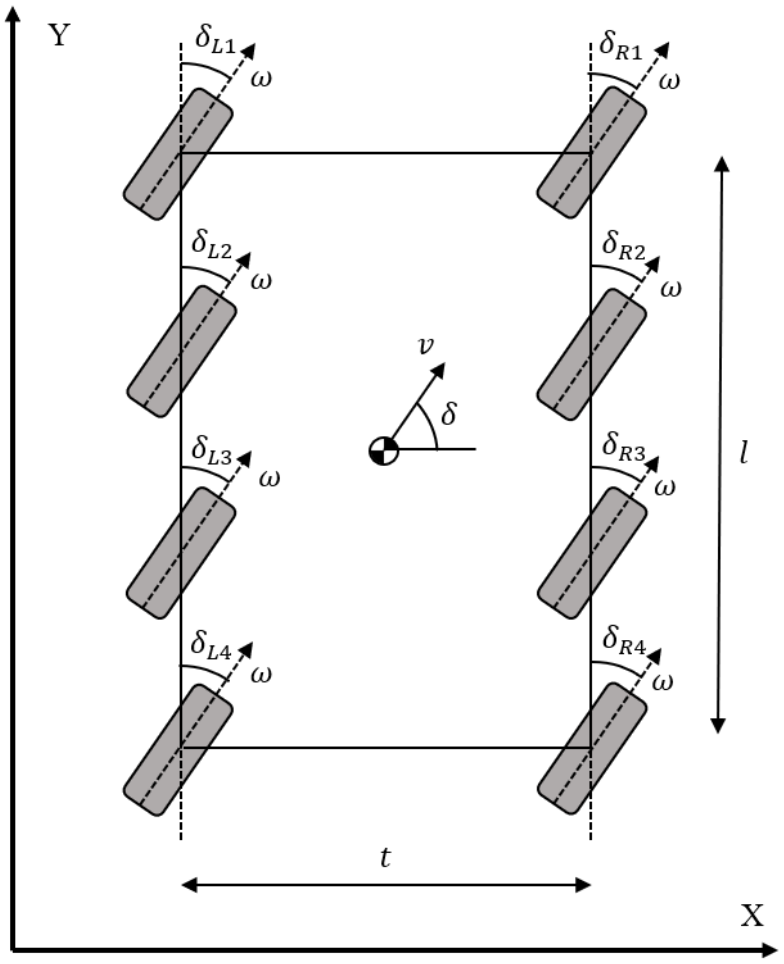

Figure 5-4 Kinematics Model of Synchronous Steer

5.3.1.3 Full System Model

To derive the full kinematics model for visual servoing, there are three frames of importance which are the world frame, the SECV's base frame and the camera frame (the base frame mentioned here is equivalent to base_link when referring to Section 1.3.2.2). The relationship between all these frames is explored in this section. Beginning with the kinematics modeling of the SECV which shares many parallels with that of a manipulator consisting of a revolute and a prismatic joint since the primary goal is position control. In this analogy, the rotation and travel of each joint, respectively, are interpreted as the

angular/steering (depending on the steering configuration) and linear velocity of the SECV with respect to the world. More specifically, the end effector frame, $\{eff\}$, coincides with the base frame, $\{base\}$, which is located at the center of the SECV. Accordingly, the origin of the manipulator base frame, $\{0\}$, which overlaps the revolute joint frame, $\{1\}$, is symbolic of the world frame that the mobile robot exists in which is demonstrated in Figure 5-5. Using this model, the position and velocity of the end effector is equivalent to that of the SECV which is obtained by solving the manipulator's forward velocity (FV) problem. It is worth noting that the orientation of the mobile robot with respect to itself is not considered in this model as the primary concern is position regulation.

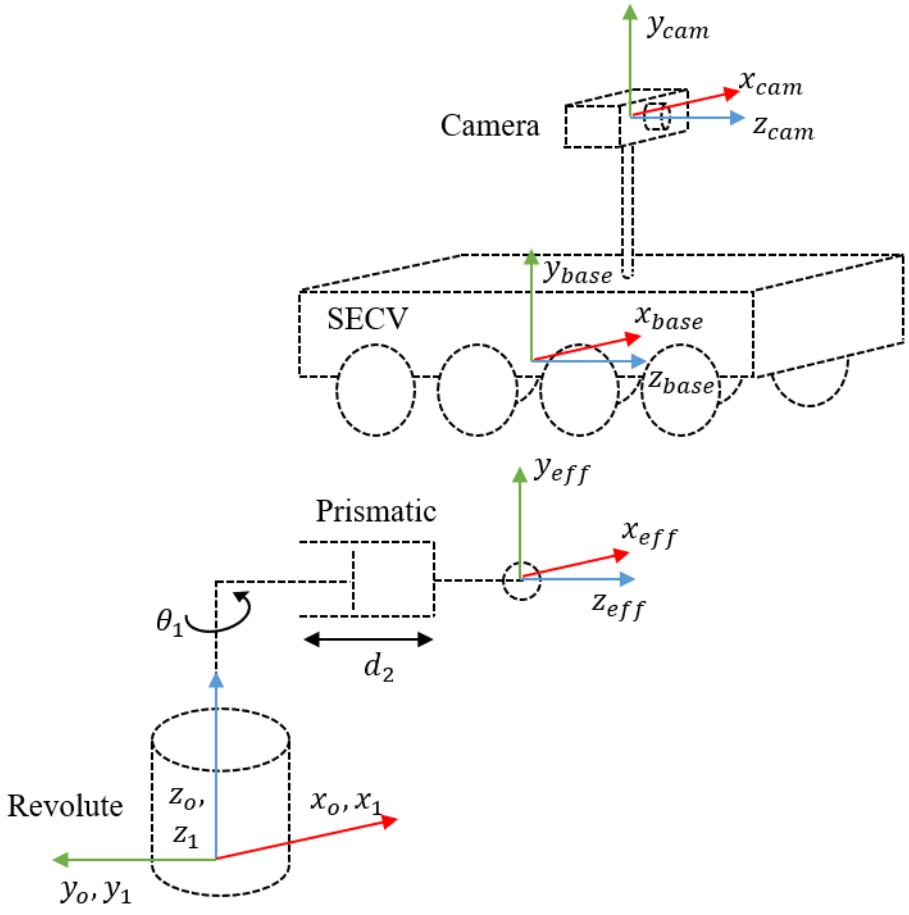

Figure 5-5 Full Kinematics Model

To solve the FV problem, a zero-displacement diagram is drawn with the Denavit Hartenberg (DH) parameters extracted and summarized as shown in Table 5-1.

Table 5-1 DH Parameters

$\mathbf{F_{i-1}}$	$\mathbf{a_{i-1}}$	$\boldsymbol{\alpha_{i-1}}$	$\mathbf{d_i}$	$\boldsymbol{\theta_i}$	$\mathbf{F_i}$
0	0	0	0	θ_1	1
1	0	$\pi/2$	d_2	0	2

From the above table, the homogeneous transformation matrix between the SECV's base frame, $\{eff\}$, and the world frame, $\{0\}$ is derived as shown below.

$$ {}^{0}_{eff}T = \begin{bmatrix} cos\theta_1 & 0 & sin\theta_1 & sin\theta_1 d_2 \\ sin\theta_1 & 0 & -cos\theta_1 & -cos\theta_1 d_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5-11} $$

By establishing the relationship between different coordinate frames, a Jacobian matrix is derived to relate the velocity of the end effector with respect to the world, ${}^{0}_{eff}V$, to the individual joint velocities $[\dot{\theta}_1 \quad \dot{d}_2]^T$. This is shown in equation 5-12 which represents the solution to the FV problem as the joint velocities, $(\dot{\theta}_1, \dot{d}_2)$ signify the robot's angular and linear velocities, respectively. More specifically, $\dot{\theta}_1$ represents SECV's angular velocity during Diamond Steer and steering velocity during Synchronous Steer as discussed in later sections. This model restricts the movement of the robot to the x-y plane of the world frame with lateral slip neglected.

$$ {}^{0}_{eff}V = \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} d_2 C_1 & S_1 \\ d_2 S_1 & -C_1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{d}_2 \end{bmatrix} = J \begin{bmatrix} \dot{\theta}_1 \\ \dot{d}_2 \end{bmatrix} \tag{5-12} $$

5.3.2 M-PBVS Algorithm Design

The M-PBVS algorithm is proposed in this section to take advantage of the presented steering configurations in a two-stage approach. The following subsections will discuss the controller design in each stage.

5.3.2.1 Stage One: Orientation Control

The first stage of the M-PBVS algorithm is to correct the orientation of the SECV using Diamond Steer. To initiate the algorithm, a closed loop orientation controller that utilizes the onboard IMU sensor is employed at first. The idea is to only borrow IMU's inaccurate readings at the beginning to decide on a pivot direction. However, this approach will not be able to accurately arrive with the intended orientation since this approach is prone to drift issues. To mitigate this problem, the proposed M-PBVS approach utilizes the onboard camera sensor to search for the visual landmark simultaneously as the SECV pivot. Once the landmark is detected, a closed loop orientation control using purely vision as feedback is deployed until the longitudinal axis of the SECV is perpendicular to the x axis of the visual landmark. By doing so, the SECV does not depend on its odometry sensors rather just the accuracy of the pose estimation algorithm. The flow diagram is illustrated in Figure 5-6 with the pivot action shown in step 1 and 2 of Figure 5-7.

To accomplish this, Lyapunov's control scheme is applied as shown in equation 5-13 where k represent the proportional gain. The error term for orientation, $e(t)_w$, is the difference between the current and desired image feature sets, (s, s_d), as extracted by a pose estimation algorithm. In this work, the pose estimation algorithm implemented is developed by [179] where the output is the position and orientation estimation of the visual landmark. The proposed M-PBVS algorithm utilizes this information and formulate the

control law with respect to the desired camera frame; therefore, the orientation error term is equivalent to the orientation vector, ${}^{cam_d}_{cam}\phi$, since the desired orientation is zero.

$$e(\dot{t})_w = -k\, e(t)_W = -k(s - s_d)_w = -k\left({}^{cam_d}_{cam}\phi - 0\right)_w \tag{5-13}$$

Moving forward, the derivation of the orientation control law begin with representing the angular velocity of the camera with respect to the desired camera frame, ${}^{cam_d}_{cam}w$, as the angular velocity of the camera with respect to itself, ${}^{cam}_{cam}w$ with the help of a rotation matrix, ${}^{cam_d}_{cam}R$.

$${}^{cam}_{cam}w = \left({}^{cam_d}_{cam}R\right)^T {}^{cam_d}_{cam}w \tag{5-14}$$

Next, a transformation matrix, $T(\phi)$, is applied to convert the orientation expressed in Euler's angles, ${}^{cam_d}_{cam}\dot{\phi}$, in to angular velocities of the camera relative to the desired camera frame, ${}^{cam_d}_{cam}w$.

$${}^{cam_d}_{cam}w = T(\phi)\,{}^{cam_d}_{cam}\dot{\phi}, where\ T(\phi) = \begin{bmatrix} 0 & -sin\varphi & cos\varphi * cos\theta \\ 0 & cos\varphi & sin\varphi * cos\theta \\ 1 & 0 & -sin\theta \end{bmatrix} \tag{5-15}$$

Accordingly, equation 5-15 is substituted into equation 5-14 to form the following.

$${}^{cam}_{cam}w = \left({}^{cam_d}_{cam}R\right)^T T(\phi)\,{}^{cam_d}_{cam}\dot{\phi} \tag{5-16}$$

Since $e(t)_w$ is equivalent to ${}^{cam_d}_{cam}\phi$; as a result, the above equation is rewritten as below which considers the rate of the error, $e(\dot{t})_w$.

$${}^{cam}_{cam}w = \left({}^{cam_d}_{cam}R\right)^T T(\phi) * e(\dot{t})_w \tag{5-17}$$

By substituting Lyapunov's control scheme from equation 5-13 into equation 5-17, the following control law for the angular velocity of the camera frame is derived.

$$ {}^{cam}_{cam}w = -k \left({}^{cam_d}_{cam}R \right)^T T(\phi) * e(t)_w \tag{5-18} $$

Next, the full kinematics model from Section 5.3.1.3 is combined with equation 5-18 to consider the base frame and wheel velocities. Since the SECV is only pivoting without translation at this stage; therefore, the two joint manipulator analogy from Figure 5-5 is reduced to just the revolute joint. As a result, the angular velocity of the SECV's base is related to the angular velocity of the camera as shown in equation 5-19 where, ${}^{base}_{cam}R$, represent an identity rotation matrix because the camera is rigidly attached to the base.

$$ {}^{base}_{base}w = {}^{base}_{cam}R \; {}^{cam}_{cam}w \tag{5-19} $$

Next, the angular velocity of the base, ${}^{base}_{base}w$, is related to the angular velocity of the base with respect to the world, ${}^{0}_{base}w$ by the following equation where the rotation matrix, ${}^{0}_{base}R$, is extracted from the homogenous transformation matrix from Table 5-1 using only the first row.

$$ {}^{0}_{base}w = {}^{0}_{base}R \; {}^{base}_{base}w \tag{5-20} $$

Using the FV solution from equation 5-12, the angular velocity of the SECV's base frame is described based on the angular joint velocity, $\dot{\theta}$, using a Jacobian matrix, J_w, as seen below.

$$ {}^{0}_{base}w = J_w \, \dot{\theta} \tag{5-21} $$

By combining equations 5-18 to 5-21, the complete law that controls the angular velocity of the SECV based on the orientation error is shown below. It is important to note that a pseudo-inverse for the Jacobian matrix, J_w^+, is applied since it is a non-square matrix.

$$\dot{\theta} = -k\, J_w^{+}\ {}_{base}^{0}R\ {}_{cam}^{base}R\ \left({}_{cam}^{cam_d}R\right)^T\ T(\phi)\ e(t)_w \tag{5-22}$$

Using differential kinematics, the angular velocity of the SECV is further expressed in terms of wheel velocities as shown in equation 5-23 with the flow diagram thereafter.

$$\dot{\theta} = \begin{bmatrix} -\frac{r}{L} & \frac{r}{L} \end{bmatrix} \begin{bmatrix} \omega_{left} \\ \omega_{right} \end{bmatrix} \tag{5-23}$$

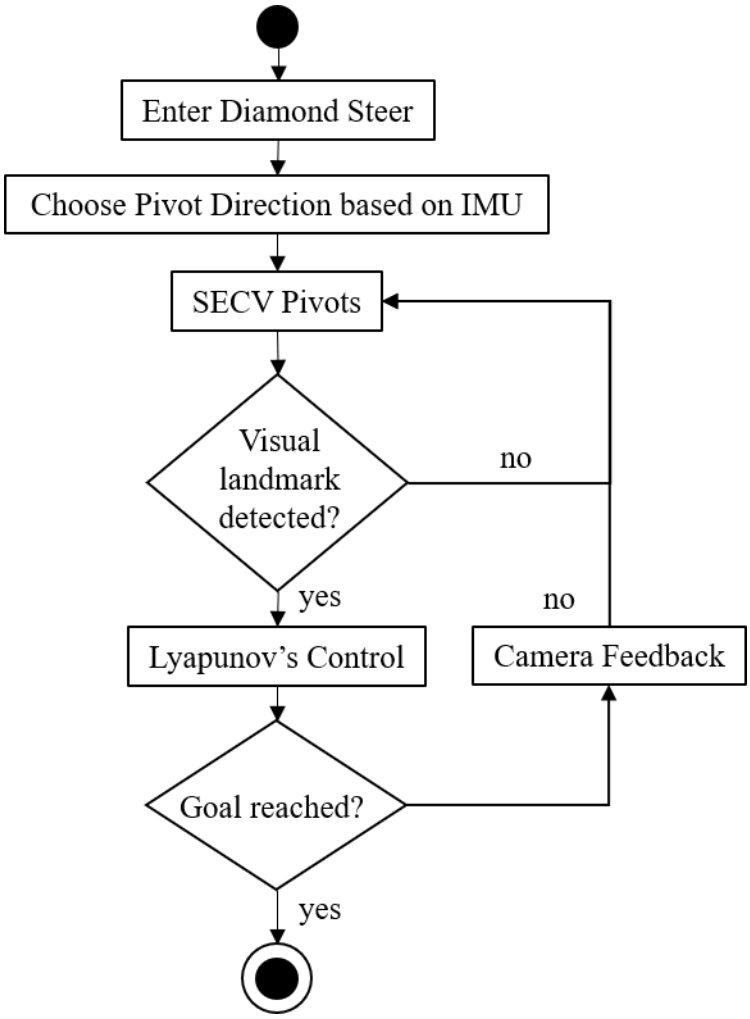

Figure 5-6 M-PBVS Activity Diagram: Stage One

5.3.2.2 Stage Two: Position Control

Once the orientation is corrected, the M-PBVS algorithm enters the second stage which utilizes the Synchronous Steer configuration. Since the SECV features a maximum steering angle, δ_{max}, of 30 degrees, two scenarios of control are possible; namely, one with a direct goal and the other with an alternate goal. The first scenario happens when the angle to the desired position, $\delta_{approach}$, is within the maximum steering angle as illustrated below.

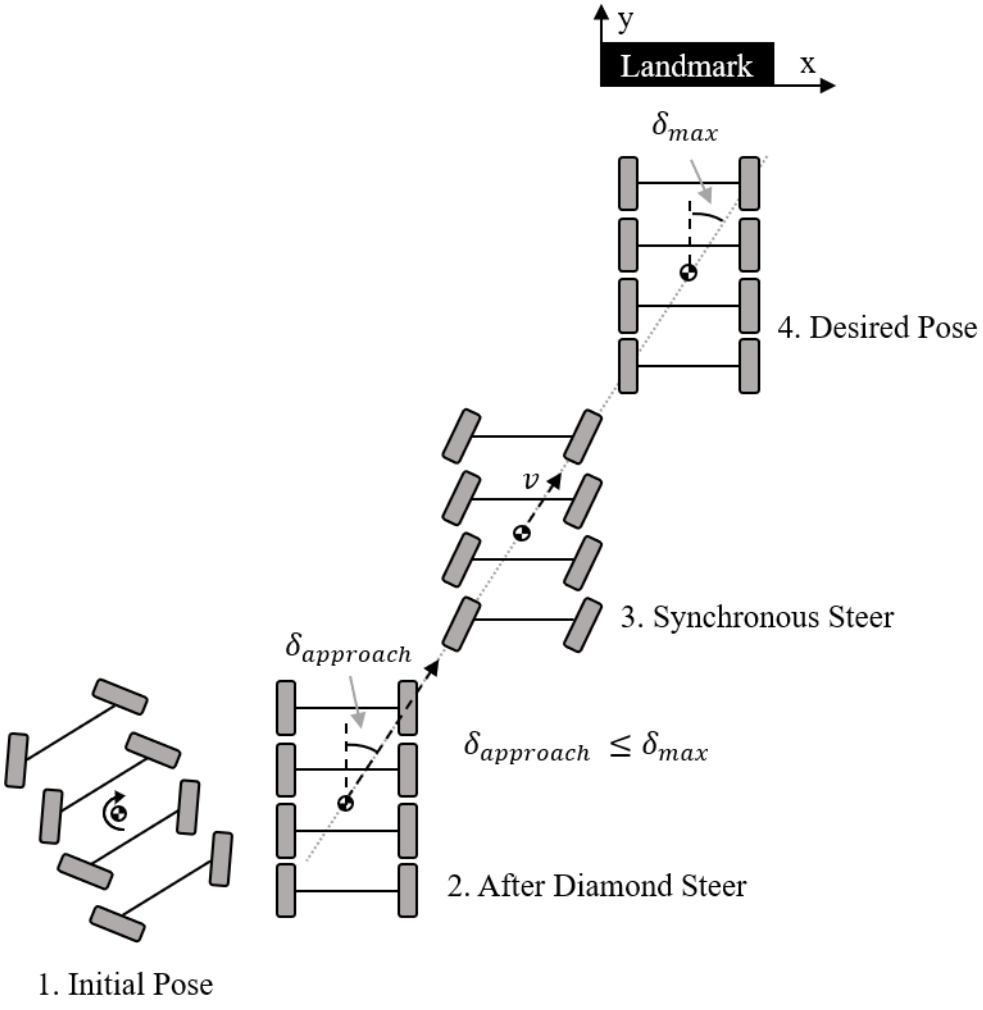

Figure 5-7 M-PBVS with Direct Goal

From there, the proposed M-PBVS controller proceeds to minimize the position error based on vision feedback. Since the heading angle of the SECV does not change during Synchronous Steer, the visual landmark remains within the camera's field of view during its course. In the second scenario, the initial estimated approach angle is greater than the maximum steering angle; as a result, an alternative goal is calculated to re-position the SECV in a position that would achieve the direct scenario. This is illustrated below.

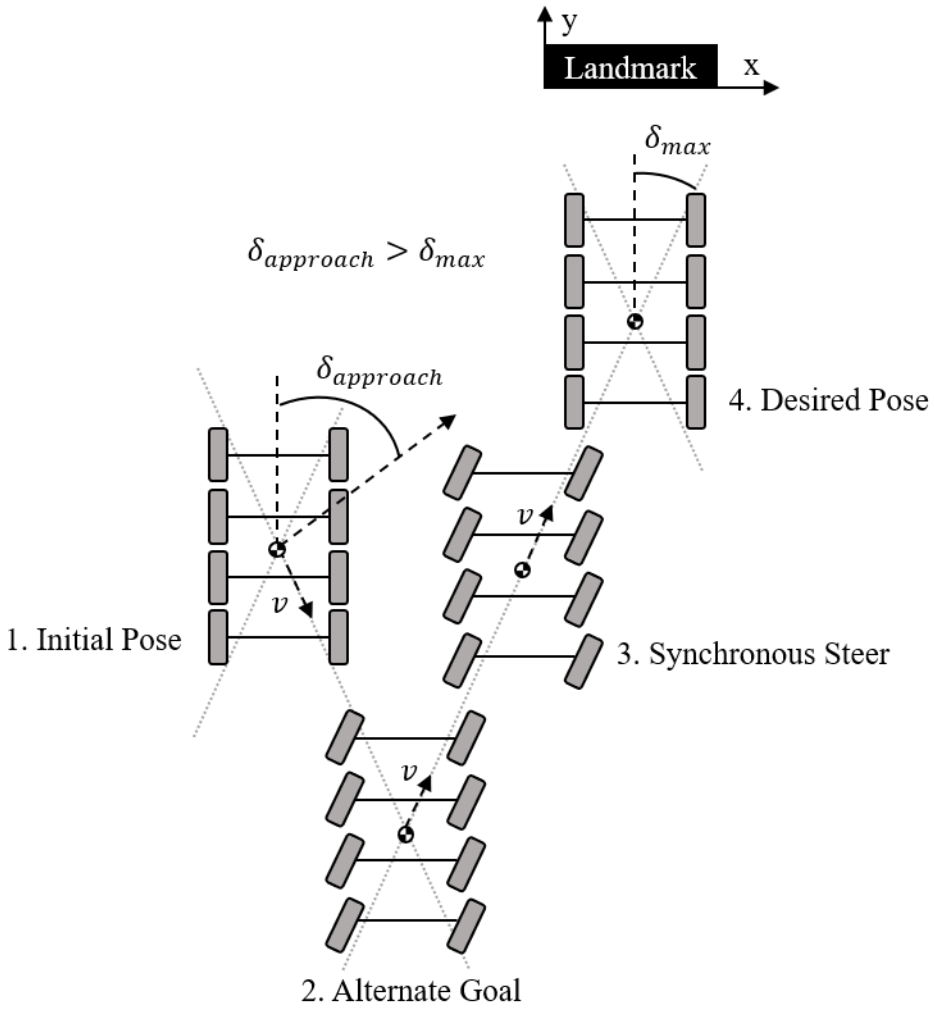

Figure 5-8 M-PBVS with Alternate Goal

More specifically, to find the approach angle and thus determine which of the two scenarios the current SECV is in, the desired position coordinates, (x_d, y_d), are compared with the current, (x, y) as shown in equation 5-24. The approach angle is then measured against the maximum steering angle to determine which is larger in value.

$$\delta_{approach} = \frac{y_d - y}{x_d - x} \tag{5-24}$$

If the approach angle is less than the maximum steering angle then the SECV is in the direct goal scenario where the steering angles are set to the following.

$$\delta_{steering} = \tan^{-1}(\delta_{approach}) \tag{5-25}$$

If the approach angle is greater than the maximum steering angle, then the SECV requires an alternate goal as mentioned previously. The alternate goal is calculated by finding the intersection between lines extended from the current and desired positions at the maximum steering angle as calculated by equations 5-26 and 5-27. These lines are represented by grey dash lines in Figure 5-9.

$$current: y = \tan(\pm\delta_{max})\ x + b \tag{5-26}$$

$$desired: y_d = \tan(\pm\delta_{max})\ x_d + b_d \tag{5-27}$$

By doing so, the alternate goal is guaranteed to be within the achievable steering angle. Once the SECV have arrived at the alternate goal, the new approach angle towards the desired goal should be equivalent to the maximum steering angle; thereby making it a direct goal scenario as shown in Figure 5-8. It is important to note that there are two alternate goal solutions at each initial positions because of paths calculated based on positive and negative δ_{max}. To solve this issue, the alternate goal with the shortest distance from its

current position is always selected. For example, alterrnate goal scenarios in Quadrant 1 and 2 would always result in the SECV generating a forward velocity as shown by Position D in Figure 5-9.

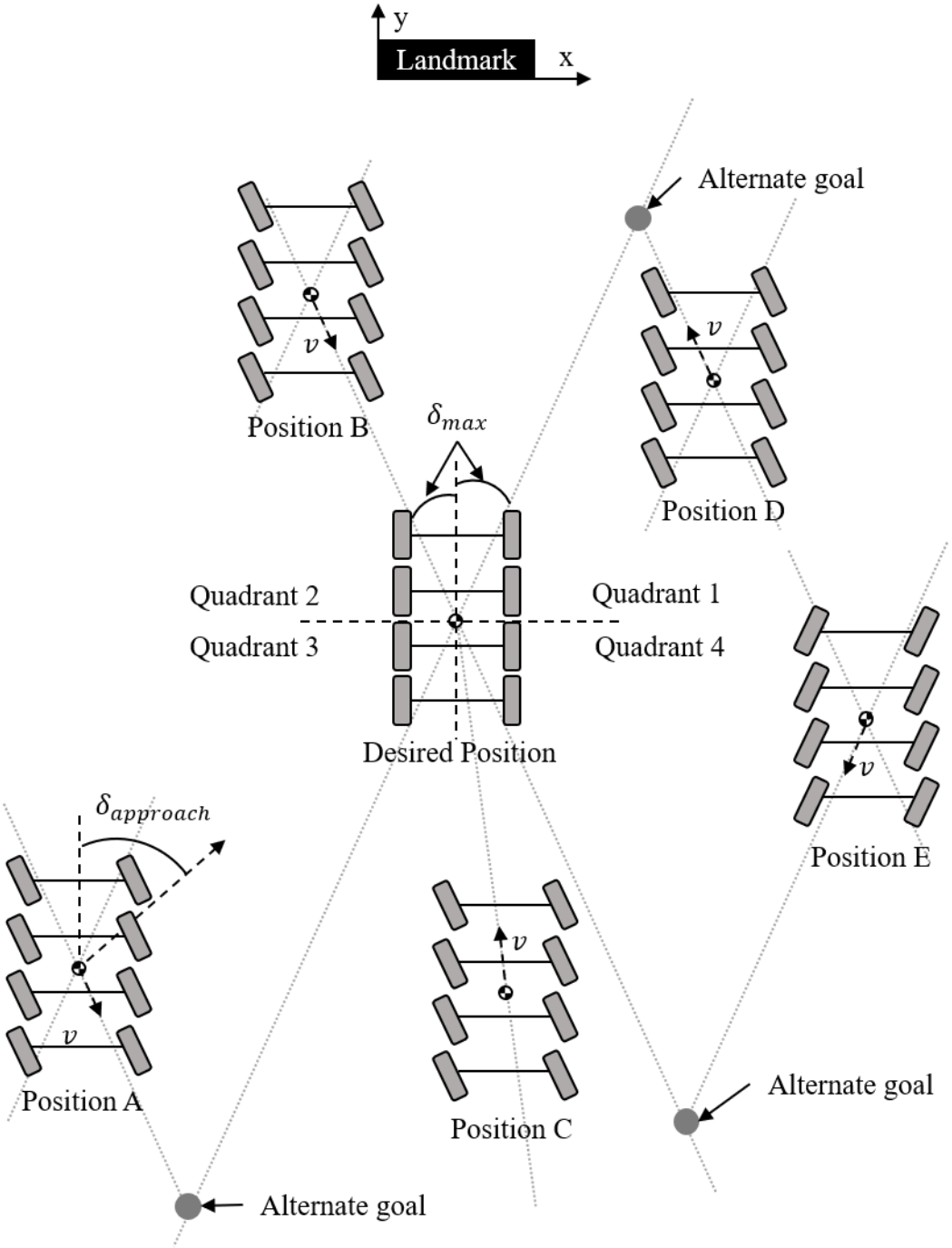

Figure 5-9 Two Scenarios for Stage Two of M-PBVS

On the other hand, the opposite is true for alternate goal scenarios in Quadrant 3 and 4 as illustrated by Positions A and E where the shortest distance requires the SECV to reverse. Position B and C belong to the direct goal scenario as they fall within the achievable steering angle.

Once the direct or alternate goal positions are either detected or calculated, Lypapunov's proportional control scheme is applied in a similar fashion to stage one. However, the only difference is that the error term in the second stage consists of two vectors which are the translational vector, ${}^{cam_d}_{cam}t$, and the orientation vector, ${}^{cam_d}_{cam}\phi$. Starting with the latter, the derivation of the orientation control for stage two is the same as stage one; however, steering velocity, $\dot{\varphi}$, is used for Synchronous Steer instead of the angular velocity from equation 5-22. In addition, the revolute joint's angle is constrained by the maximum steering angle. The following illustrates the steering control law.

$$\dot{\varphi} = -k\, J_w^{+}\, {}^{0}_{base}R\, {}^{base}_{cam}R \left({}^{cam_d}_{cam}R\right)^{T} T(\phi)\, e(t)_w \qquad (5\text{-}28)$$

$$where - \delta_{max} < \theta_1 < \delta_{max}$$

On the other hand, Lyapunov's control scheme is applied to the translational vector similarly to equation 5-13 for orientation as shown below where $e(t)_t = {}^{cam_d}_{cam}t$.

$$\dot{e(t)}_t = -k\, e(t)_t \qquad (5\text{-}29)$$

Next, the translational vector is represented in terms of translational camera velocity, ${}^{cam}_{cam}v$, as shown below.

$${}^{cam}_{cam}v = \left({}^{cam_d}_{cam}R\right)^{T} {}^{cam_d}_{cam}\dot{t} \qquad (5\text{-}30)$$

From here, the rate of change of the translational vector, ${}^{cam_d}_{cam}\dot{t}$, is equivalent to the rate of the error as suggested by equation 5-29. As a result, the translational control law is written as follows which is similar to what was derived for the orientation in equation 5-18.

$$ {}^{cam}_{cam}v = -k\left({}^{cam_d}_{cam}R\right)^T e(t)_t \tag{5-31} $$

The above control law is capable of controlling the translational velocity of the camera; however, it is not complete in the sense that it does not consider SECV's base frame. Therefore, rotation matrices between the world and camera frame from equation 5-11 is substituted into equation 5-31 to formulate the translational control law below.

$$ v = -k\, J_t^+ \, {}^0_{base}R \; {}^{base}_{cam}R \left({}^{cam_d}_{cam}R\right)^T e(t)_t \tag{5-32} $$

Lastly, the full control law for stage two of the proposed M-PBVS is completed by combining the steering control from equation 5-28 and the translational control as seen above to form the following where $s(d)$ is a skew matrix that describes the camera's position with respect to the base frame.

$$ \begin{bmatrix} v \\ \dot{\varphi} \end{bmatrix} = -k\, J^+ \, E \, L \, e(t) \tag{5-33} $$

$$ E = \begin{bmatrix} {}^0_{base}R & 0_{3x3} \\ 0_{3x3} & {}^0_{base}R \end{bmatrix} \begin{bmatrix} {}^{base}_{cam}R & s(d)\,{}^{base}_{cam}R \\ 0_{3x3} & {}^{base}_{cam}R \end{bmatrix} $$

$$ where - \delta_{max} < \theta_1 < \delta_{max} \tag{5-34} $$

$$ s(d) = \begin{bmatrix} 0 & -r_Z & r_y \\ r_z & 0 & -r_x \\ -r_y & r_x & 0 \end{bmatrix} $$

$$ L = \begin{bmatrix} \left({}^{cam_d}_{cam}R\right)^T & 0 \\ 0 & \left({}^{cam_d}_{cam}R\right)^T * T(\phi) \end{bmatrix} \tag{5-35} $$

The following illustrates the activity diagram of stage two of the proposed M-PBVS approach.

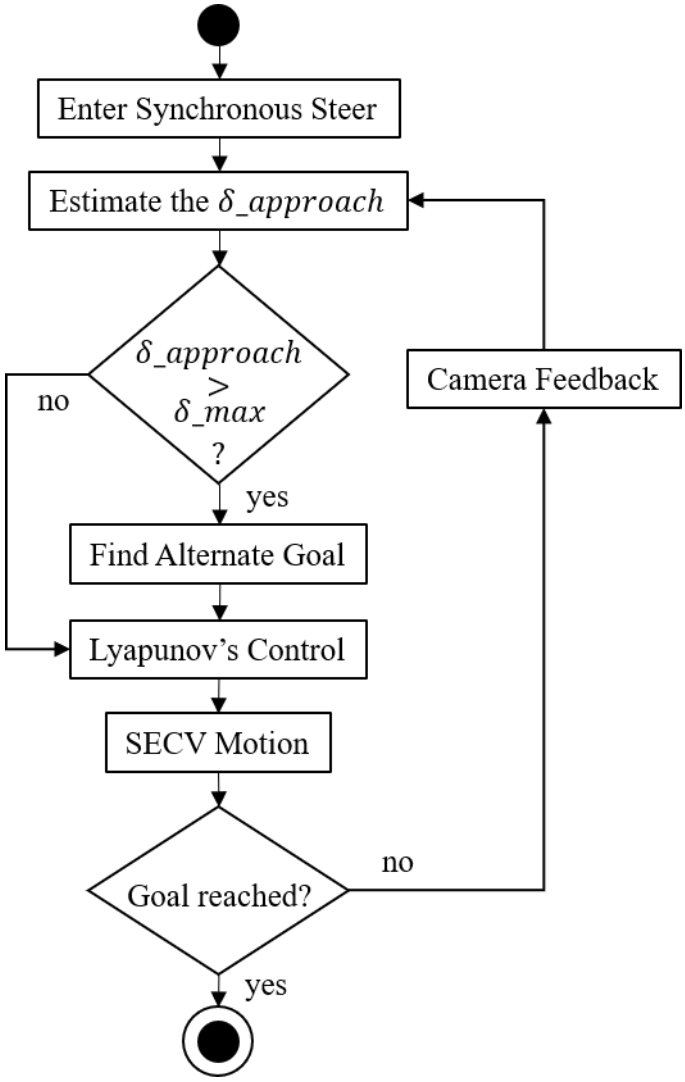

Figure 5-10 M-PBVS Activity Diagram: Stage Two

5.4 Full Docking Algorithm

As mentioned in the introduction, docking is a type of maneuver that requires the mobile platform to accurately arrive at the desired pose for purposes such as charging, loading…etc. Navigation methodologies presented in the previous chapter alone will not be able to meet the tight tolerance that docking requires; therefore, the proposed M-PBVS algorithm presented in this chapter is added to the end of regular navigation algorithms to improve upon this limitation. The pseudocode of the proposed mobile robot docking approach is presented below. Firstly, the previously acquired map is launched along with the localization algorithm. The global planner generates a path from the SECV's current location to the desired docking location. Next, the TEB local planner is deployed to generate achievable velocities to avoid any obstacles along its path. Once the TEB local planner brings the SECV to the docking area, the M-PBVS algorithm will utilize the external visual landmark to accurately correct its position and orientation to match that of the desired in the presence of odometry sensor inaccuracy and drift.

Table 5-2 Docking Algorithm Pseudocode

1.	launch map
2.	launch localization algorithm
3.	run Dijkstra's Algorithm
4.	while (SECV != desired docking vicinity)
5.	run TEB
6.	while (SECV != docked)
7.	run M-PBVS

5.5 Summary

In summary, this chapter presents a close quarter pose correction approach, called the M-PBVS algorithm which builds on top of the traditional PBVS by including alternative steering configurations. The intent is to create a precise vision-based navigation method that is not susceptible to odometry inaccuracy and drift. To do so, the proposed algorithm features two stages with each taking advantage of an alternative steering configuration; namely, Diamond and Synchronous Steer. The kinematics model of each steering configuration and its combination with the full SECV system that considers the position of the camera is derived. From there, two control laws using Lyapunov's proportional scheme is derived for each of the stages. The first stage corrects the orientation error by using Diamond Steer to pivot about the SECV's center. The second stage detects the goal position with respect to a visual landmark and subsequently calculates the appropriate steering angle to either reach the desired pose directly or find an alternative goal. The result of the two-stage algorithm is a grid that consists of intersections between lines with slopes that represent the maximum steering angle. Lastly, the proposed M-PBVS algorithm is attached to the navigation methodologies presented in Chapter 4 to form a complete, accurate docking algorithm. In the next chapter, this approach is evaluated for its effectiveness and efficiency when compared with AWS planners such as the TEB.

Chapter 6. Results and Discussion

6.1 Introduction

In this chapter, the proposed SECV prototype is subjected to various testing to assess its operational performance, autonomous navigation abilities as well as the proposed M-PBVS algorithm. For vehicle operations, attributes such as max speed, Ramp Travel Index, steering time, max grade, turning radius…etc., are studied. These tests will provide an in-depth look at the performance of the proposed prototype. Next, the navigation algorithms described in Chapter 4 are implemented with the onboard computer and sensors to evaluate its obstacle avoidance capability. By doing so, the drift issue over long distance navigation is also examined. This leads into physical testing of the proposed M-PBVS approach to evaluate its ability to conduct close quarters pose correction. Lastly, the results from this are compared with the performance of the standard AWS planner to further validate the improvements that it offers.

6.2 Vehicle Performance Experiments

Starting with vehicle performance tests which are intended to quantify the functionality and operational limits of the SECV. The results from the following series of tests are tabulated.

6.2.1 Max Speed, Acceleration, Braking Distance and Steering Time

To find the maximum speed, the SECV is driven over two distances in a straight line, one at 309 inches (7.85m) and another at 515 inches (13.08m), to ensure that the vehicle can carry its speed over both short and long distances. In addition, acceleration time is accounted by providing a five-meter head way to ensure the vehicle have reached top speed

before entering the tests. By timing the SECV over the two tests, the vehicle speed is calculated. The following table summarizes the test results where the average max speed of the prototype is approximately 6.51 km/h. Using a smartphone application, the SECV acceleration was obtained to be approximately 1.64 m/s^2. This is an acceptable speed and acceleration for the SECV prototype. If higher speeds are required, it is possible to change the pulley box ratio as mentioned in the Section 3.2.3 to increase the motor's output speed.

Table 6-1 Max Speed Results

Run	Distance (in)	Time (s)	Speed (km/h)
1	309	4.28	6.60
2	515	7.34	6.42

The minimum braking distance test assesses the vehicle's braking capability when traveling at maximum speed. This information is not only good to know for its operator but is crucial when developing autonomous and active safety systems for the SECV. The braking time is also measured from this value.

Table 6-2 Minimum Braking Distance

Test	Results
Minimum Braking Distance (ft)	3.25
Minimum Braking Time (s)	0.5

The steering time test is used to determine the lock-to-lock (full steer left to full steer right) response time. A quick steering time makes the vehicle appear "snappy," and able to quickly maneuver around obstacles; however, the vehicle becomes more difficult to control at higher speeds which creates problems in the adaptation of autonomous systems. Slower

steering response is easier for implementing autonomous features and is a better fit for this autonomy research platform. The result of this test is 0.8 seconds which is deemed acceptable.

6.2.2 Ramp Travel Index

The goal of this test is to determine the Ramp Travel Index (RTI) value of the SECV. The RTI_{20} is an index used to classify vehicles based on their suspension flex on a 20-degree ramp. Stock SUVs have an RTI between 400 and 550, and some off-road competition vehicles may even surpass 1000. During the test, one of the front-most wheels is lifted until another wheel, often the next consecutive wheel behind the control wheel, lifts off the ground. The first measurement needed is the distance between the control wheel and the next consecutive wheel. The second measurement is the highest point the control wheel reaches before another wheel leaves the ground. The calculation of the RTI is completed with the following equation. The result of the SECV is an RTI of 2616.21 which is significantly better than any commercially available passenger vehicle.

$$RTI_{20} = \frac{control\ wheel\ height}{distance\ to\ second\ wheel} * \frac{1000}{sin20°} \quad (6\text{-}1)$$

Table 6-3 Ramp Travel Index Result

Wheel Height	Distance to 2nd wheel	RTI
8.5 in	7.5 in	2616.21

6.2.3 Static Roll Threshold and Maximum Grade

The intent of the static roll threshold test is to measure the roll-over potential of a vehicle. Though most critical in the transport truck industry, a vehicle's rollover threshold in emergency maneuvers or around sharp bends is an important vehicle specification. While it is extremely unlikely that the SECV would roll on the flat ground, this threshold would help the operator understand the vehicle's limitations, especially when cornering at high speeds and navigating uneven and hilly terrain. This test is performed by placing the vehicle on a ramp and then subsequently increasing the angle until the vehicle slides or roll over. The following table summarizes the results of this test which are a testament to the low center of gravity and wide stance that the vehicle exhibits. The maximum ramp angle that the vehicle could achieve was 52 degrees which gives a tilt table ratio of 1.19g.

Table 6-4 Static Roll Threshold Results

Degrees	Results
5	No issues
15	No issues
25	No issues
35	No issues
45	Vehicle begins to get light on raised wheels
50	Vehicle light on raised wheels
52.5	Vehicle rollover

The goal of this test is to determine the maximum grade that the SECV can effectively climb. To perform this test, an adjustable ramp that varies between 5 and 45 degrees, in increments of 5 degrees, is created. The SECV is then aligned with the base of the ramp where the throttle is increased until the SECV either climbs the ramp or becomes static. The test began with the ramp at 5 degrees and increase by 5 degrees after every successful run. The maximum angle is determined at the angle which the vehicle can mount the ramp but not ascend.

Table 6-5 Maximum Grade Results

Degrees	Results
5	Able to climb
10	Able to climb
15	Able to climb
20	Able to climb
25	Able to climb
30	Able to overpower friction; however, starting to slip
35	Unable to climb.

6.2.4 Turning Radius and Slalom

The goal of this test is to determine and compare, the minimum turning radius for each of the steering configurations mentioned in Section 3.2.4 . To find the minimum radius, the SECV starts from rest with the maximum steering angles set. The throttle is then set to a constant speed as the SECV begins driving in a circle. To ensure that the turning circle is accurate, the SECV drives two laps successively. If the SECV ends back at the starting position after both laps, then the diameter will be recorded, and halved, resulting

in the minimum turning radius. The following table shows the minimum turning radius between the different steering configurations.

Table 6-6 Minimum Turning Radius Results

Steering Configuration	Minimum Radius
Front Wheel Steer	77.88 inches
4^{th} Axle Steer	54.88 inches
All Wheel Steer	42.63 inches

The slalom test is another steering test that evaluates the SECV's maneuverability. In this test, the vehicle must zig zag between a series of pylons placed at equal distances apart in a straight line. The test is for all steering configurations with the minimum pylon distance tabulated below. The results obtained from this test corresponds with the minimum steering radius test in showing that the all-wheel steer configuration offers the tightest turning ability.

Table 6-7 Slalom Test Results

	Front Wheel Steer	4^{th} Axle Steer	All Wheel Steer
Pylon Distance (m)	2.44	1.83	1.52

6.3 Autonomous Navigation Experiments

In this section, the autonomous navigation ability of the SECV is experimentally tested. Obstacles that did not exist during the mapping phase is placed between the SECV and its desired pose. As mentioned in Section 3.4 and 4.4 , global and local path planners are implemented within the ROS network as illustrated in Figure 3-9. The intent is to evaluate the planners' ability to consider both the map and real-time sensor data. Additionally, these tests will also confirm the lower controller's ability to execute high-level plans accordingly. A total of three scenarios are chosen; namely, One Obstacle, Slalom and Parking. For the first test, only a single obstacle is placed between the SECV and its goal. Next, the Slalom test evaluates the vehicle's ability to maneuver between two obstacles in both directions. In addition, this test also evaluates the SECV's ability to account for obstacles that appear suddenly as one obstacle is hidden from the vehicle's initial sensor range. The parking test is intended to evaluate the ability to maneuver in tight spaces. All three tests represent common scenarios for an autonomous platform in a point to point navigation application. Lastly, an odometry drift test is conducted to evaluate the implication of drift with respect to total distance traveled to provide insight into the sensors' accuracy and precision.

6.3.1 Experimental Setup

As mentioned in Section 5.3.1.3 , the base frame of the SECV is located at the bottom center of the chassis with the positive *x* axis (blue) and *y* axis (green) pointing forward and to the left, respectively, as seen in Figure 6-1. The sensor bridge described from Section 3.3.1 is installed on top of the chassis to mount the laser scanner and IMU (a camera is included but not used in this experiment). Other hardware necessary for this test includes

the laptop, batteries, lower controllers and encoders for each wheel. Additionally, the SECV is set to the AWS configuration throughout all testing since it offers the best cornering abilities based on the results from Section 6.2 . Furthermore, all experiments are conducted in an indoor environment with flat smooth surfaces and opaque, rigid obstacles. This is also shown in Figure 6-2 where the test environment composes of an empty straight hallway with garbage cans that represent obstacles. Note that the number of obstacles varies between tests as the scene shown represents only the Slalom Test in Section 6.3.3 . The following tests will present results for SECV trajectory, linear/angular velocities and experimentally achieved steering angles based on Ackermann geometry.

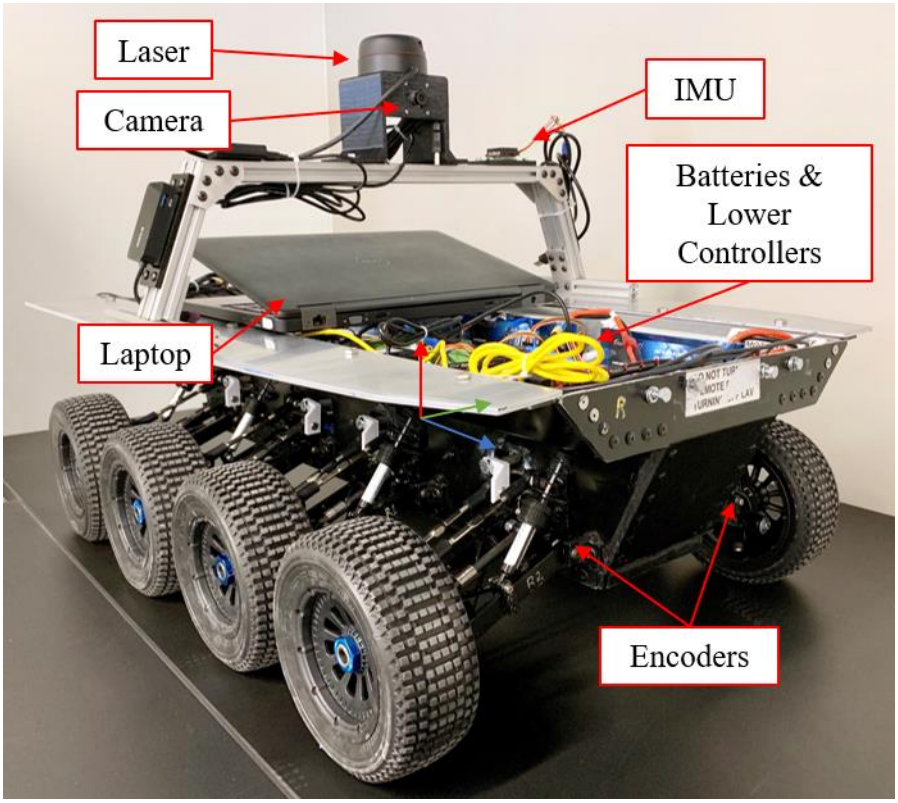

Figure 6-1 Experimental Setup

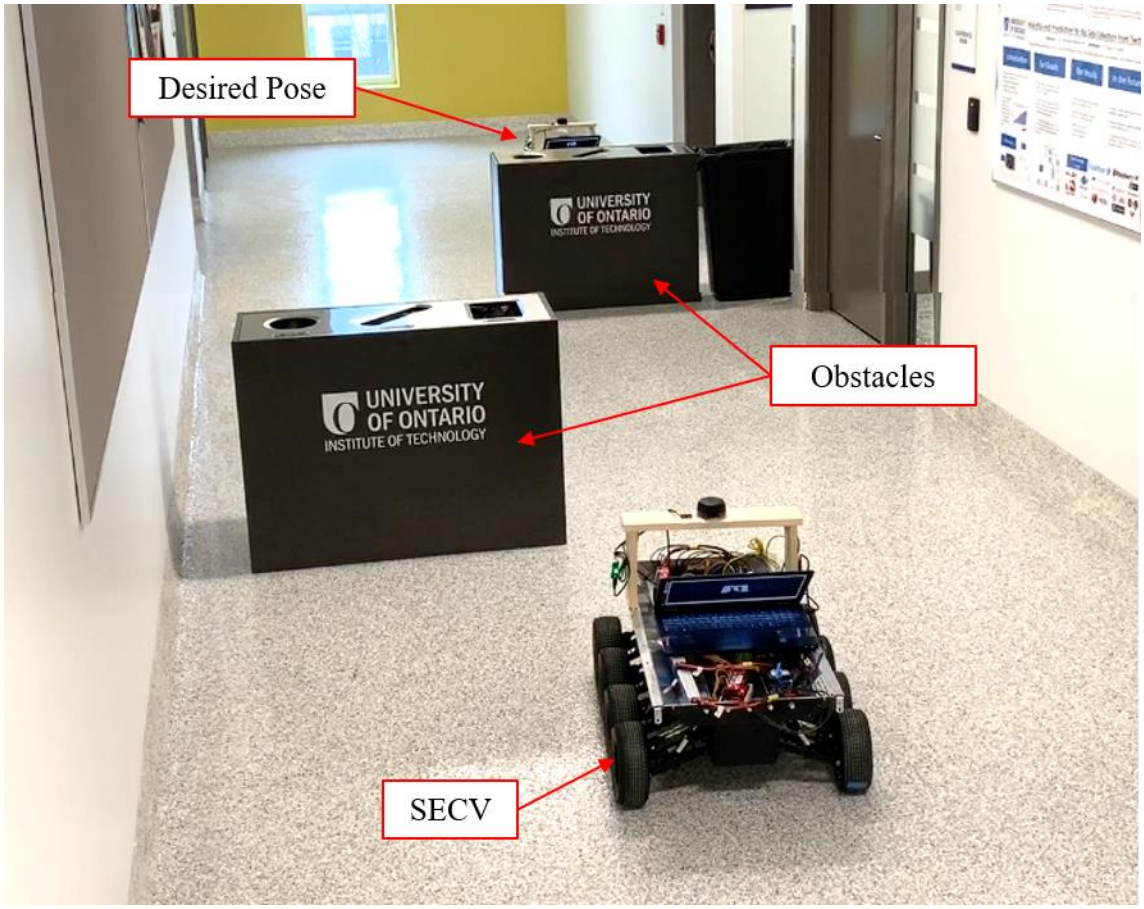

Figure 6-2 Autonomous Navigation Test Setup (Slalom Example)

6.3.2 One Obstacle Test

To start, an obstacle of approximately two meters wide is placed four meters in front of the SECV's initial pose. The desired pose is located behind the obstacle at approximately (9, -0.5). The resultant trajectory generated by the path planners is shown in Figure 6-3 which successfully illustrates the ability of both the Dijkstra and Timed Elastic Band, algorithm at planning paths around the obstacles towards its goal. As the SECV navigates, a cost map is updated with new sensor readings that consequently alters the local plan as the vehicle progresses. This behavior is noticeable when the vehicle is at approximately (3, -1.5) where it realized the obstacle is wider than previously anticipated, as a result, re-

corrected the heading angle to steer farther right. Moving forward, the maximum linear velocity of the SECV is set to 0.3 m/s for this experiment. In Figure 6-4, the desired and actual velocities are shown to reach the set values. Since the desired velocity represents the output of the TEB local planner, the actual velocity data exhibits significantly more noise; however, it is apparent that the vehicle was able to follow the path planner commands as overall trends of desired and actual data resembles each other. At around 11 and 24 seconds, the desired velocity decreased to just below 0.3 m/s as a result of computing hardware limitations. Based on the software differential as mentioned in Section 4.5.2 , the wheels on the left and right side of the vehicle experience different velocities to achieve better cornering maneuverability. This difference in velocity considers not only the linear velocity, but also the angular velocity which can be seen in Figure 6-5. In this figure, the desired and actual angular velocities are shown with more noise present in the latter. Based on the base frame assigned in Section 6.3.1 , a positive angular velocity implies a left turn and vice versa. The SECV began to steer right and away from the obstacle at around the 8 second mark. As the SECV passes the obstacle, it steers left and around behind the obstacle. Towards the end, the SECV steered back right just enough to correct its pose to be parallel with its initial orientation. Both the linear and angular velocities are given to the lower controller to calculate the desired steering angles based on Ackermann geometry. The angles for the front two axles are shown in Figure 6-6. It is important to note that the rear steering angles exhibit the same magnitude but opposite in direction (for spacing and clarity, only the front two axle steering angles are displayed). In the rest of this section, RVIZ (as described in Section 1.3.2.4) screen captures will be attached with snapshots from the physical experiment as seen in Figure 6-7.

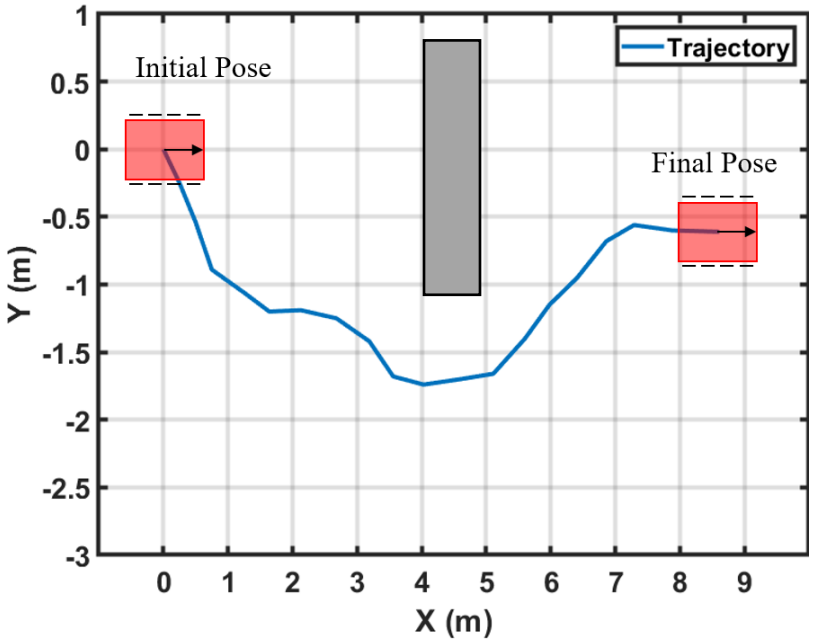

Figure 6-3 One Obstacle Trajectory

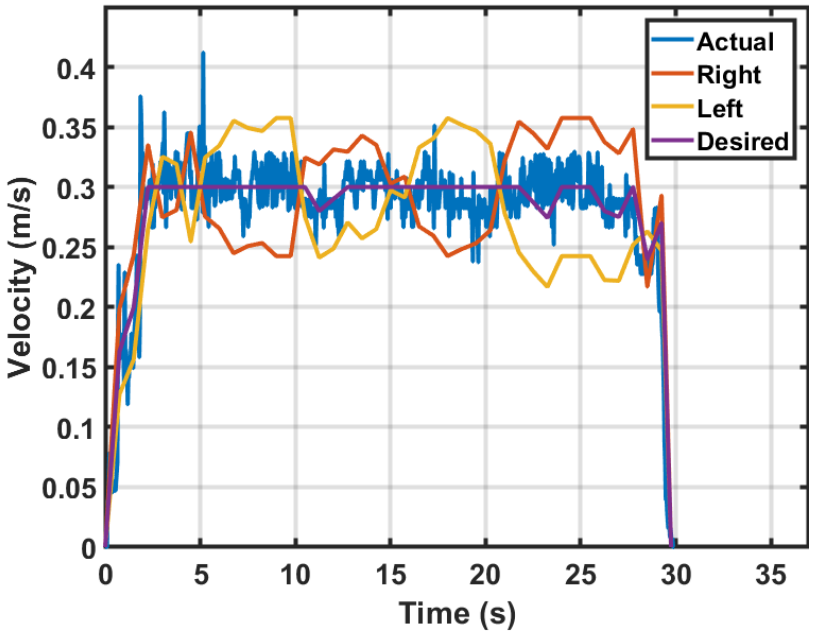

Figure 6-4 One Obstacle Linear Velocity

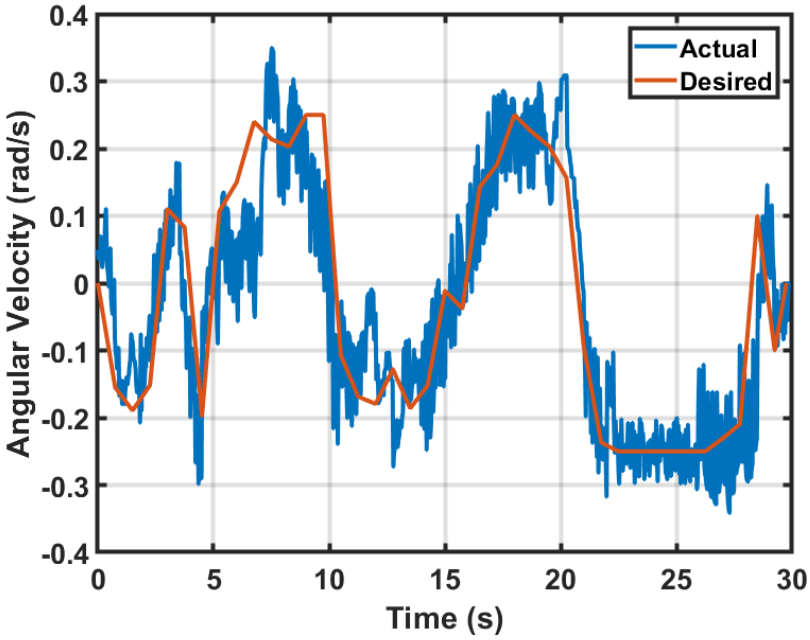

Figure 6-5 One Obstacle Angular Velocity

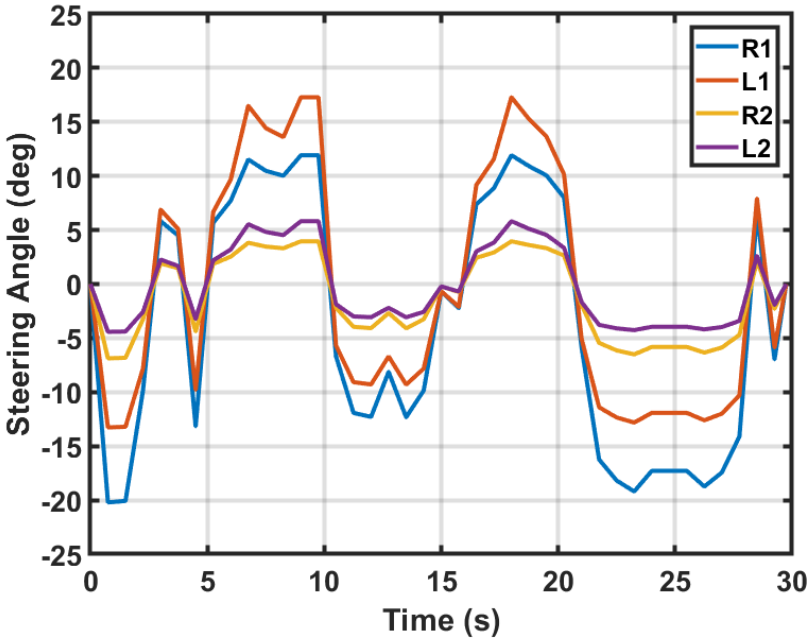

Figure 6-6 One Obstacle Steering Angles

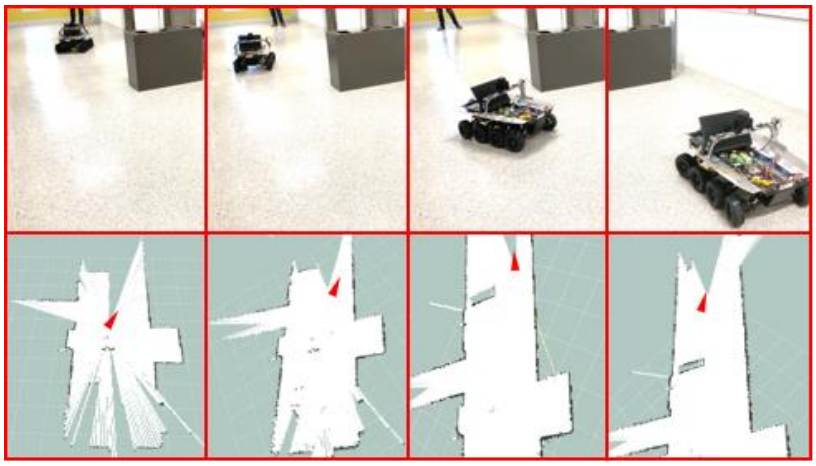

Figure 6-7 One Obstacle: Physical Experiment (top), RVIZ (bottom)

6.3.3 Slalom Test

The Slalom test places two back to back obstacles between the initial and desired SECV pose. Since the second obstacle is placed behind the first; therefore, it is hidden from the sensors' initial view. As a result, the Dijkstra's algorithm will plan a global path based on only the first obstacle. The second obstacle will only be observed as the SECV navigates around the first. The intent here is to test if the vehicle is able to adjust for new sensor data fast enough to alter its initial global path to compensate for the second obstacle. To do this, the vehicle begins at the origin with a goal at approximately $(10, -0.8)$ in the map frame. The experimentally generated trajectory is shown in Figure 6-8 with the desired and actual velocities achieved presented in Figure 6-9. As the previous test, the left and right wheel velocities zig-zag one another as the angular velocities from Figure 6-10 is taken into account. In this figure, the vehicle attempts to clear the first and second obstacle between 0 to 18 seconds and 18 to 28 seconds, respectively. With the previously assigned convention, it is easy to see that the differential speed from Figure 6-9 matches accordingly as outer wheels always exhibit a higher velocity than inner due to turning radius difference.

In Figure 6-11, the steering angles of the front two axles are illustrated. From this figure, the maximum steering angle reached is 35 degrees. When looking closer at Figure 6-11 around 25 seconds, the vehicle is attempting to steer right after clearing the second obstacle. During this, the first axle right wheel exhibits the highest turning angle, followed by the left wheel of the same axle and then the right and left wheels of the second axle, accordingly. This relationship represents the Ackermann geometry that was discussed in Section 4.5.1 . Figure 6-12 shows consecutive images of the slalom experiment with the top row displaying the physical SECV and the bottom row displaying the accompanying laser data visualization. This test shows the overall navigation ability of the vehicle to clear back to back obstacles by steering both to the left and right and arriving at the desired destination.

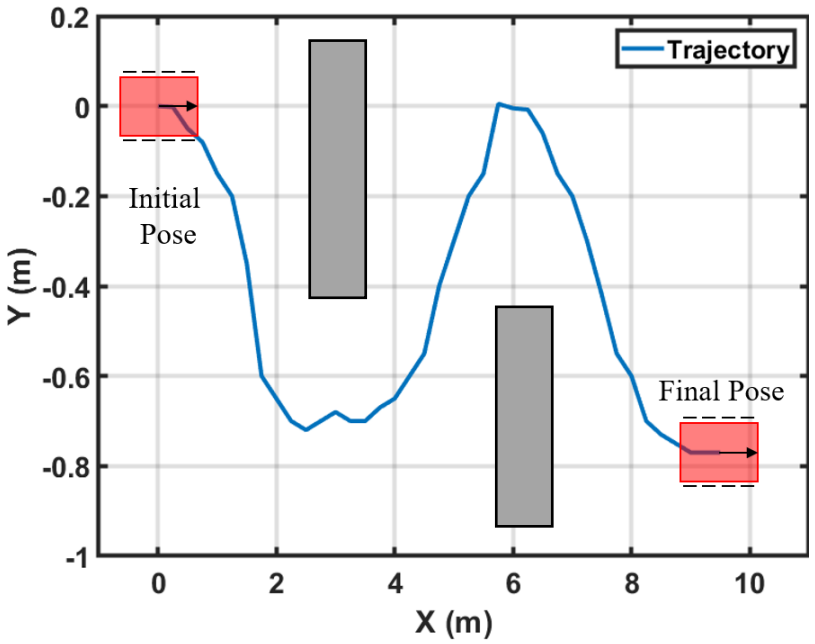

Figure 6-8 Slalom Trajectory

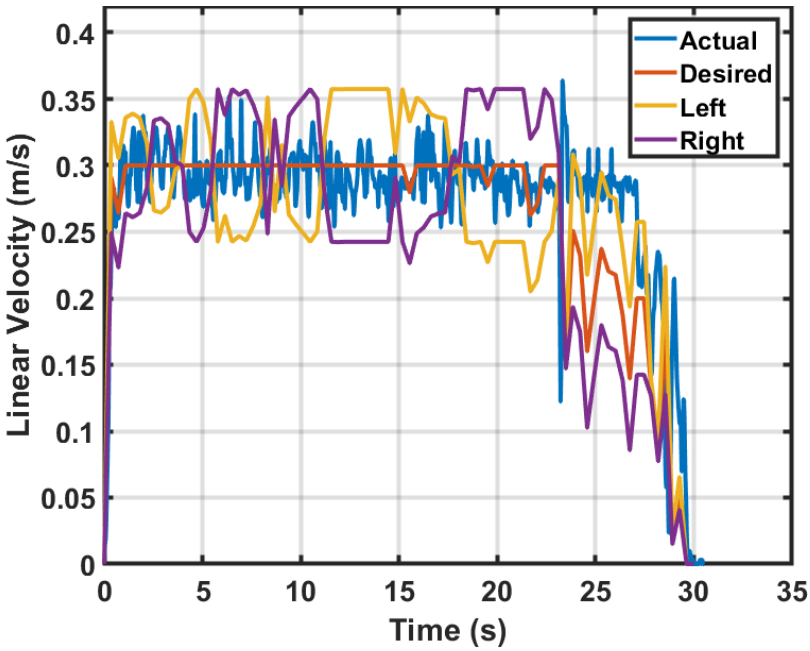

Figure 6-9 Slalom Linear Velocity

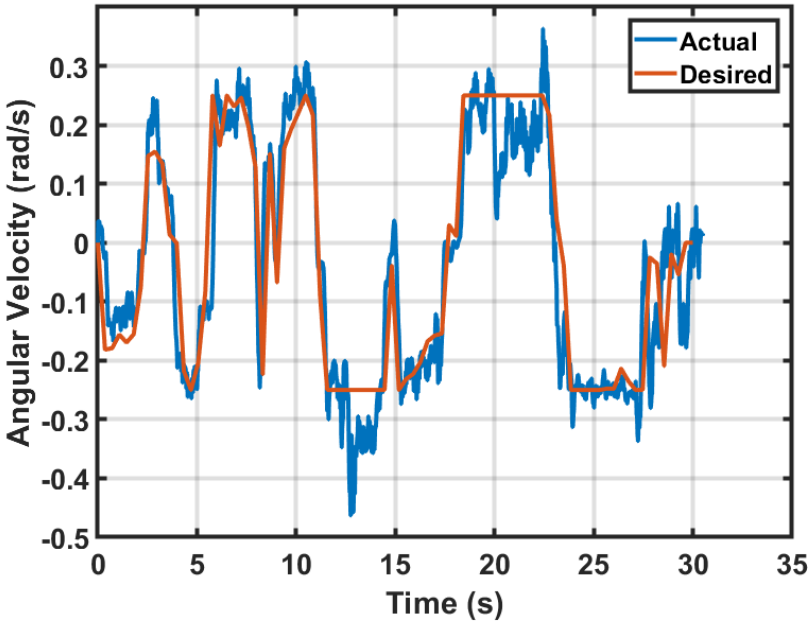

Figure 6-10 Slalom Angular Velocity

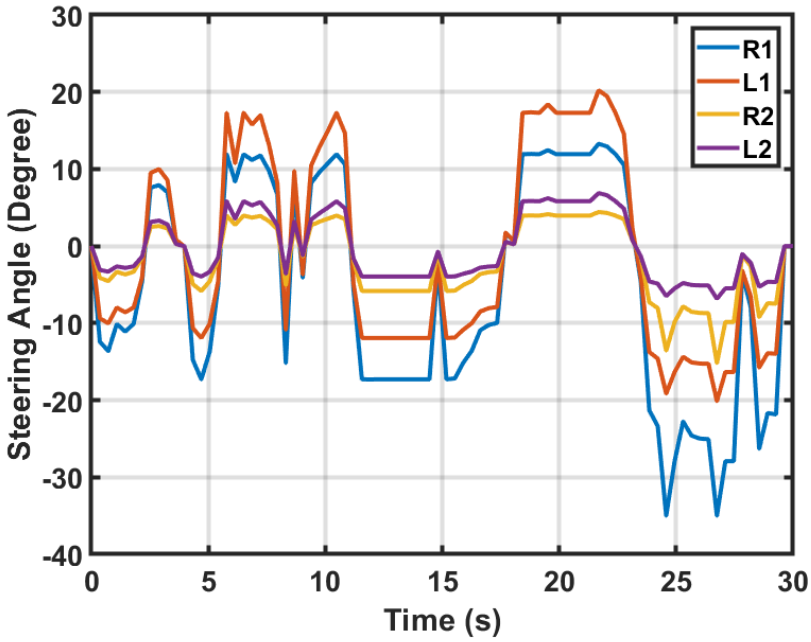

Figure 6-11 Slalom Steering Angles

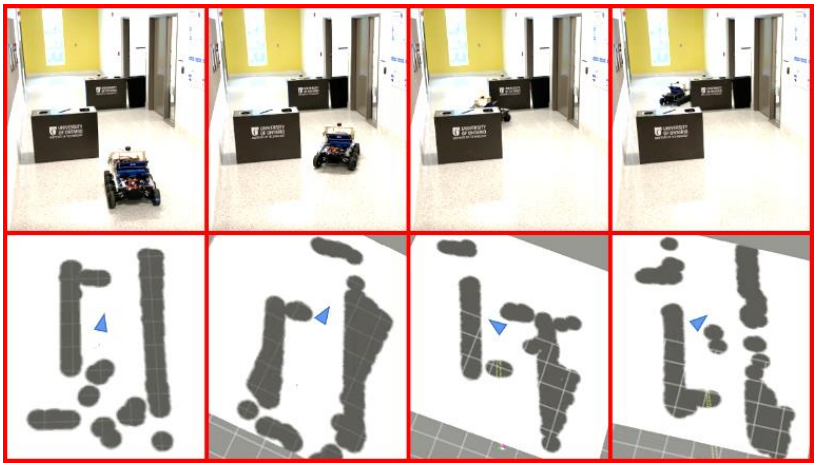

Figure 6-12 Slalom: Physical Experiment (top), RVIZ (bottom)

6.3.4 Parking Test

The Parking Test is intended to evaluate the SECV's ability to avoid obstacles in tight spaces. As shown by Figure 6-13, the SECV starts at the origin and is given a goal of (3.5, 1.5) where a parking spot of approximately 1.5 meters wide is located. This parking

spot requires the vehicle to perform a total orientation displacement of 90 degrees to the left from its initial heading angle to enter without colliding with the walls on the right. In the implemented algorithm, the vehicle is aware of its own size and minimum achievable turning radius. When planning an appropriate path, all obstacles detected are inflated with a pre-determined radius to create a cost map. With these things in mind, the vehicle decided to first steer towards the right to create enough space for the sharp left turn due to the turning radius constraint. Figure 6-13 illustrates the trajectory that was taken which took approximately 27 seconds to complete. Like the previous experiment, the maximum linear and angular velocities are also set to 0.3 m/s and 0.25 rad/s, respectively. In Figure 6-14, the linear velocity reached the maximum velocity throughout its course until the end where it begins to correct its pose. The differential velocities of the left and right wheels of vehicle are generated based on the angular velocities shown in Figure 6-15. From this figure, the first 5 seconds illustrates the vehicles attempt to steer right around the first obstacle. For the next 20 seconds that follows, the vehicle tries to steer left into the parking spot. At around 16 and 24 seconds, the change in angular velocities shows the SECV's attempt to correct its pose. The angular velocities' effect is also apparent in Figure 6-16 which shows the steering angles of the front two axles. In this experiment, the maximum steering angle reached is approximately 28 degrees with most of the time spent around 18 degrees for the first axle inner wheel. Figure 6-17 shows consecutive images of the physical experiment and the laser data visualization in the top and bottom row, respectively. From this experiment, the SECV was able to successfully park itself. In addition, the results shown here also validates the proposed localization algorithm as it was able to keep up with the large change in orientation.

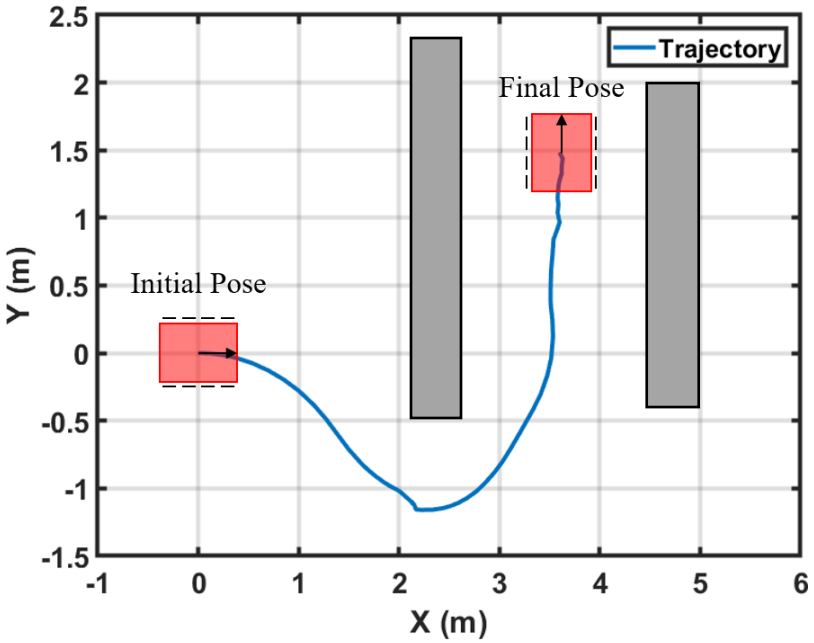

Figure 6-13 Parking Trajectory

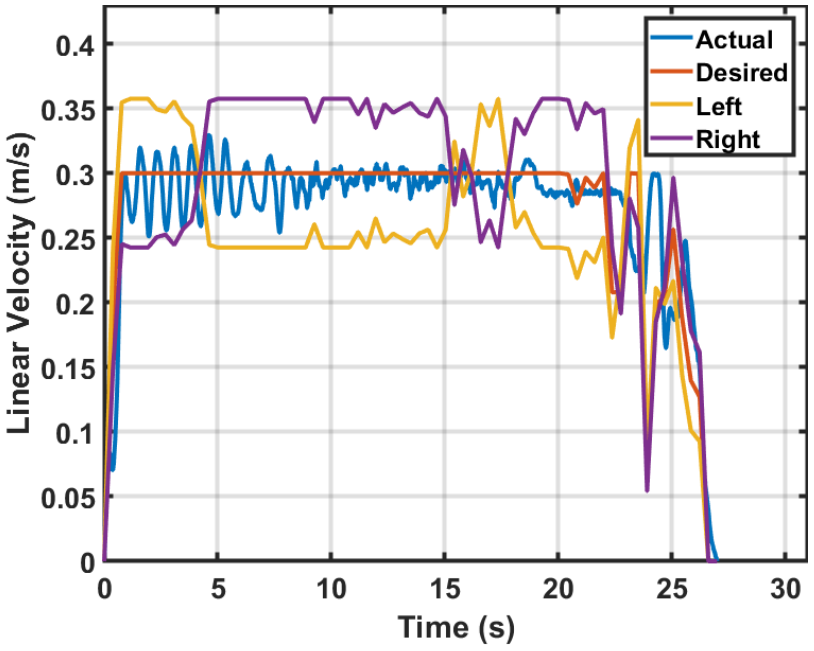

Figure 6-14 Parking Linear Velocity

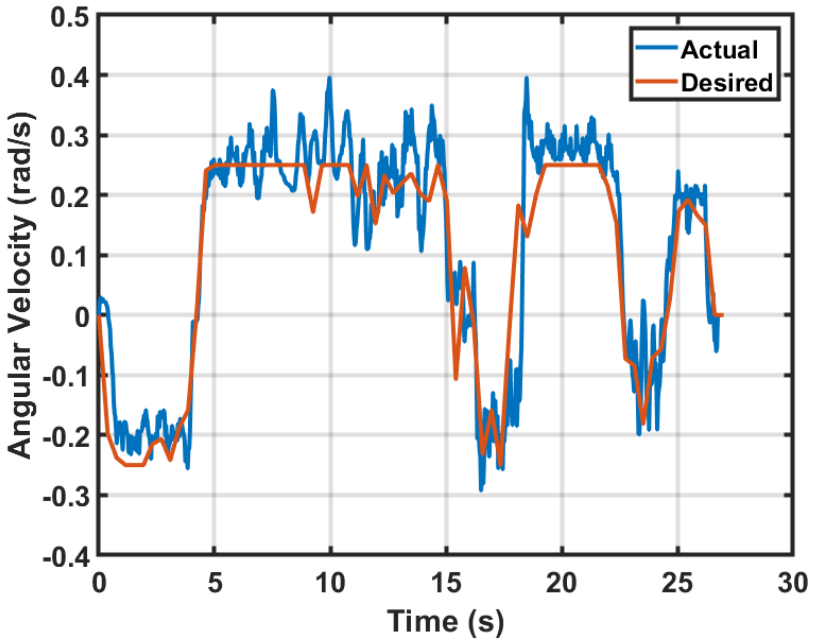

Figure 6-15 Parking Angular Velocity

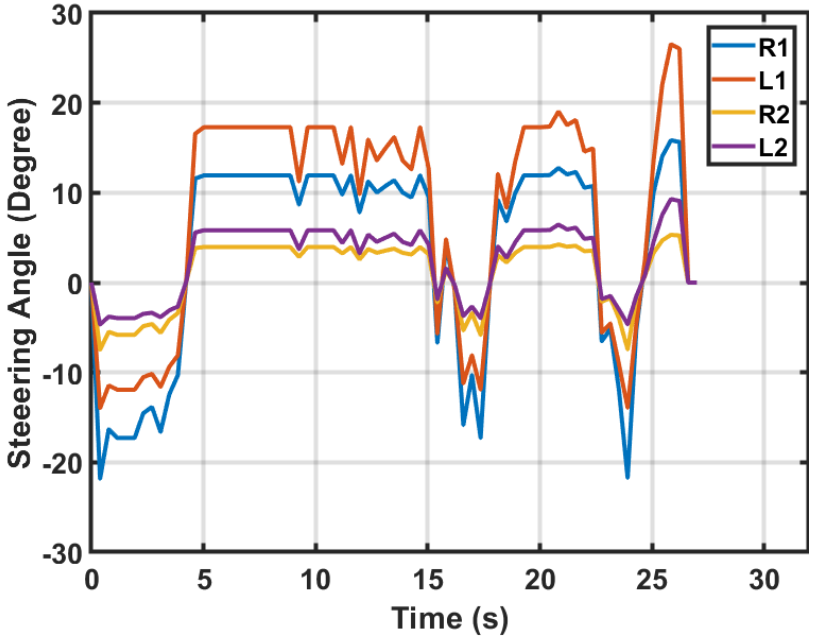

Figure 6-16 Parking Steering Angles

Figure 6-17 Parking: Physical Experiment (top), RVIZ (bottom)

6.3.5 Odometry Drift Test

Sensor drift is a result of error accumulation over a long distance and/or time of navigation. As a result of this drift, navigation becomes inaccurate as the SECV lose track of its position within the map. The following test studies the effects of the odometry sensor drift by sending the SECV over different distances to a common desired pose. More specifically, this test is conducted with three initial regions that are located throughout the hallway as seen in Figure 6-18. Within each region, three different positions and orientations are chosen as starting poses. From there, all poses are sent to the desired pose via the Dijkstra's and TEB algorithm with AWS configuration. To ensure proper reset of error, sensor power is cycled between tests. Table 6-8 tabulates the different starting poses as well as the final achieved poses with their respective average position and orientation error. From these results, it is evident that Region 1 exhibit the highest amount of drift, leading to the largest deviation of 2.06 meters and 14.67 degrees between achieved and desired. Region 3 achieves a more accurate result with the deviation of 0.98 meters and 10.67 degrees. The final positions are plotted in Figure 6-19 which shows the tolerance

window that is calculated by taking the average error of all final positions. The result is a circle with a 1.5-meter radius.

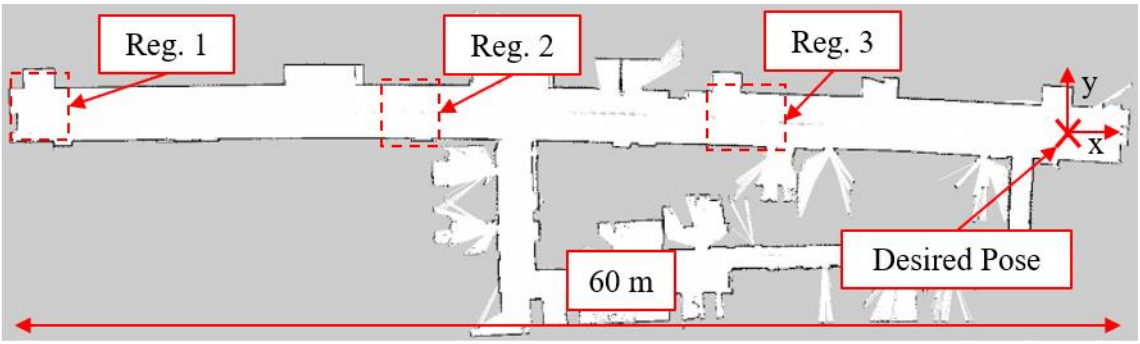

Figure 6-18 Test Environment

Table 6-8 Odometry Drift Results

	Starting Pose (x, y, θ)	**Final Pose (x_f, y_f, θ_f)**	**Avg. Pos. Error**	**Avg. Ori. Error**
Reg. 1 – Pos. 1	(-60, 0, 0)	(1.85, -1.1, 15)	2.06 meter	14.67 deg.
Reg. 1 – Pos. 2	(-58, 1, -10)	(-1.5, -1.4, -12)		
Reg. 1 – Pos. 3	(55, 1.2, 5)	(-1.4, 1.4, 17)		
Reg. 2 – Pos. 1	(40, -1, 15)	(1.63, 1.2, 14)	1.57 meter	12.33 deg.
Reg. 2 – Pos. 2	(42, 0.5, 0)	(-1.3, 0.5, 11)		
Reg. 2 – Pos. 3	(44, 1, -10)	(-0.9, -0.95, -12)		
Reg. 3 – Pos. 1	(27, 1, -15)	(0.8, -0.75, 9)	0.98 meter	10.67 deg.
Reg. 3 – Pos. 2	(26, -0.5, 17)	(-0.7, 0.8, 12)		
Reg. 3 – Pos. 3	(23, -1.4, 5)	(0.6, 0.5, -11)		

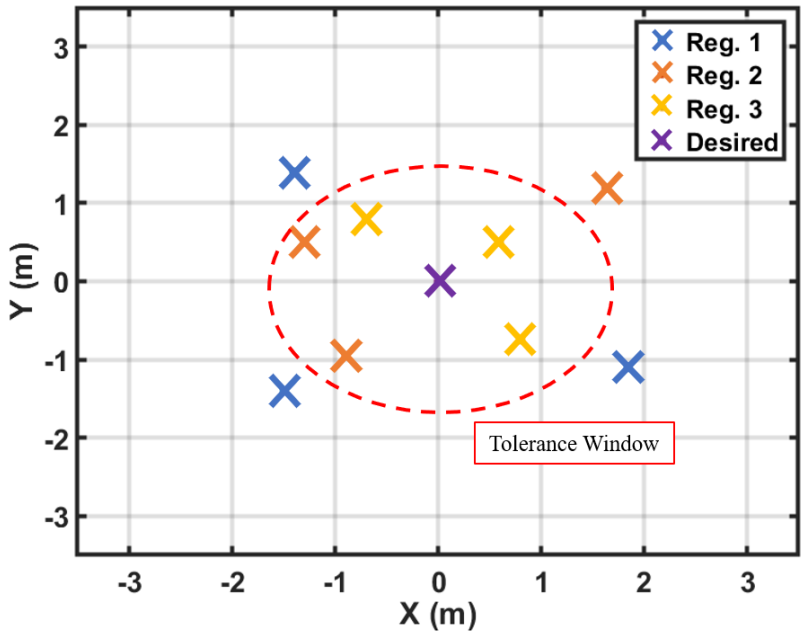

Figure 6-19 Odometry Drift Test: Final Positions

6.4 M-PBVS Experiments

Based on the results obtained from the odometry drift test, it is evident that the M-PBVS algorithm must be able to bring the SECV's pose to within centimeters from an average tolerance window of 1.5 meters. The following section evaluates the M-PBVS algorithm proposed in Chapter 5 by assessing its performance in terms of close quarters pose correction. Since the accuracy of the proposed algorithm depends highly on the pose estimation algorithm employed, the first test in this section evaluates its performance attributes. After the pose estimation algorithm is evaluated, two initial scenarios are chosen within the tolerance window to evaluate the pose correction performance. Starting with the experimental setup, all tests conducted in this section are performed in an indoor lab environment with smooth surfaces. A desired pose is chosen in front of a visual landmark at (0, 0) as shown in Figure 6-20. From the desired pose, four quadrants are identified based

on a cartesian coordinate system similar to what was shown in Figure 5-9. As mentioned, the two tests conducted are to evaluate both scenarios of M-PBVS which are Direct Goal and Alternate Goal. The intent is to assess the algorithm's ability to correct the SECV's pose to match that of the desired based on an external visual landmark acquired from [181]. The heading angle, ϑ, in the following experiment is measured relative to the longitudinal axis of the SECV where a counter clock wise rotation is deemed positive. Experimental data such as trajectory, linear/angular velocity as well as position and orientation errors are presented.

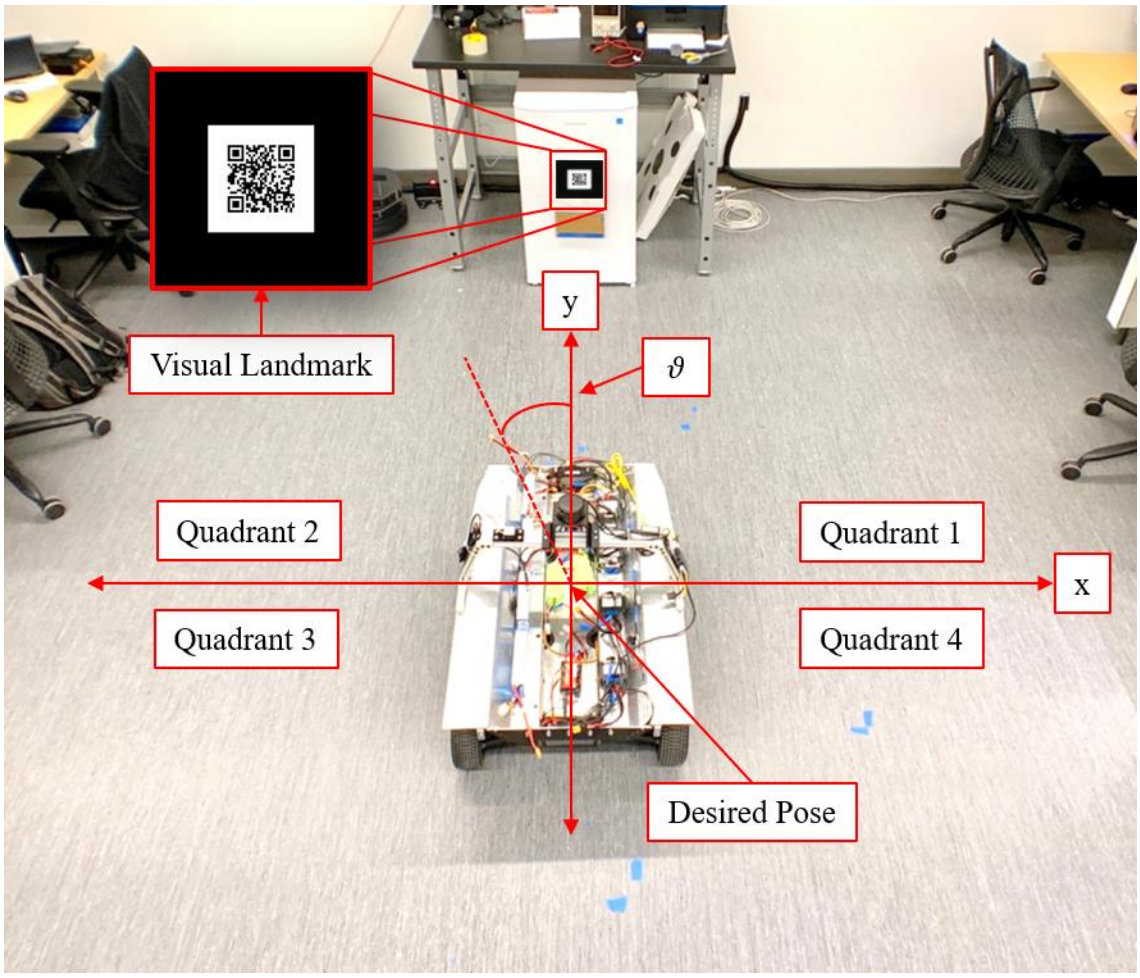

Figure 6-20 M-PBVS Experimental Setup

6.4.1 Pose Estimation Test

As mentioned, this section evaluates the pose estimation algorithm employed in this work which is developed by [180]. To assess the accuracy of the algorithm, four different locations are measured and estimated with a ruler and the estimation algorithm, respectively. The results from both are tabulated and plotted below with an average position error of 3.5 centimeters.

Table 6-9 Pose Estimation Test Results

	Theoretical	Experimental	Error (cm)
Position 1	(-1, -0.5)	(-1.0336, -0.4895)	3.52
Position 2	(0, 1)	(0.0433, 1.0216)	4.84
Position 3	(0, -1)	(0.0201, -0.9752)	3.19
Position 4	(1, 0.5)	(0.9882, -0.5221)	2.51

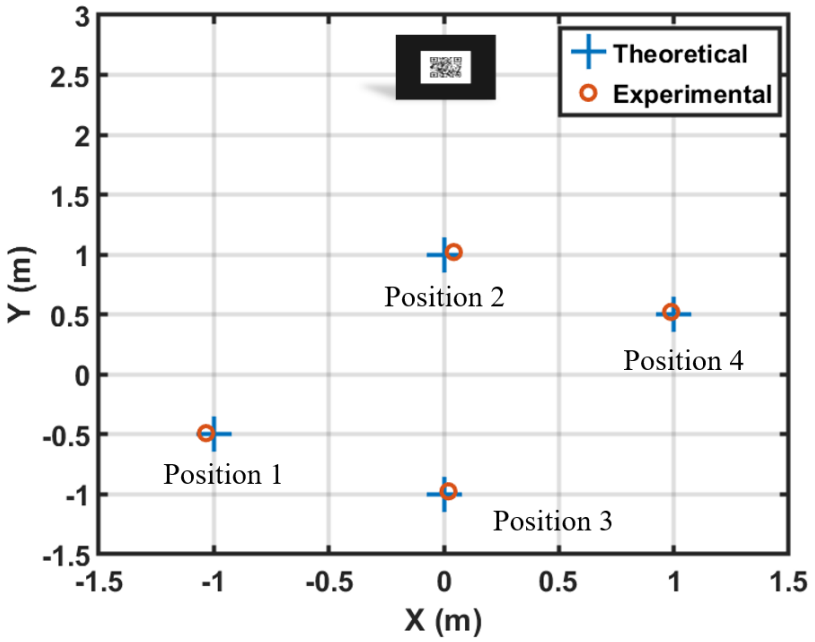

Figure 6-21 Pose Estimation Test Results

6.4.2 Direct Goal Test

In the Direct Goal test, position and orientation error are introduced by placing the SECV at position (0.15, 0.9) with a -0.2 rad heading angle in Quadrant 1. As mentioned before, the desired position is the origin of the cartesian plane with a 0 rad heading angle. In Figure 6-22, the initial and final pose of the SECV is illustrated on top of the trajectory that is generated by the proposed M-PBVS algorithm as the result of both steering and velocity control. Starting with the first stage, an angular velocity is generated to reduce the heading error by using the Diamond Steer configuration. This velocity is evident during the first 4 seconds of the test where it reached a maximum of 0.15 rad/s, pivoting the SECV clockwise about its center as seen in Figure 6-23. Because of this, the orientation error is reduced to approximately zero as shown in Figure 6-24 while both the linear velocity and position error remains unchanged. Once the orientation is corrected, the SECV enters the second stage of M-PBVS which first determines whether it is in a direct goal or alternate goal scenario. Based on the calculation from equation 5-24, the initial position qualifies as a direct goal scenario as the initial $\delta_{approach}$ is approximately 10 degrees which is less than the δmax. Consequently, the generated steering angles are shown by Figure 6-25 which remains between 8 to 10 degrees in a Synchronous Steer configuration until it arrives at the intended goal. In parallel with the steering control, the linear velocity reached a maximum of 0.2 m/s before slowly reducing to zero over the next 20 seconds. During the second stage, the angular velocity is zero while the linear velocity is negative because the desired pose is set to behind the initial as shown in Figure 6-23. Snapshots of the physical experiment along with the camera's field of view is shown in Figure 6-26. The result from this test successfully illustrates the M-PBVS's ability to correct SECV's initial pose to

centimeter accuracy as the final achieved position is (0.01, -0.04) which yields an error percentage of 5.6%.

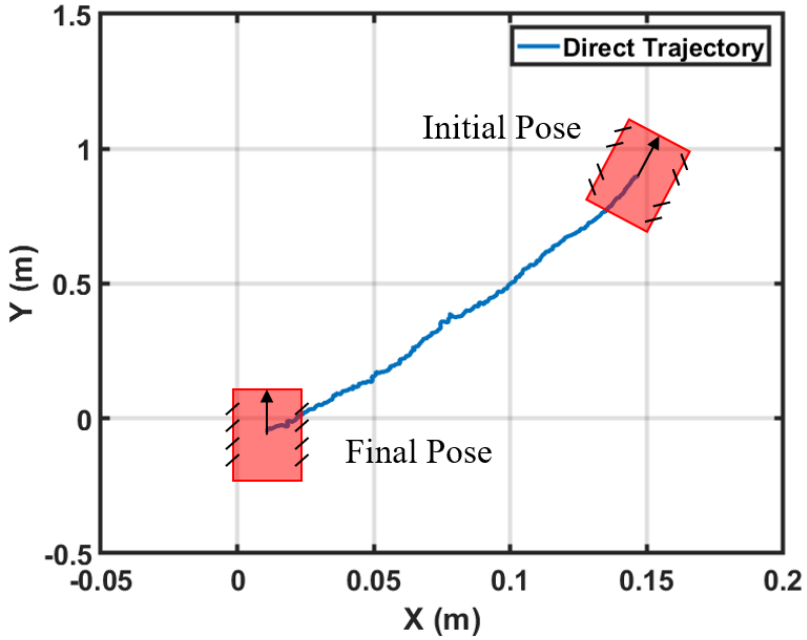

Figure 6-22 Direct Goal Trajectory

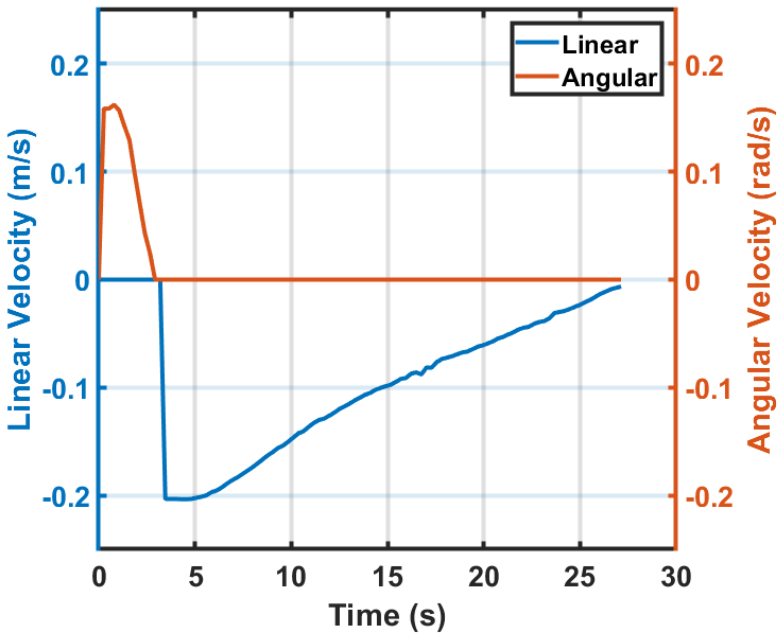

Figure 6-23 Direct Goal Velocity

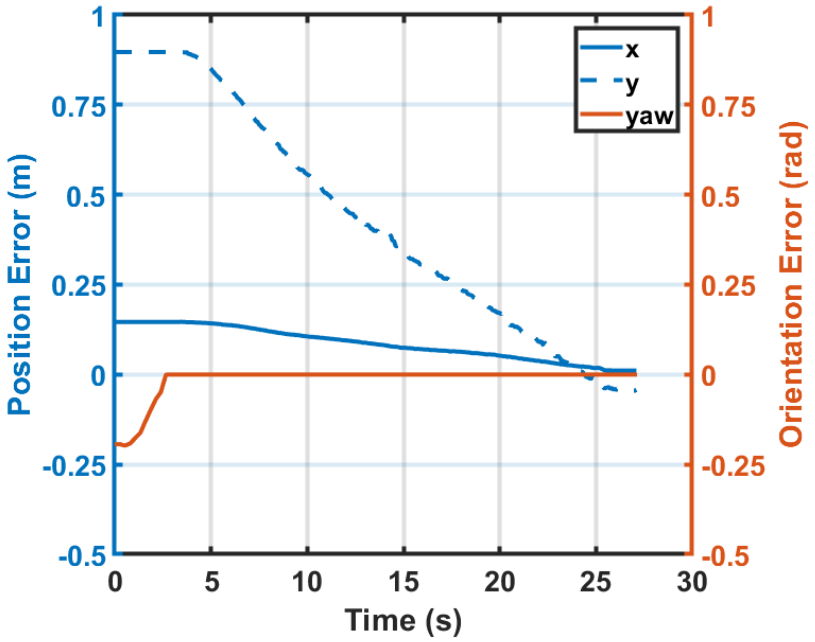

Figure 6-24 Direct Goal Error

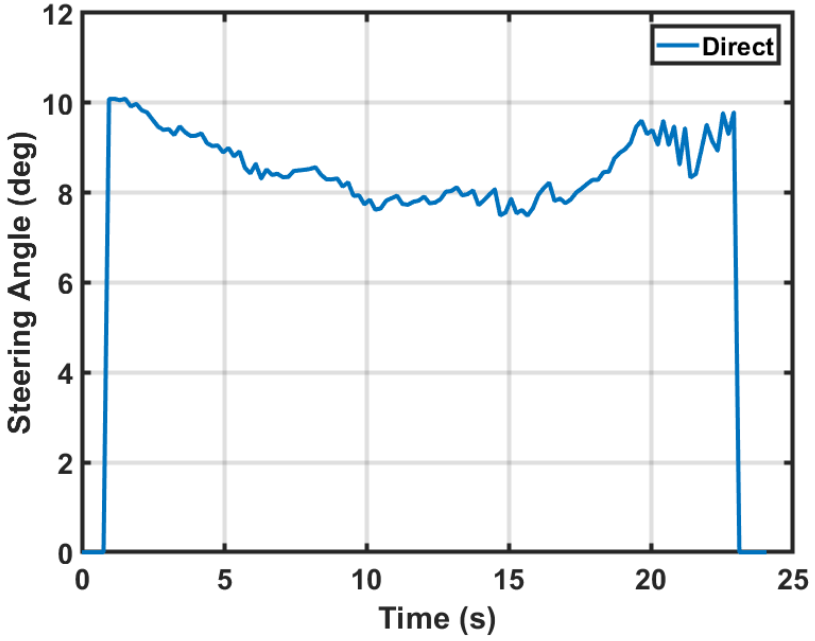

Figure 6-25 Direct Goal Steering Angle

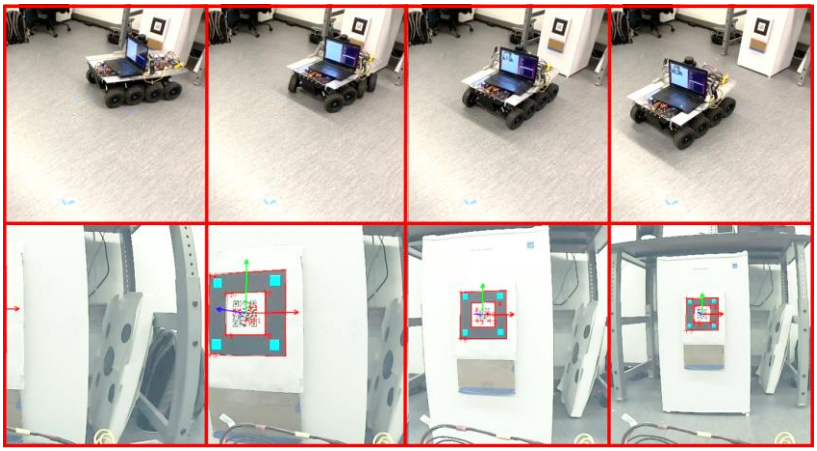

Figure 6-26 Direct Goal: Physical Experiment (top), Camera View (bottom)

6.4.3 Alternate Goal Test

After successful completion of the Direct Goal test, this section presents experimental results of the M-PBVS's ability to solve an alternate goal scenario to correct the SECV's initial pose. The test begins with the SECV at (-0.36, 0.59) with a heading angle of 0.25 rad in Quadrant 2 as shown in Figure 6-27. Like the previous test, the first stage of M-PBVS corrects the orientation by generating a negative angular velocity for the Diamond Steer configuration to pivot the vehicle clockwise. In this case, the angular velocity reached a maximum of -0.25 rad/s while the yaw error is also corrected within the first 4 seconds as shown in Figure 6-28 and Figure 6-29, respectively. After stage one corrects the orientation error, the M-PBVS algorithm recognizes that the current SECV position requires an alternate goal as the initial $\delta_{approach}$ is 32 degrees which exceeds the δmax. As a result, an alternate goal is calculated to be (-0.24, 0.87) by finding the intersection of two lines as demonstrated in Section 5.3.2.2 . The outcome is a calculated steering angle of approximately 25 degrees to reach the alternate goal. The experimental results show that the alternate goal is reached at approximately 10 seconds as seen in Figure

6-29 where the error in the y axis increases to approximately 0.9. This is an increase from 0.6 at the beginning of stage two because the vehicle is moving away from the desired goal at a maximum velocity of 0.21 m/s. Once arrived at the alternate goal, the new $\delta_{approach}$ is calculated to be approximately 17-degree angle which is achievable when compared to the initial $\delta_{approach}$ angle. The experimentally achieved steering angles are shown in Figure 6-30. Like the previous test, the desired pose is also set behind the initial; therefore, the linear velocity is negative after reaching the alternate goal as it reverses towards the desired pose with Synchronous Steering. From here, the position error begins to decrease over the next 17 seconds as shown in Figure 6-29 where the SECV eventually reach a final position of (-0.008, 0.0009). It is worth mentioning that this final position yields a 1.1% error. Consecutive images of the physical experiment and camera view is presented in Figure 6-31.

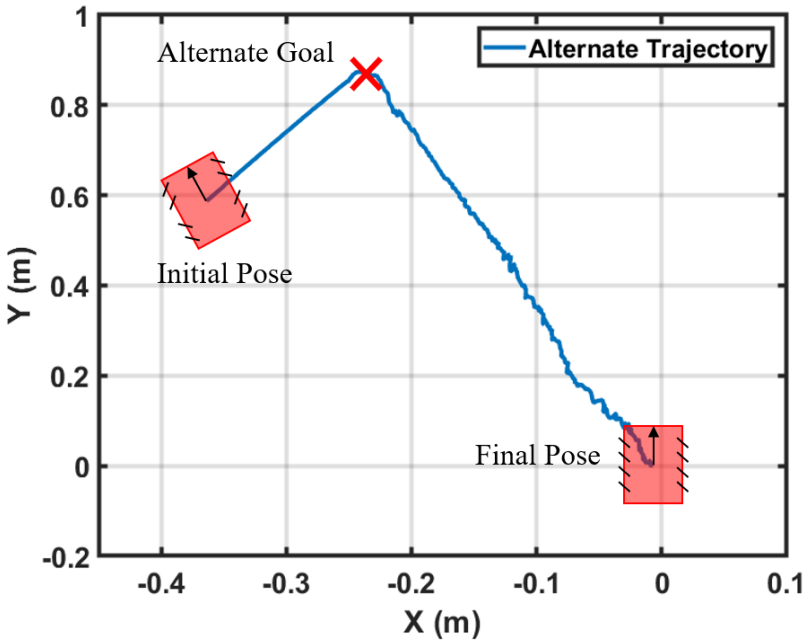

Figure 6-27 Alternate Goal Trajectory

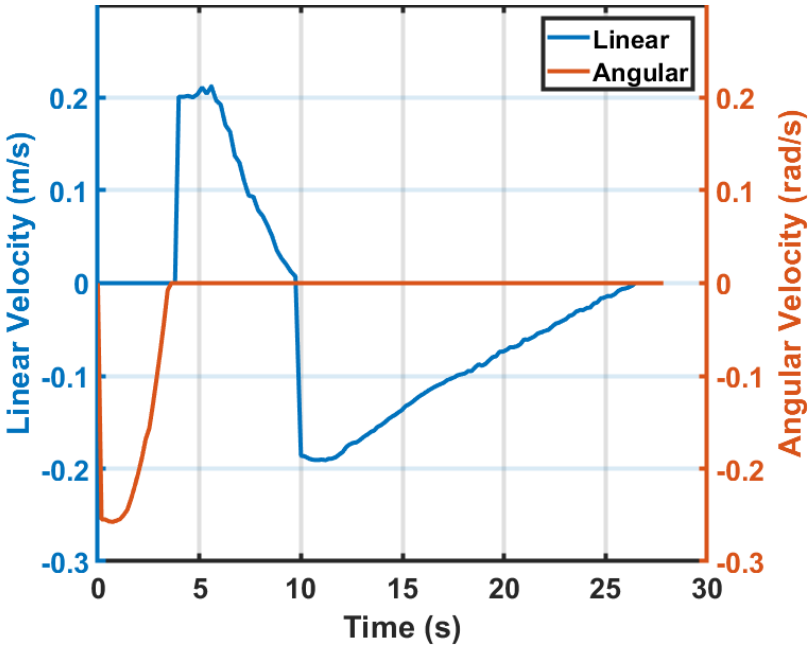

Figure 6-28 Alternate Goal Velocity

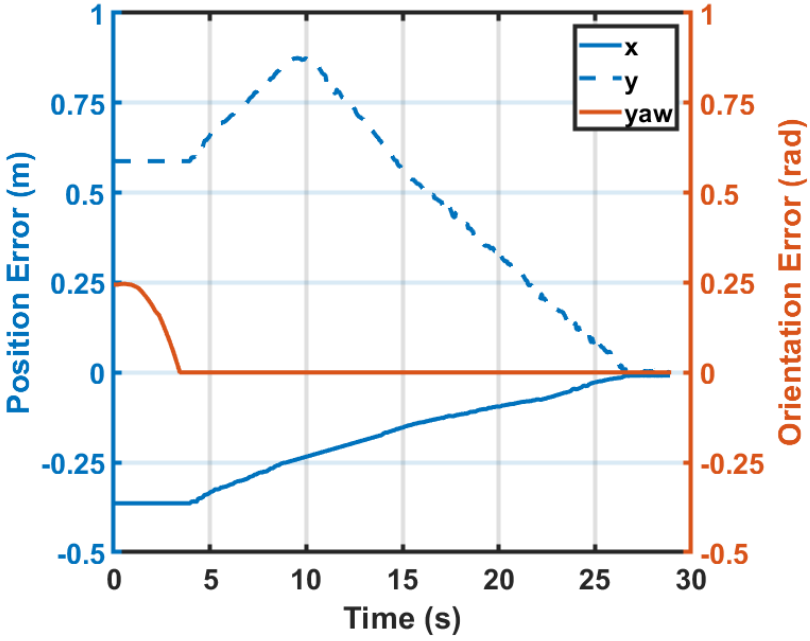

Figure 6-29 Alternate Goal Error

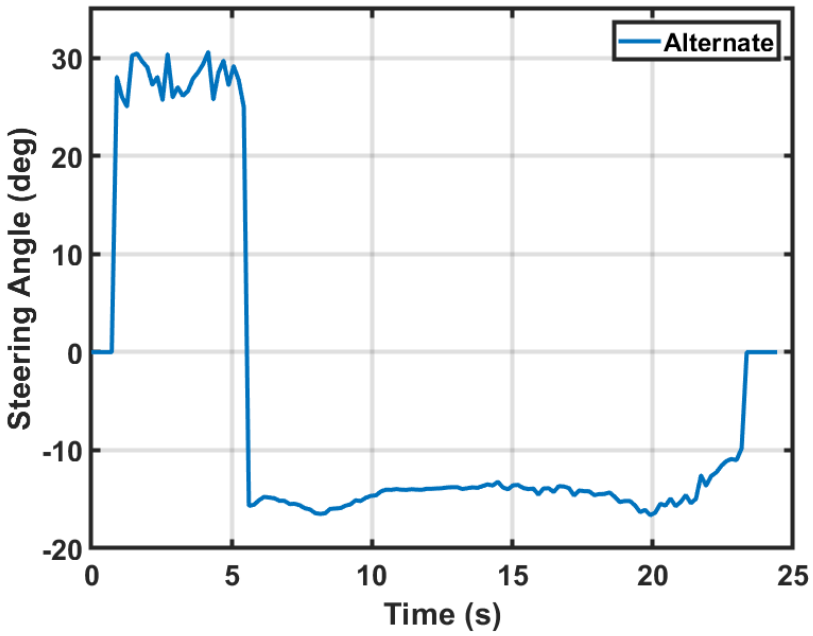

Figure 6-30 Alternate Goal Steering Angle

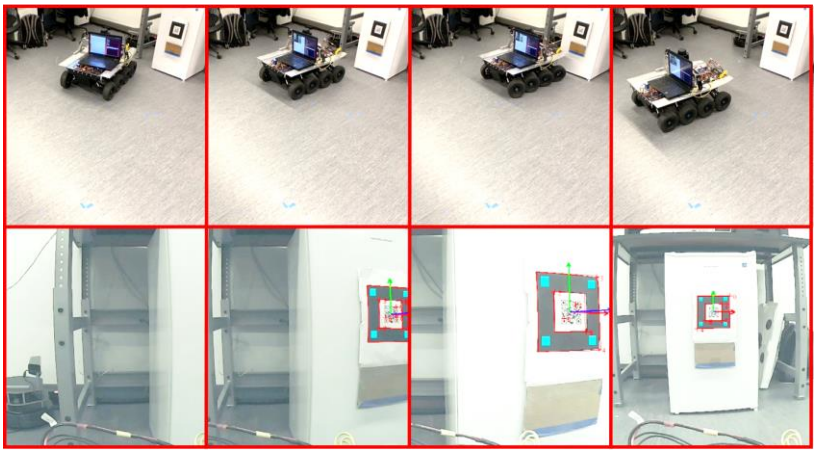

Figure 6-31 Alternate Goal: Physical Experiment (top), Camera View (bottom)

6.4.4 Experiment Discussion

It is important to note that the desired slope and current slope are set to 20 degrees and 25 degrees, respectively, for the physical experiment which is different than what was described in Section 5.3.2 . The reason for this is because using the maximum steering

angle would give the SECV no room for correction. For example, in an ideal world, the SECV can travel at the desired angled paths perfectly without deviation; however, this is not possible in real life due to imperfections within the mechanical systems such as non-perfect wheel alignments. To combat this, the current slope values are intentionally set to higher values than the desired slope, so that the SECV is guaranteed to arrive at an alternate goal that creates an achievable direct goal. The results from the two tests presented in this section show that the proposed M-PBVS algorithm works and can achieve centimeter accuracy with an average error percentage of 3.35%.

6.5 TEB vs M-PBVS Experiments

The results from Section 6.3 showed that the proposed SECV can achieve point to point navigation while avoiding collisions; however, the drift test from Section 6.3.5 revealed an average position error of 1.5 meters when traveling up to 60 meters. This average error is extremely high; as a result, the full docking algorithm proposed in Section 5.4 is evaluated here. More specifically, this section assesses the difference between pose correction using the TEB + AWS method and the M-PBVS approach. Similar to the previous section, two tests are performed to incorporate both Direct Goal and Alternate Goal scenarios. Additionally, the two scenarios are chosen to be within the tolerance window while residing in Quadrant 3 and 4 to differentiate from the results shown in the previous section. Experimental results regarding robot trajectory, pose error, total distance traveled, and arrival accuracy is presented and discussed here.

6.5.1 Pose Correction with Direct Goal Comparison

From the results of Section 6.3.5 , the average position error is represented by a circle with a radius of 1.5 meters. To achieve docking, this error must be brought down significantly to within 10 cm; therefore, close quarters pose correction is necessary. To do this, the proposed M-PBVS algorithm is compared with the TEB planner to study the difference in performance. Starting with a Direct Goal scenario, the SECV is placed at position (-0.25, -1.47) in Quadrant 3 with an orientation of 0.28 rad to simulate the final position of the SECV after traveling from a far initial pose with the intent to arrive at the origin. From here, the SECV is first equipped with AWS and TEB to adjust its own pose by resending the desired pose to the path planner. This test is repeated with the M-PBVS algorithm from the same position and orientation. The resultant trajectories of both algorithms are shown in Figure 6-32. Due to the minimum turning radius nature of AWS, the TEB trajectory shows a series of changes in direction in order to correct its pose. On the other hand, the M-PBVS algorithm is able to utilize its two-stage process in combination with alternative steering modes to arrive at the desired pose in a more efficient and precise manner. When looking at the total distances, the TEB planner traveled 6.10 meters compared to the 1.44 meters by the M-PBVS algorithm. This characteristic is also apparent when looking at the position errors in Figure 6-33 which shows the M-PBVS algorithm able to converge in nearly half the time of the TEB planner. Furthermore, the TEB planner also sees its error oscillate while the M-PBVS algorithm sees its error steadily decrease. Not to mention, the M-PBVS algorithm is able to arrive much closer to the desired as seen by the final achieved position of (-0.017, -0.07), while the TEB achieved a (0.12, 0.17). Based on the position results, the M-PBVS algorithm yields a 5.78% error

while the TEB yields a 21.63% error. In addition, the final achieved orientation of the TEB planner is -0.26 rad which is also higher than M-PBVS's 0.003 rad as seen in Figure 6-34. It is worth mentioning that this test is conducted after cycling power to all the sensors to eliminate previously acquired error. Regardless, the TEB planner is unable to arrive as precisely as the proposed M-PBVS algorithm even without significant sensor drift. The following table tabulates the performance difference between the two algorithms during this test.

Table 6-10 Pose Correction Comparison with Direct Goal

	Final Pose	Pos. Error %	Ori. Error %	Total Distance
TEB	(0.12, 0.17, -0.26)	21.63%	92.86%	6.10 m
M-PBVS	(-0.017, -0.07, 0.003)	5.78%	1.07%	1.44 m

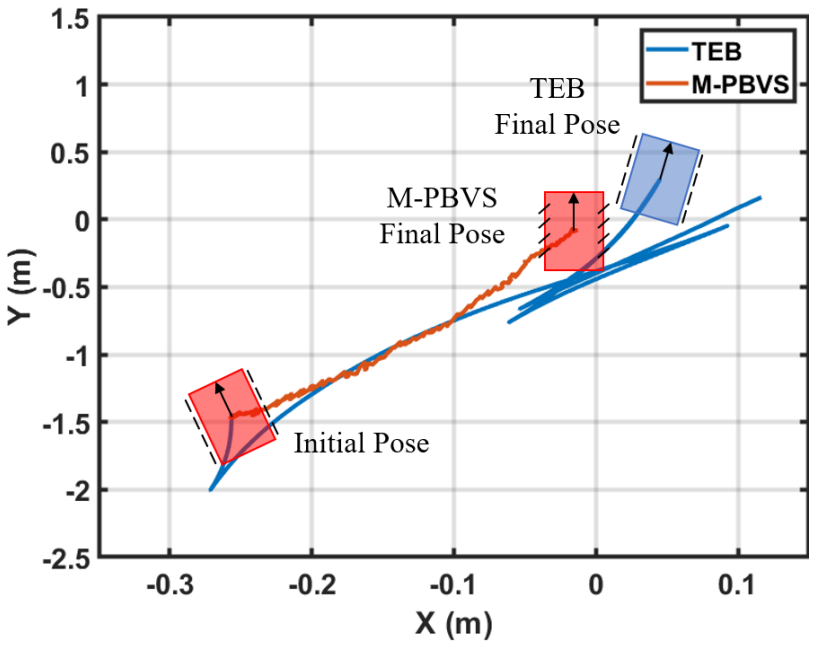

Figure 6-32 TEB vs M-PBVS Direct Goal Trajectory

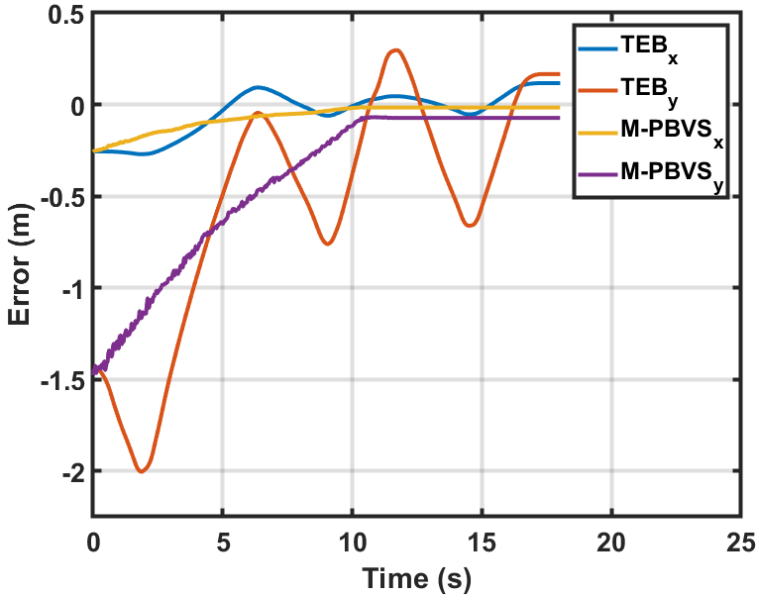

Figure 6-33 TEB vs M-PBVS Direct Goal Position Error

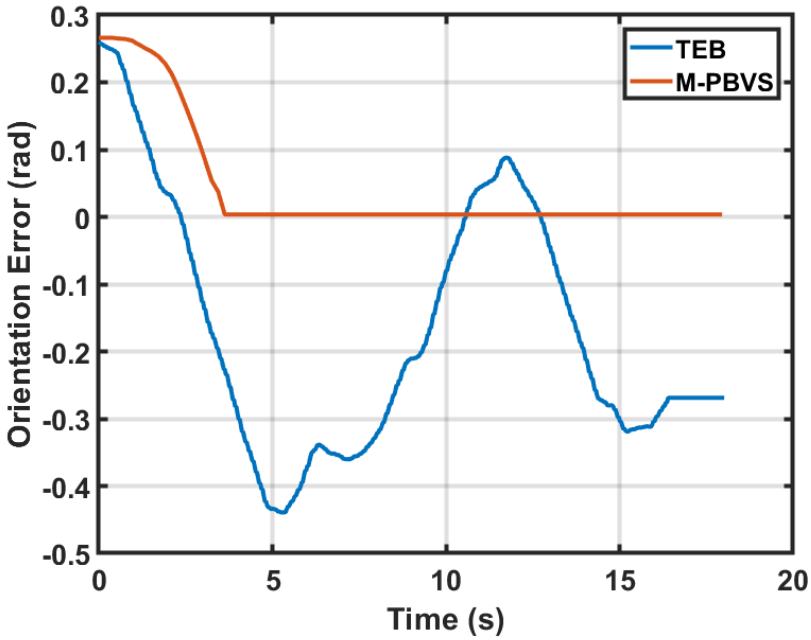

Figure 6-34 TEB vs M-PBVS Direct Goal Orientation Error

6.5.2 Pose Correction with Alternate Goal Comparison

To gain a better insight at the performance difference between TEB and M-PBVS, another initial pose is selected for this experiment. This time, the initial pose is located at (0.64, -0.91) within Quadrant 2 with an initial orientation of -0.28 rad. This position is chosen to locate within the tolerance window deduced from Section 6.3.5 in addition to being an Alternate Goal scenario. In this test, the power is also cycled to evaluate both algorithms at their best performance which is with minimal sensor error. The trajectory of both algorithms is plotted in Figure 6-34 which further validates the previously inferred trend. For example, the TEB planner commands the AWS equipped SECV by changing direction for a total of four times while the M-PBVS algorithm generates a path that only requires a change in direction once. The result of this is a much higher traveled distance of 8.91 meters for the TEB when compared to the 1.85 meters by the M-PBVS algorithm. Furthermore, the M-PBVS's efficiency is also prevalent when looking at Figure 6-36 where the position error reduced to zero in 11 seconds compared to 19 seconds from the TEB planner. In addition, the final achieved position of the TEB and M-PBVS algorithm are, (0.23, 0.17) and (0.02, -0.05), respectively which shows that the M-PBVS algorithm is much more accurate at position correction. In addition, the orientation error from Figure 6-37 further validates the ability of the M-PBVS algorithm as it reduces the orientation error within 3 seconds with an error of 0.71%. The TEB planner utilizes multiple turns to correct its orientation yet its final result yields less accuracy than the M-PBVS algorithm with an error of 60.71%. From both comparison tests in this section, it is evident that the simplicity of the M-PBVS algorithm achieves higher accuracy and efficiency in both

position and orientation correction when compared to the TEB planner. The following table tabulates the performance metrics between the two algorithms during this test.

Table 6-11 Pose Correction Comparison with Alternate Goal

	Final Pose	Pos. Error %	Ori. Error %	Total Distance
TEB	(0.23, 0.17, 0.17)	27.31%	60.71%	8.91 m
M-PBVS	(0.02, -0.05, -0.002)	4.31%	0.71%	1.85 m

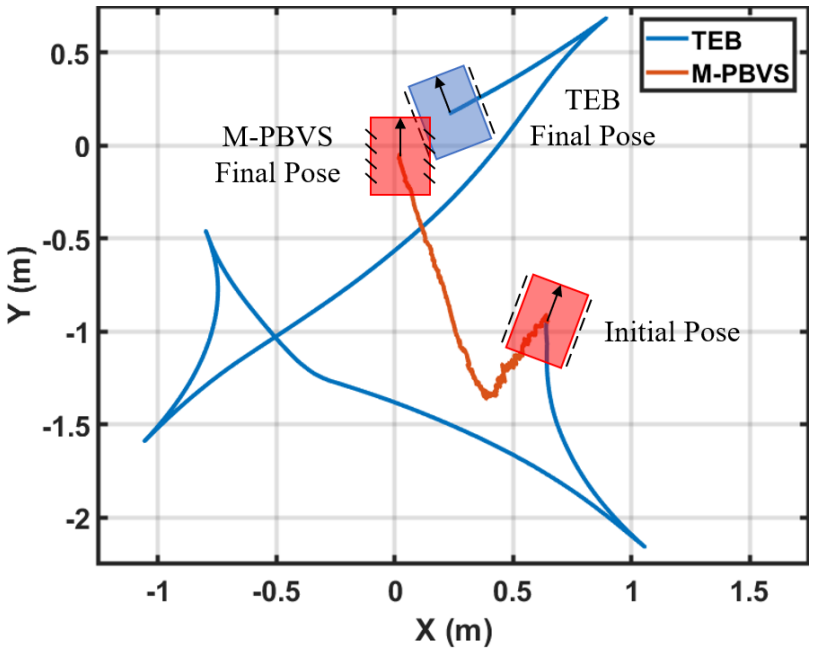

Figure 6-35 TEB vs M-PBVS Alternate Goal Trajectory

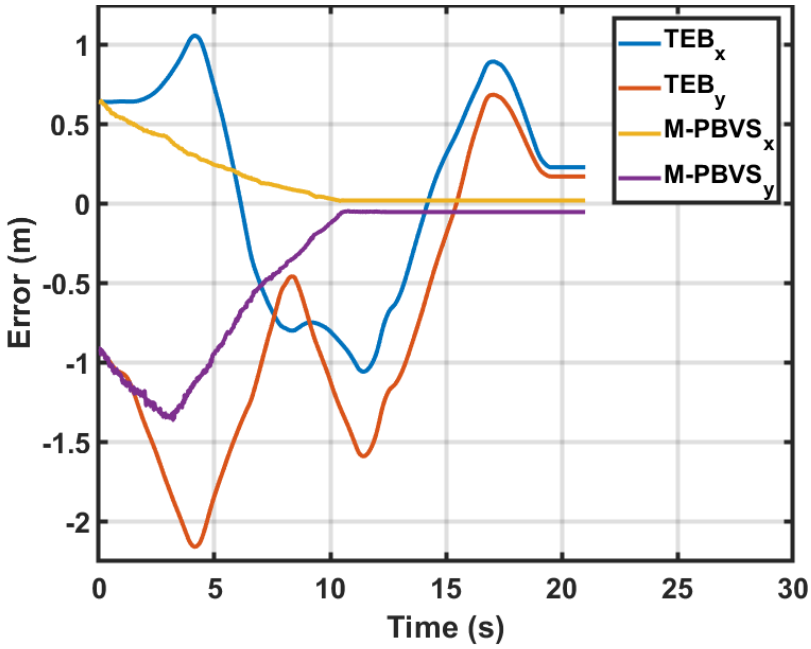

Figure 6-36 TEB vs M-PBVS Alternate Goal Position Error

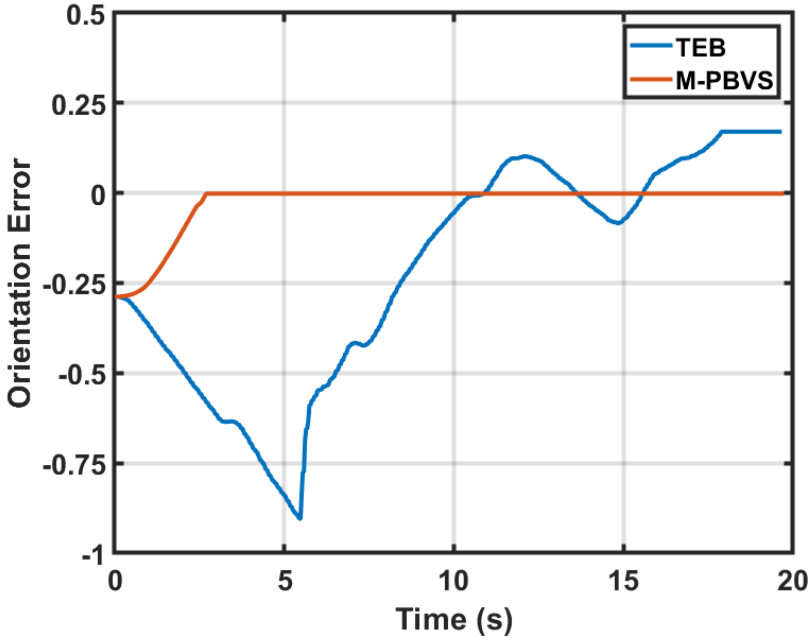

Figure 6-37 TEB vs M-PBVS Alternate Goal Orientation Error

6.6 Experiments Summary

In summary, this chapter presents experimental results that quantify the SECV's operational performance, navigation attributes, the proposed M-PBVS algorithm's abilities and its comparison with the current state of the art planner, TEB. The summary from each of these four categories is discussed below.

6.6.1 Vehicle Performance Experiments Conclusion

The obtained results from a series of testing in this section exceeded user satisfaction as they showed that the vehicle prototype is in a functioning state as it can travel at a max speed of 6.5 km/h with a steering response time of 0.8s. These values were deemed acceptable for future navigation research which are often conducted with speeds as low as 1.8 km/h for platforms this scale. These parameters are summarized in the table below.

Table 6-12 Vehicle Performance Parameters

Parameter	Value
Max Speed	6.5 km/h
Max Acceleration	1.64 m/s^2
Braking Distance	3.25 ft
Steering Time	0.8 s
Ramp Travel Index	2616.21
Static Roll Threshold	50 $degrees$
Maximum Grade	30 $degrees$
Turning Radius	AWS: 42.63 $inches$, $4AS$: 54.88 $inches$, FWS, 77.88 $inches$
Slalom	AWS: 1.52m, $4AS$: 1.83m, FWS: 2.44 m

6.6.2 Autonomous Navigation Experiments Conclusion

In this section, autonomous navigation methodologies from Chapter 4 were implemented with the onboard hardware and software architecture as described in Chapter 3. The result is a SECV that is capable of planning paths in a previously acquired map that accounts for building floorplan as well as new obstacles. To evaluate both the modular software and electronics architecture in addition to the lower controllers, three different tests were conducted in a hallway with flat smooth surfaces and static, opaque obstacles. The first test featured just one obstacle in the middle of the hallway to check if the SECV can clear it. The second test featured two obstacles back to back with the second one partially hiding behind the first. The intent was to examine if the vehicle can respond fast enough after clearing the first obstacle to subsequently avoid the second. In addition, the obstacles were placed in a way that required the SECV to perform a slalom maneuver which provided insight in to its steering ability. Lastly, a parking test that required a sharp turn in to a tight space was also conducted. All three tests successfully showed the SECV's ability to autonomously navigate and avoid collisions when given an initial and desired pose. Furthermore, the developed steering and speed control as well as the incremental localization algorithm also proved to be fully functioning and able to keep up with the high-level path planning.

Although the standard navigation methodologies tested in Section 6.3 showed promising capabilities in point to point navigation applications, these approaches were vulnerable to sensor drift when traveling for a long distance. To further study the effects of sensor drift, nine different initial poses were chosen for the SECV to start. A common desired goal was then selected and the final achieved poses from each of the nine initial

poses were recorded based on using TEB and AWS configuration. The results showed that the arrival position and orientation error are proportionally related to the distance between the desired and starting pose. In other words, SECVs that came from the farthest initial pose finished with the largest deviation from the desired. By taking all the deviations from each test, an average error of 1.5 meters was calculated. This error was then represented as a tolerance window in the form of a circle with a 1.5-meter radius. This was an important experimental result as it provided an approximated initial pose deviation for the proposed M-PBVS algorithm.

6.6.3 M-PBVS Experiments Conclusion

After the navigation methodologies were implemented and the electronic and software architecture were proven to be fully functioning, a close quarter pose correction algorithm was developed to take advantage of the SECV's multi-steerable system. The intent of this algorithm is to mitigate the drift issue that is common amongst all odometry sensors. Based on the drift results acquired from the previous test, the proposed M-PBVS algorithm must be able to bring the 1.5-meter, and 12.56-degree tolerance down to within 10 centimeters and 3 degrees, respectively, to ensure successful docking. To begin, the accuracy of the pose estimation algorithm was first examined as it ultimately dictates the accuracy of the M-PBVS algorithm. From this test, it was determined that the pose estimation algorithm featured an average error of 3.5 centimeters. To examine the performance of the proposed M-PBVS algorithm, two tests were conducted which are namely; Direct Goal and Alternate Goal. For the Direct Goal experiment, the M-PBVS algorithm achieved a final position and orientation error of 5.78% and 1.07%, respectively which translated to a position error of approximately 7.20 centimeter and an orientation

error of 0.17 degree. On the other hand, the Alternate Goal experiment achieved a final position and orientation error of 5.39 centimeter and 0.13 degree which translated to a percentage of 4.31% and 0.71%, respectively. From these results, both tests yield successful outcome as the vehicle was able to arrive at the intended goal with results well below the required.

6.6.4 TEB vs M-PBVS Experiments Conclusion

In this section, the proposed M-PBVS algorithm was compared with TEB to evaluate its performance upgrades in pose correction. The goal was to create a complete algorithm that enabled the SECV to travel long distances while arriving with centimeter accuracy. Two tests were conducted to represent both the Direct and Alternate goal scenarios where the results showed that the M-PBVS algorithm was far more accurate and faster than TEB at close quarters pose correction. This was because the M-PBVS algorithm utilized a visual landmark to correct its pose effectively in a two-stage manner while the TEB algorithm utilized the map and the localization algorithm. Not to mention, the TEB planner was also constrained by the SECV's minimum turning radius. As a result of these limitations, the TEB planner finished the Direct Goal test with a total traveled distance of 6.10 meters compared to the 1.44 meters achieved by the M-PBVS algorithm. The same performance trend was apparent in the Alternate Goal test where the M-PBVS algorithm traveled nearly 7.06 meters less. In addition, the M-PBVS algorithm repeatedly scored a higher accuracy as presented by both its position and orientation error percentage when compared to the TEB's. These results validated the benefits of the M-PBVS algorithm as it was able to capitalize on the mechanical design of the proposed SECV platform.

Chapter 7. Conclusion and Future Work

7.1 Conclusion

Today, automotive research is shifting focus towards driver-less vehicle technologies, which is closely related to autonomous navigation research for mobile robotics. Although traditional passenger vehicles which include two axles and front wheel steering are studied extensively with real-world deployment, an area that remains relatively unexplored is the application with military based multi-axle vehicles. Because of this, the motivation behind this thesis was to focus on this joint area. As such, a literature review of current armored vehicle designs, scaled MWMS robotic platforms, navigation methodologies and docking methods were presented with research gaps identified. From this, it was apparent that a lack of effort has been made in designing and developing a scaled eight-wheeled, MS, robotic vehicle. As a result, this thesis details the design and development of a scaled electric combat vehicle with the intention to improve upon the current LAVs by implementing multi-axle steering and autonomous capabilities.

To accomplish this, the SECV was proposed and completed in this thesis as a proof of concept which entailed designing custom mechanical systems from the ground up. These mechanical systems included the chassis, suspension, steering and driving assemblies. Electronics hardware including computing unit, sensors and actuators were successfully integrated within the vehicle. Furthermore, custom low-level software and drivers were created to form a complete software development environment within ROS. This environment is modular in nature; therefore, allowing future researchers to easily implement their code for any additional hardware and software features. In terms of the low-level control, PID algorithms were executed for both speed and steering angles.

Localization was achieved through an incremental method that combined both the wheel encoders and IMU for position and orientation, respectively. High-level path planning with the Dijkstra's algorithm and the Timed Elastic Band planner were successfully integrated within the SECV. Since the navigation algorithms described were prone to odometry sensor drift issues, a close quarter pose correction algorithm based on vision feedback was proposed to alleviate this problem. The proposed algorithm known as M-PBVS was based on traditional PBVS with modifications that centered around using alternative steering configurations. More specifically, the M-PBVS algorithm featured two stages with one employing Diamond Steer while the other Synchronous Steer. By doing so, waypoints were calculated based on vision feedback to achieve centimeter accuracy in pose correction.

Lastly, the complete SECV prototype was subjected to numerous vehicle performance testing to quantify its operational abilities and limits. With this knowledge, navigation algorithms that include mapping, localization and path planning were examined with physical experiments. Next, the proposed M-PBVS algorithm was experimentally assessed to evaluate its pose correction ability. This led to a comparison with the current state of the art planner, TEB, to gauge the performance difference. The results from the M-PBVS testing showed the proposed algorithm's ability to achieve superior accuracy in much faster time with less total traveled distance and change in directions. By combining the M-PBVS algorithm with standard path planners, the final SECV was capable of a point to point navigation that arrives precisely. In summary of the experiments conducted, the Vehicle Performance and Autonomous Navigation tests validated the SECV's ability to serve as an autonomous navigation research platform with its modular electronics hardware and software architecture. On the other hand, the M-PBVS tests showed that the proposed

algorithm performs precise close quarters navigation with greater results when compared to the standard path planners. The outcomes of the experiments were successful as they showed the SECV's ability by achieving performances that exceeded the requirements.

In conclusion, the proposed SECV prototype brings a novel and effective design to the joint field of military vehicles and mobile robotic platforms. Its alternative steering capabilities in combination with the software developed creates a complete proof of concept package that is validated experimentally. As such, the SECV exemplifies the potential to spark a more innovative future for the design of military vehicles beyond the contributions of this thesis.

7.2 Future Work

Recommended future work may be divided in to the following three sections.

7.2.1 SECV Improvements

- A battery management system with monitoring features to unite all batteries as a single source.
- Replace the drive system with shorter DC motors to increase serviceability.

7.2.2 Autonomous Navigation Improvements

- Include state of the art machine learning algorithms for obstacle avoidance, path following … etc which would entail adaptation of a GPU based computer such as the Nvidia Jetson within the SECV.
- Other types of work include a more sophisticated localization algorithm that employs sensor fusion between absolute and incremental sensors for precise long-distance navigation.

7.2.3 M-PBVS Improvements

- Panoramic cameras can be implemented to significantly increase the field of view which will improve the docking performance.
- Other methods of visual servoing such as image-based techniques can be explored to eliminate the need for a previously known visual tag model.
- Additionally, sensor fusion algorithms can be studied to improve the location estimation for further accuracy during docking.

References

[1] M. MacRae, "Karl Benz," ASME, April 2012. [Online]. Available: https://www.asme.org/engineering-topics/articles/automotive/karl-benz. [Accessed December 2018].

[2] J. Y. Wong, Theory of Ground Vehicles: Fourth Edition, Ottawa, Canada: John Wiley & Sons, INC., 2008.

[3] ROS, "About ROS," Open Source Robotics Foundation, [Online]. Available: http://www.ros.org/about-ros/. [Accessed Janurary 2019].

[4] C. Robotics, "Simulating Jackal," Clearpath Robotics, 2015. [Online]. Available: http://www.clearpathrobotics.com/assets/guides/jackal/simulation.html. [Accessed February 2019].

[5] "LAV-25 Light Armoured Vehicle," Army Recognition, 27 September 2018. [Online]. Available: https://www.armyrecognition.com/us_army_wheeled_and_armoured_vehicle_uk/lav-25_8x8_light_armoured_vehicle_technical_data_sheet_specifications_pictures_video.html.

[6] "Light Armored Vehicles - LAV III APC," General Dynamics Land Systems, [Online]. Available: https://www.gdls.com/products/light-armored-vehicles/LAV-III-APC.html. [Accessed January 2019].

[7] "LAV III," General Dynamics Land Systems - Canada, [Online]. Available: https://www.gdlscanada.com/products/LAV/LAV-III.html. [Accessed January 2019].

[8] G. D. L. S. -. Canada, "LAV 6.0," A General Dynamics Business, [Online]. Available: https://www.gdlscanada.com/products/LAV/LAV-6.0.html. [Accessed January 2019].

[9] A. Genys, "LAV 700," Military-Today.com, [Online]. Available: http://www.military-today.com/apc/lav_700.htm. [Accessed January 2019].

[10] G. D. L. S. -. Canada, "LAV 700," General Dynamics Land Systems, [Online]. Available: https://www.gdlscanada.com/products/LAV/LAV-700.html. [Accessed January 2019].

[11] M. R. Islam, F. H. Chowdhury, S. Rezwan, M. J. Ishaque, J. U. Akanda, A. S. Tuhel and B. B. Riddhe, "Novel Design and Performance Analysis of a Mars

Exploration Robot," in *Third International Conference on Research in Computational Intelligence and Communication Networks*, Kolkata, India, 2017.

[12] M. S. T. Kand, R. Sadeghian and M. T. Masouley, "Design, Analysis and Construction of a Novel Flexible Rover Robot," in *3rd RSI International Conference on Robotics and Mechatronics*, Tehran, Iran, 2015.

[13] A. K. Gupta and V. K. Gupta, "Design and Development of Six-wheeled Multi-terrain Robot," in *International Conference on Control, Automation, Robotics and Embedded Systems*, Jabalpur, India, 2013.

[14] V. S. Kumar, I. Gogul, M. D. Raj, S. K. Pragadesh and J. S. Sebastin, "Smart Autonomous Gardening Rover with Plant Recognition Using Neural Networks," *Procedia Computer Science,* vol. 93, pp. 975-981, 2016.

[15] J. Yuan, W. Zhang, Y. Song, L. Du and F. Li, "Design of Novel Terrain Adaptable Cascading Wheeled Mobile Platform with Passive Planetary Swing Structure," in *IEEE International Conference on Robotics and Biomimetics*, Phuket, Thailand, 2011.

[16] K. Suzuki, A. Nakano, G. Endo and S. Hirose, "Development of Multi-wheeled Snake-like Recuse Robots with Active Elastic Trunk," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vilamoura, Algarve, Portugal, 2012.

[17] G. Kouros and L. Petrou, "PANDORA Monstertruck: A 4WS4WD Car-Like Robot for Autonomous Exploration in Unknown Environments," in *12th IEEE Conference on Industrial Electronics and Applications*, Siem Reap, Cambodia, 2017.

[18] T.-H. S. Li, M.-H. Lee, C.-W. Lin, G.-H. Liou and W.-C. Chen, "Design of Autonomous and Manual Driving System for 4WIS4WID Vehicle," *IEEE Access*, vol. 4, pp. 2256-2271, 2016.

[19] Q. Qiu, Z. Fan, Z. Meng, Q. Zhang, Y. Cong, B. Li, N. Wang and C. Zhao, "Extended Ackerman Steering Principle for the Coordinated Movement Control of a Four Wheel Drive Agricultural Mobile Robot," *Computers and Electronics in Agriculture,* vol. 152, pp. 40-50, 2018.

[20] M. H. Ko, K. C. Kim, A. Suprem, N. P. Mahalik and B. S. Ryuh, "4WD Mobile Robot for Autonomous Steering using Single Camera Based Vision System," *International Journal of Intelligent Unmanned Systems*, vol. 2, no. 3, 2013.

[21] Y. Ye, L. He and Q. Zhang, "Steering Control STrategies for a Four-Wheel-Independent-Steering Bin Managing Robot," *IFAC-PapersOnLine,* vol. 49, no. 16, pp. 39-44, 2016.

[22] C. Yi, S. Jeong and J. Cho, "Map Representation for Robots," *Smart Computing Review,* vol. 2, no. 1, 2012.

[23] H. Cheng, H. Chen and Y. Liu, "Topological Indoor Localization and Navigation for Autonomous Mobile Robot," *IEEE Transactions on Automation Science and Engineering,* vol. 12, no. 2, 2015.

[24] C. Ye, "Navigating a Mobile Robot by a Traversability Field Histogram," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 37, no. 2, 2007.

[25] J. S. Oh, Y. H. Choi, J. Park Bae and Y. F. Zheng, "Complete Coverage Navigation of Cleaning Robots Using Triangular-Cell-Based Map," *IEEE Transactions on Industrial Electronics,* vol. 51, no. 3, 2004.

[26] S. Riisgaard and M. R. Blas, SLAM for Dummies: A Tutorial Approach to Simultaneous Localization and Mapping, MIT OCW, 2005.

[27] H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping: Part 1," *IEEE Robotics & Automation Magazine,* 2006.

[28] T. Bailey and H. Durrant-Whyte, "Simultaneous Localization and Mapping (SLAM): Part II," *IEEE Robotics & Automation Magazine,* 2006.

[29] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid and J. J. Leonard, "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust Perception Age," *IEEE Transactions on Robotics,* vol. 32, no. 6, 2016.

[30] P. Kim, J. Chen and Y. K. Cho, "SLAM-Driven Robotic Mapping and Registration of 3D Point Clouds," *Automation in Construction,* vol. 89, 2018.

[31] D. Droeschel, M. Schwarz and S. Behnke, "Continuous Mapping and Localization for Autonomous Navigation in Rough Terrain Using a 3D Laser Scanner," *Robotics and Autonomous Systems,* vol. 88, 2017.

[32] J. Cao, B. Zeng, J. Liu, Z. Zhao and Y. Su, "A Novel Relocation Method for Simultaneous Localization and Mapping based on Deep Learning Algorithm," *Computers and Electrical Engineering,* vol. 63, 2017.

[33] R. Koch, S. May, P. Murmann and A. Nuchter, "Identification of Transparent and Specular Reflective Material in Laser Scans to Discriminate Affected

Measurements for Faultless Robotic SLAM," *Robotics and Autonomous Systems,* vol. 87, 2017.

[34] L. Teslic, I. Skrjanc and G. Klancar, "Using a LRF Sensor in the Kalman-Filtering-Based Localization of a Mobile Robot," *ISA Transactions,* vol. 49, 2010.

[35] K. Lenac, A. Kitanov, R. Cupec and I. Petrovic, "Fast Planar Surface 3D SLAM Using LIDAR," *Robotics and Autonomous Systems,* vol. 92, 2017.

[36] P. d. l. Puente and D. Rodriguez-Losada, "Feature Based Graph SLAM with High Level Representation using Rectangles," *Robotics and Autonomous Systems,* vol. 63, 2015.

[37] J. Zhang and S. Singh, "Low-Drift and Real-Time Lidar Odometry and Mapping," *Auton robot,* vol. 41, 2017.

[38] W. Xi, Y. Ou, J. Peng and G. Yu, "A New Method for Indoor Low-cost Mobile Robot SLAM," in *International Conference on Information and Automation*, Macau, China, 2017.

[39] G. Galleogs, M. Meilland, P. Rives and A. I. Comport, "Appearance-Based SLAM Relying on a Hybrid Laser/Omnidirectional Sensor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems,* Taipei, Taiwan, 2010.

[40] F. Tungadi and L. Kleeman, "Autonomous Loop Exploration and SLAM with Fusion of Advanced Sonar and Laser Polar Scan Matching," *Robotica,* vol. 30, 2012.

[41] K.-H. Lin, C.-H. Chang, A. Dopfer and C.-C. Wang, "Mapping and Localization in 3D Environments Using a 2D Laser Scanner and a Stereo Camera," *Journal of Information Science and Engineering,* vol. 28, 2012.

[42] R. C. Luo and C. C. Lai, "Enriched Indoor Map Construction Based on Multisensor Fusion Approach for Intelligent Service Robot," *IEEE Transactions on Industrial Electronics,* vol. 59, no. 8, 2012.

[43] R. C. Luo and C. C. Lai, "Multisensor Fusion-Based Concurrent Environment Mapping and Moving Object Detection for Intelligent Service Robotics," *IEEE Transactions on Industrial Electronics,* vol. 61, no. 8, 2014.

[44] K. Kamarudin, S. Mamduh, A. Yeon, R. Visvanathan, A. Shakaff, A. Zakaria, L. M. Kamarudin and N. A. Rahim, "Improving Performance of 2D SLAM Methods by Complementing Kinect with Laser Scanner," in *IEEE International Symposium on Robotics and Intelligent Sensors,* 2015.

[45] A. Ratter and C. Sammut, "Fused 2D/3D Position Tracking for Robust SLAM on Mobile Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hamburg, Germany, 2015.

[46] X. Sun, F. Sun, B. Wang, J. Yin, X. Sheng and Q. Xiao, "Robotic Autonomous Exploration SLAM Using Combination of Kinect and Laser Scanner," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Daegu, Korea, 2017.

[47] C.-C. Lai and K.-L. Su, "Artificial Beacons with RGB-D Environment Mapping for Indoor Mobile Robot Localization," *Sensors and Materials,* vol. 28, no. 6, 2016.

[48] C. Qian, H. Liu, J. Tang, Y. Chen, H. Kaartinen, A. Kukko, L. Zhu, X. Liang, L. Chen and J. Hyyppa, "An Integrated GNSS/INS/LiDAR-SLAM Positioning Method for Highly Accurate Forest Stem Mapping," *Journal of Remote Sensing*, vol. 9, no. 3, 2017.

[49] A. R. Vidal, H. Rebecq, T. Horstschaefer and D. Scaramuzza, "Ultimate SLAM? Combining Events, Imaes, and IMU for Robust Visual SLAM in HDR and High-Speed Scenarios," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, 2018.

[50] J. J. Leonard and H. F. Durrant-Whyte, "Mobile Robot Localization by Tracking Geometric Beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, 1991.

[51] Y. Kim, J. An and J. Lee, "Robust Navigational System for a Transporter Using GPS/INS Fusion," *IEEE Transactions on Industrial Electronics,* vol. 65, no. 4, 2018.

[52] F. Peyret, D. Betaille, C. Pinana, R. Toledo-Moreo, A. Skarmeta and M. Ortiz, "GNSS Autonomous Localization," *IEEE Robotics & Automation Magazine*, 2014.

[53] F. Aghili and A. Salerno, "Driftless 3D Attitude Determination and Positioning of Mobile Robots by Integration of IMU with Two RTK GPSs," *IEEE/ASME Transactions on Mechatronics,* vol. 18, no. 1, 2013.

[54] M. M. Atia, S. Liu, H. Nematallah, T. B. Karamat and A. Noureldin, "Integrated Indoor Navigation System for Ground Vehicles with Automatic 3D Alignment and Position Initialization," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, 2015.

[55] S. Ginzburg and S. Nokleby, "Indoor Localization of an Omni-Directional Wheeled Mobile Robot," *Transactions of the Canadian Society for Mechanical Engineering*, vol. 37, no. 4, 2013.

[56] J. Jung, S.-M. Lee and H. Myung, "Indoor Mobile robot Localization and Mapping Based on Ambient Magnetic Fields and Aiding Radio Sources," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 7, 2015.

[57] B. Gozick, K. P. Subbu, R. Dantu and T. Maeshiro, "Magnetic Maps for Indoor Navigation," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 12, 2011.

[58] P. Merriaux, Y. Dupuis and P. S. X. Vasseur, "Fast and Robust Vehicle Positioning on Graph-based Representation of Drivable Maps," in *IEEE International Conference on Robotics and Automation*, Seattle, Washington, 2015.

[59] J.-H. Kim and J.-C. Lee, "Dead-Reckoning Scheme for Wheeled Mobile Robots Moving on Curved Surfaces," *Journal of Intelligent and Robotic Systems*, vol. 79, 2015.

[60] B.-S. Cho, W.-s. Moon, W.-J. Seo and K.-R. Baek, "A Dead Reckoning Localization System for Mobile Robots Using Inertial Sensors and Wheel Revolution Encoding," *Journal of Mechanical Science and Technology*, vol. 25, no. 11, 2011.

[61] F. Chenavier and J. L. Crowley, "Position Estimation for a Mobile Robot Using Vision and Odometry," in *IEEE International Conference on Robotics and Automation*, Nice, France, 1992.

[62] R. Jiang, S. Yang, S. S. Ge, H. Wang and T. H. Lee, "Geometric Map-Assisted Localization for Mobile Robots Based on Uniform-Gaussian Distribution," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, 2017.

[63] H. Lategahn and C. Stiller, "Vision-Only Localization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, 2014.

[64] M. A. Brubaker and A. U. R. Geiger, "Map-Based Probabilistic Visual Self-Localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 4, 2016.

[65] S. H. Cho and S. Hong, "Map Based Indoor Robot Navigation and Localization Using Laser Range Finder," in *11th International Control, Automation, Robotics and Vision Conference*, Singapore, 2010.

[66] T. Epton and A. Hoover, "Improving Odometry Using a Controlled Point Laser," *Autonomous Robot,* vol. 32, 2012.

[67] Z. Zhu and C. Taylor, "Conservative Uncertainty Estimation in Map-Based Vision-Aided Navigation," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 53, no. 2, 2017.

[68] Y. Lu and D. Song, "Visual Navigation Using Heterogeneous Landmarks and Unsupervised Geometric Constraints," *IEEE Transactions on Robotics,* vol. 31, no. 3, 2015.

[69] S.-B. Han, J.-H. Kim and H. Myung, "Landmark-Based Particle Localization Algorithm for Mobile Robots With a Fish-Eye Vision System," *IEEE/ASME Transactions on Mechatronics,* vol. 18, no. 6, 2013.

[70] X. Li, X. Zhang, B. Zhu and X. Dai, "A Visual Navigation Method of Mobile Robot Using a Sketched Semantic Map," *International Journal of Advanced Robotic Systems,* vol. 9, 2012.

[71] S. Nedevschi, V. Popescu, R. Danescu, T. Marita and F. Oniga, "Accurate Ego-Vehicle Global Localization at Intersections Through Alignment of Visual Data With Digital Map," *IEEE Transactions on Intelligent Transportation Systems,* vol. 14, no. 2, 2013.

[72] M. Beinhofer, J. Muller and W. Burgard, "Effective Landmark Placement for Accurate and Reliable Mobile Robot Navigation," *Robotics and Autonomous Systems,* 2013.

[73] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik,* vol. 1, pp. 269-271, 1959.

[74] P. E. Hart, N. J. Nilsson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics,* vol. 4, no. 2, pp. 100-107, 1968.

[75] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *The International Journal of Robotics Research,* vol. 5, no. 1, 1986.

[76] A. Stentz, "Optimal and Efficient Path Planning for Partially-Known Environments," in *The International Conference on Robotics and Automation*, 1994.

[77] O. Khatib and S. Quinlan, "Elastic bands: connecting path planning and control," in *IEEE International Conference on Robotics and Automation*, Atlanta, GA, 1993.

[78] D. Fox, W. Burgard and S. Thrun, "The dynamic window approach to cliision avoidance," *Robotics & Automation Magazine,* pp. 23-33, 1997.

[79] C. Roesmann, T. Woesch, F. Hoffman and T. Bertram, "Trajectory modification considering dynamic constraints of autonomous robots," in *ROBOTIK 2012; 7th German Conference on Robotics,* Munich, Germany, 2012.

[80] G. J. Agin, "Real Time Control of a Robot with a Mobile Camera," SRI International, Menlo Park, California, 1979.

[81] Y. Xu, J. Peng, W. Yu, Y. Fang and W. Liu, "Three-Step Epipolar-Based Visual Servoing for Nonholonomic Robot with FOV Constraint," *Jounral of Applied Mathematics,* 2014.

[82] J. Chen, B. Jia and K. Zhang, "Trifocal Tensor-Based Adaptive Visual Trajectory Tracking Control of Mobile Robots," *IEEE Transactions on Cybernetics,* vol. 47, no. 11, 2017.

[83] F. Chaumette and S. Hutchinson, "Visual Servo Control Part I: Basic Approaches," *IEEE Robotics & Automation Magazine,* 2006.

[84] F. Chaumette and S. Hutchinson, "Visual Servo Control Part II: Advanced Approaches," *IEEE Robotics & Automation Magazine,* 2007.

[85] I. Siradjuddin, I. A. Siradjuddin and S. Adhisuwignjo, "An Image Based Visual Control Law for a Differential Drive Mobile Robot," *International Journal of Mechanical & Mechatronics Engineering,* vol. 15, no. 6, 2015.

[86] G. L. Mariottini, G. Oriolo and D. Prattichizzo, "Image-Based Visual Servoing for Nonholonomic Mobile Robots Using Epipolar Geometry," *IEEE Transactions on Robotics,* vol. 23, no. 1, 2007.

[87] H. M. Becerra, J. B. Hayet and C. Sagues, "A Single Visual-Servo Controller of Mobile Robots with Super-Twisting Control," *Robotics and Autonomous Systems,* vol. 62, 2013.

[88] G. Allibert, E. Courtial and F. Chaumette, "Predictive Control for Constrained Image-Based Visual Servoing," *IEEE Transactions on Robotics,* vol. 26, no. 5, 2010.

[89] H. Lang, Y. Wang and C. W. de Silva, "Visual Servoing with LQR Control for Mobile Robots," in *IEEE International Conference on Control and Automation*, Xiamen, China, 2010.

[90] M. Ito, T. Ebata and M. Shibata, "Vision-based Switching Control Strategy for a Nonholonomic Wheeled Mobile Robot with a Pan Camera," 2013.

[91] M. Liu, C. Pradalier and R. Siegwart, "Visual Homing From Scale With an Uncalibrated Omnidirectional Camera," *IEEE Transactions on Robotics,* vol. 29, no. 6, 2013.

[92] H. Aliakbarpour, O. Tahri and H. Araujo, "Visual Servoing of Mobile Robots using Non-Central Catadioptric Cameras," *Robotics and Autonomous Systems,* vol. 62, 2014.

[93] X. Liang, H. Wang, Y.-H. Liu, W. Chen and Z. Jing, "Image-based Position Control of Mobile Robots with a Completely Unknown Fixed Camera," *IEEE Transactions on Automatic Control,* 2018.

[94] N. Jiang, J. Lv, Y. Kobayashi and T. Emaru, "Adaptive Image-based Visual Servoing of Nonholonomic Mobile Robot With On-Board Camera," in *IEEE/SICE International Symposium on System Integration*, Nagoya, Japan, 2015.

[95] C. Lopez-Franco, M. Lopez-Franco, A. Y. Alanis, J. Gomez-Avila and N. Arana-Daniel, "Real-Time Inverse Optimal Neural Control for Image Based Visual Servoing with Nonholonomic Mobile Robots," *Mathematical Problems in Engineering*, 2015.

[96] F. Ke, Z. Li and C. Yang, "Robust Tube-Based Predictive Control for Visual Servoing of Constrained Differential-Drive Mobile Robots," *IEEE Transactions on Industrial Electronics,* vol. 65, no. 4, 2018.

[97] A. Cherubini, F. Chaumette and G. Oriolo, "An Image-based Visual Servoing Scheme for Following Paths with Nonholonomic Mobile Robots," in *10th International Conference on Control, Automation, Robotics and Vision*, Hanoi, Vietnam, 2008.

[98] A. Khelloufi, N. Ouadah, A. Oualid and N. Achour, "Image-based Path Following for Nonholonomic Mobile Robot," in *3rd International Conference on Control, Engineering & Information Technology*, Tlemcen, Algeria, 2015.

[99] H. M. Becerra, C. Sagues, Y. Mezouar and J.-B. Hayet, "Visual Navigation of Wheeled Mobile Robots using Direct Feedback of a Geometric Constraint," *Journal of Autonomous Robots,* vol. 37, 2014.

[100] S. R. Bista, P. R. Giordano and F. Chaumette, "Appearance-Based Indoor Navigation by IBVS Using Line Segments," *IEEE Robotics and Automation Letters,* vol. 1, no. 1, 2016.

[101] S. R. Bista, P. R. Giordano and F. Chaumette, "Combining Line Segments and Points for Appearance-based Indoor Navigation by Image Based Visual Servoing," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vancouver, Canada, 2017.

[102] G. Caron, E. Marchand and E. M. Mouaddib, "Photometric Visual Servoing for Omnidirectional Cameras," *Autonomous Robot*, vol. 35, 2013.

[103] F. Safia and C. Fatima, "Visual Path Following by an Omnidirectional Mobile Robot using 2D Visual Servoing," in *The 5th International Conference on Electrical Engineering*, Boumerdes, Algeria, 2017.

[104] D. A. de Lima and A. C. Victorino, "A Visual Servoing Approach for Road Lane Following with Obstacle Avoidance," in *IEEE 17th International Conference on Intelligent Transportation Systems*, Qingdao, China, 2014.

[105] A. Cheurbini, F. Spindler and F. Chaumette, "Autonomous Visual Navigation and Laser-Based Moving Obstacle Avoidance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, 2014.

[106] C.-Y. Tsai, K.-T. Song, X. Dutoit, H. V. Brussel and M. Nuttin, "Robust Visual Tracking Control System of a Mobile Robot Based on a Dual-Jacobian Visual Interaction Model," *Robotics and Autonomous Systems*, vol. 57, 2009.

[107] M. Ito, T. Hiratsuka and M. Shibata, "Image-Based Visual Target Following for a Nonholonomic Wheeled Mobile Robot with a Single Fixed Camera," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 1, 2012.

[108] Z. L. Wang and Y. H. Liu, "Visual Regulation of a Nonholonomic Wheeled Mobile Robot with Two Points Using Lyapunov Functions," in *IEEE International Conference on Mechatronics and Automation*, Xi'an, China, 2010.

[109] H. M. Becerra and C. Sagues, "Dynamic Pose-Estimation from the Epipolar Geometry for Visual Servoing of Mobile Robots," in *IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011.

[110] M. F. Yahya and M. R. Arshad, "Position-Based Visual Servoing for Underwater Docking of an Autonomous Underwater Vehicle," in *IEEE 6th International Conference on Underwater System Technology: Theory & Applications*, Penang, Malaysia, 2016.

[111] Y. Wang, J. Thunberg and X. Hu, "A Transformation of the Position Based Visual Servoing Problem into a Convex Optimization Problem," in *51st IEEE Conference on Decision and Control*, Maui, Hawaii, 2012.

[112] Z. Li and C. Yang, "Vision-Based Model Predictive Control for Steering of a Nonholonomic Mobile Robot," *IEEE Transactions on Control Systems Technology,* vol. 24, no. 2, 2016.

[113] R. O. Fernandes, M. Fonseca and A. G. S. Congeicao, "Visual Servoing with Fiducial Landmarks for Differential Drive Mobile Robot," in *Robotics Symposium and 2017 Brazilian Symposium on Robotics*, Curitiba, Brazil, 2017.

[114] D. Guo and K. K. Leang, "Position and Linear Velocity Estimation for Position-Based Visual Servo Control of an Aerial Robot in GPS-Denied Environments," in *ASME Dynamic Systems and Control Conference*, Tysons, Virginia, 2017.

[115] I. Sa and P. Corke, "Close-quarters Quadrotor Flying for a Pole Inspection with Position Based Visual Servoing and High-Speed Vision," in *International Conference on Unmanned Aircraft Systems*, Orlando, USA, 2014.

[116] J.-H. Jean and F.-L. Lian, "Robust Visual Servo Control of a Mobile Robot for Object Tracking Using Shape Parameters," *IEEE Transactions on Control Systems Technology,* vol. 20, no. 6, 2012.

[117] A. Cherubini and F. Chaumette, "A Position-based Visual Servoing Scheme for Following Paths with Nonholonomic Mobile Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, France, 2008.

[118] A. Cherubini, F. Chaumette and G. Oriolo, "Visual Servoing for Path Reaching with Nonholonomic Robots," *Robotica,* vol. 29, 2011.

[119] N. R. Gans and S. A. Hutchinson, "Stable Visual Servoing Through Hybrid Switched-System Control," *IEEE Transactions on Robotics,* vol. 23, no. 3, 2007.

[120] E. Malis, F. Chaumette and S. Boudet, "2-1/2-D Visual Servoing," *IEEE Transactions on Robotics and Automation,* vol. 15, no. 2, 1999.

[121] A. Parikh, R. Kamalapurkar, H.-Y. Chen and W. E. Dixon, "Homography Based Visual Servo Control with Scene Reconstruction," in *IEEE 54th Annual Conference on Decision and Control*, Osaka, Japan, 2015.

[122] Y. Fang, W. E. Dixon, D. M. Dawson and P. Chawda, "Homography-Based Visual Servo Regulation of Mobile Robots," *IEEE Transactions on Systems, Man , And Cybernetics,* vol. 35, no. 5, 2005.

[123] X. Zhang, Y. Fang and X. Liu, "Motion-Estimation-Based Visual Servoing of Nonholonomic Mobile Robots," *IEEE Transactions on Robotics,* vol. 27, no. 6, 2011.

[124] G. Mekonnen, S. Kumar and P. M. Pathak, "Wireless Hybrid Visual Servoing of Omnidirectional Wheeled Mobile Robots," *Robotics and Autonomous Systems,* vol. 75, 2016.

[125] M.-F. Ricky Lee, F. Hsin and S. Chiu, "A Hybrid Visual Servo Control System for the Autonomous Mobile Robot," in *IEEE/SICE International Symposium on System Integration*, Kobe, Japan, 2013.

[126] B. Li, Y. Fang and X. Zhang, "Projection Homography Based Uncalibrated Visual Servoing of Wheeled Mobile Robots," in *53rd IEEE Conference on Decision and Control*, Los Angeles, USA, 2014.

[127] B. Li, Y. Fang and X. Zhang, "Visual Servo Regulation of Wheeled Mobile Robots With an Uncalibrated Onboard Camera," *IEEE/ASME Transactions on Mechatronics,* vol. 21, no. 5, 2016.

[128] X. Zhang, Y. Fang, B. Li and J. Wang, "Visual Servoing of Nonholonomic Mobile Robots With Uncalibrated Camera-to-Robot Parameters," *IEEE Transactions on Industrial Electronics,* vol. 64, no. 1, 2017.

[129] B. Li, X. Zhang, Y. Fang and W. Shi, "Visual Servo Regulation of Wheeled Mobile Robots With Simultaneous Depth Identification," *IEEE Transactions on Industrial Electronics,* vol. 65, no. 1, 2018.

[130] G. Lopez-Nicolas, N. R. Gans, S. Bhattacharya, C. Sagues, J. J. Guerrero and S. Hutchinson, "Homography-Based Control Scheme for Mobile Robots With Nonholonomic and Field-of-View Constraints," *IEEE Transactions on Systems, Man, And Cybernetics,* vol. 40, no. 4, 2010.

[131] B. Jia and S. Liu, "Switched Visual Servo Control of Nonholonomic Mobile Robots with Field-of-View Constraints Based on Homography," *Control Theory and Technology,* vol. 13, no. 4, 2015.

[132] B. Jia and S. Liu, "Homography-Based Visual Predictive Control of Tracked Mobile Robot with Field-of-View Constraints," *International Journal of Robotics and Automation,* vol. 30, no. 5, 2015.

[133] G. Hu, N. Gans, N. Fitz-Coy and W. Dixon, "Adaptive Homography-Based Visual Servo Tracking Control via a Quaternion Formulation," *IEEE Transactions on Control Systems Technology,* vol. 18, no. 1, 2010.

[134] W. Mackunis, N. Gans, A. Parikh and W. E. Dixon, "Unified Tracking and Regulation Visual Servo Control For Wheeled Mobile Robots," *Asian Journal of Control,* vol. 16, no. 3, 2014.

[135] B. Li, Y. Fang, G. Hu and X. Zhang, "Model-Free Unified Tracking and Regulation Visual Servoing of Wheeled Mobile Robots," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, 2016.

[136] O. F. S. Aviles, C. C. G. Segura, J. C. M. Hernandez, M. M. Mauledoux and M. S. Dutra, "Ackerman Model for a Six-Wheeled Robot," *Appplied Mechanics and Materials*, vol. 823, pp. 441-446, 2016.

[137] M. Stania, "Analysis of the Kinematics of an Eight-Wheeled Mobile Platform," *Solid State Phenomena*, vol. 198, pp. 67-72, 2013.

[138] E. A. Martinez-Garcia, E. Lerin-Garcia and R. Torres-Cordoba, "A Multi-configuration kinematic model for active drive/steer four-wheel robot structures," *Robotica*, vol. 33, no. 1, 2015.

[139] T. Oksanen and R. Linkolehto, "Control of four wheel steering using independent actuators," *IFAC Proceedings Volumes*, vol. 46, no. 18, pp. 159-163, 2013.

[140] C. Kim, A. M. Ashfaq, S. Kim, S. Back, Y. Kim, S. Hwang and J. Jang, "Motion Control of a 6WD/6WS wheeled platform with in-wheel motors to improve its maneuverability," *International Journal of Control, Automation and Systems*, vol. 13, pp. 434-442, 2015.

[141] W. G. Kim, J. Y. Kang and K. Yi, "Drive control system design for stability and maneuverability of a 6WD/6WS vehicle," *International Journal of Automotive Technology*, vol. 12, pp. 67-74, 2011.

[142] R. Oftadeh, M. M. Aref, R. Ghabcheloo and J. Mattila, "Bounded-velocity motion control of four wheel steered mobile robots," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Wollongong, NSW, Australia, 2013.

[143] A. P. Aliseichik and V. E. Pavlovsky, "The model and dynamic estimates for the controllability and comfortability of a multiwheel mobile robot motion," *Automation and Remote Control*, vol. 76, no. 4, pp. 675-688, 2015.

[144] D. Wang and F. Qi, "Trajectory Planning for a Four-Wheel-Steering Vehicle," in *IEEE International Conference on Robotics & Automation*, Seoul, Korea, 2001.

[145] T. M. Howard and A. Kelly, "Trajectory and Spline Generation for All-Wheel Steering Mobile Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 2007.

[146] C.-J. Lin, S.-M. Hsiao, Y.-H. Wang, C.-H. Yeh, C.-F. Huang and T.-H. S. Li, "Design and implementation of a 4WS4WD mobile robot and its control

applications," in *International Conference on System Science and Engineering*, Budapest, Hungary, 2013.

[147] H. Bo, "Precise Navigation for a 4WS mobile robot," *Journal of Zhejiang University*, vol. 7, pp. 185-193, 2006.

[148] P. Dai and J. Katupitiya, "Path planning and tracking of a 4WD4WS vehicle to be driven under force control," in *IEEE International Conference on Mechatronics and Automation*, Tianjin, China, 2014.

[149] Y. Li, L. He and L. Yang, "Path-following control for multi-axle car-like wheeled mobile robot with nonholonomic constraint," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Wollongong, NSW, Australia, 2013.

[150] R. Solea, A. Filipescu, S. Filipescu and Dumitrascu, "Sliding-mode controller for four-wheel-steering vehicle: Trajectory-tracking problem," in *8th World Congress on Intelligent Control and Automation*, Jinan, China, 2010.

[151] S. Ghaffari and M. R. Homaeinezhad, "Intelligent path following of articulated eight-wheeled mobile robot with nonholonomic constraints," in *4th International Conference on Robotics and Mechatronics*, Tehran, Iran, 2016.

[152] S. Ghaffari and M. Homaeinezhad, "Autonomous path following by fuzzy adaptive curvature-based point selection algorithm for four-wheel steering car like mobile robot," *Journal of Mechanical Engineering Science*, vol. 232, no. 15, pp. 2655-2665, 2018.

[153] H. Ragheb, M. El-Gindy and Kishawy, "Torque distribution control for multi-wheeled combat vehicle," in *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Buffalo, New York, 2014.

[154] P. D'Urso and M. El-Gindy, "Development of control strategies of a multi-wheeled combat vehicle," *International Journal of Automation and Control*, vol. 12, no. 3, 2018.

[155] A. Mohamed, M. El-Gindy, J. Ren and H. Lang, "Optimal Collision-Free Path Planning for an Autonomous Multi-Wheeled Combat Vehicle," in *International Design Engineering TEchnical Conferences and Computers and Information in Engineering Conference*, Cleveland, Ohio, 2017.

[156] A. Mohamed, M. El-Gindy and J. Ren, "Design and Performance Analysis of Robust H-infinity Controller for a Scaled Autonomous Multi-Wheeled Combat Vehicle HEading Control," in *ASME International Design Engineering Technical*

Conferences & Computers and Information in Engineering Conference, Quebec City, Canada, 2018.

[157] W. G. Walter, The Living Brain, W. W. Norton & Company; Second Edition edition, 1963.

[158] M. Kim, H. W. Kim and N. Y. Chong, "Automated Robot Docking using Direction Sensing RFID," in *IEEE International Conference on Robotics and Automation*, Roma, Italy, 2007.

[159] Y. Hu, A. Vibhute, S. Foong and G. S. Soh, "Autonomous Docking of Miniature Spherical Robots with an External 2D Laser Rangefinder," in *IEEE Region 10 Conference*, Singapore, Singapore, 2016.

[160] K.-L. Su, Y.-L. Liao, S.-P. Lin and S.-F. Lin, "An Interactive Auto-recharging System for Mobile Robots," *International Journal of Automation and Smart Technology*, vol. 4, no. 1, 2014.

[161] K. L. Su, J. H. Guo, C. W. Hung and Y. C. Song, "Design an Auto-recharging System for Mobile Robots," *Applied Mechanics and Materials*, vol. 190, pp. 666-672, 2012.

[162] K. Roufas, Y. Zhang, D. Duff and M. Yim, "Six Degree of Freedom Sensing for Docking Using IR LED Emitters and Receivers," in *Experimental Robotics VII*, Waikiki, Hawaii, USA, 2000.

[163] Y. Niu, J. Zhang, T. Meng and H. Wang, "Design of a Home Surveillance Robot with Self-Recharging Capabilities," in *Third International Symposium on Knowledge Acquisition and Modeling*, Wuhan, China, 2010.

[164] G. Song, H. Wang, J. Zhang and T. Meng, "Automatic Docking System for Recharging Home Surveillance Robots," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 2, 2011.

[165] S. Saravanan and K. Ramamoorthy, "Implementation of Automatic Docking System Based Home Surveillance Robot," *Global Research and Development Journal for Engineering*, 2016.

[166] P. Won, M. Biglarbegian and W. Melek, "Development of an Effective Docking System for Modular Mobile Self-Reconfigurable Robots Using Extended Kalman Filter and Particle Filter," *Robotics*, vol. 4, pp. 25-49, 2015.

[167] N. Barnes and Z.-Q. Liu, "Fuzzy Control for Active Perceptual Docking," in *International Conference on Fuzzy Systems*, Melbourne, Victoria, 2001.

[168] M. C. Silverman, D. Nies, B. Jung and G. S. Sukhatme, "Staying alive: a docking station for autonomous robot recharging," in *IEEE International Conference on Robotics and Automation*, Washington, DC, 2002.

[169] T.-L. Chien, "Developing a vision-based auto-recharging system for mobile robots," *Artificial Life and Robotics*, vol. 16, no. 1, pp. 74-77, 2011.

[170] R. Quilez, A. Zeeman, N. Mitton and J. Vandaele, "Docking autonomous robots in passive docks with Infrared sensors and QR codes," in *International Conference on Testbeds and Research Infrastructures for the Development of Networks*, Vancouver, Canada, 2015.

[171] F. Guangrui and W. Geng, "Vision-based autonomous docking and re-charging system for mobile robot in warehouse environment," in *2nd International Conference on Robotics and Automation Engineering*, Shanghai, China, 2017.

[172] L. Santos, F. N. d. Santos, J. Mendes, N. Ferraz, J. Lima, R. Morais and P. Costa, "Path Planning for Automatic Recharging System for Steep-slope Vineyard Robots," *Third Iberian Robotics Conference*, vol. 1, pp. 261-272, 2018.

[173] P. Cui, W. Yan and Y. Wang, "Reactive Path Planning Approach for Docking Robots in Unknown Environment," *Journal of Advanced Transportation,* vol. 2017, pp. 1-11, 2017.

[174] H. Koyasu and M. Wada, "Plugin-docking system for autonomous charging using particle filter," in *Thirteenth International Conference on Quality Control by Artificial Vision*, Tokyo, Japan, 2017.

[175] W. Burgard, C. Stachniss, M. Bennewitz and K. Arras, Introduction to Mobile Robotics: Bayes Filter - Particle Filter and Monte Carlo Localization.

[176] C. Stachniss, Robot Mapping: Short Introduction to Particle Filters and Monte Carlo Localization.

[177] G. Grisetti, C. Stachniss and W. Burgard, "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters," *IEEE Transactions on Robotics,* vol. 23, no. 1, pp. 34-46, 2007.

[178] F. Chaumette and S. Hutchinson, "Visual Servo Control Part 1: Basic Approaches," 2006.

[179] "ViSP Visual Servoing Platform," Inria, [Online]. Available: https://visp.inria.fr/. [Accessed February 2019].

[180] F. Novotny, "Visp_auto_tracker," ROS.org, [Online]. Available: http://wiki.ros.org/visp_auto_tracker. [Accessed February 2019].

[181] P. Marin-Plaza, A. Hussein, D. Martin and A. d. l. Escalera, "Global and Local Path Planning Study in a ROS-Based Research Platform for Autonomous Vehicles," *Journal of Advanced Transportation,* vol. 2018, pp. 1-10, 2018.

[182] C. Army, "Light Armoured Vehicle (LAV) III," Government of Canada, 13 December 2017. [Online]. Available: http://www.army-armee.forces.gc.ca/en/vehicles/light-armoured-vehicle.page. [Accessed Janurary 2019].

[183] G. Grisetti, G. D. Tipaldi, C. Stachniss, W. Burgard and D. Nardi, "Fast and accurate SLAM with Rao-Blackwellized particle filters," *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 30-38, 2007.