แนวทางแบบบูรณาการเพื่อการระบุตำแหน่งพร้อมกันกับการทำแผนที่และระเบียบวิธีการวางแผนเส้นทางสำหรับหุ่นยนต์เคลื่อนที่ในร่ม

นายฮง คัก เหงียน

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2557

INTEGRATED APPROACH TO SIMULTANEOUS LOCALIZATION AND MAPPING WITH PATH PLANNING ALGORITHMS FOR INDOOR MOBILE ROBOTS

Mr. Hong Khac Nguyen

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2014

Thesis Title	INTEGRATED APPROACH TO SIMULTANEOUS LOCALIZATION AND MAPPING WITH PATH PLANNING ALGORITHMS FOR INDOOR MOBILE ROBOTS
By	Mr. Hong Khac Nguyen
Field of Study	Electrical Engineering
Thesis Advisor	Assistant Professor Manop Wongsaisuwan, Ph.D.

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

. . . . Dean of the Faculty of Engineering
(Professor Bundhit Eua-arporn, Ph.D.)

THESIS COMMITTEE

. . . . Chairman
(Professor David Banjerdpongchai, Ph.D.)

. . . . Thesis Advisor
(Assistant Professor Manop Wongsaisuwan, Ph.D.)

. . . . External Examiner
(Assistant Professor Itthisek Nilkhamhang, Ph.D.)

ฮง คัก เหงียน: แนวทางแบบบูรณาการเพื่อการระบุตำแหน่งพร้อมกันกับการทำแผนที่และระเบียบวิธีการวางแผนเส้นทางสำหรับหุ่นยนต์เคลื่อนที่ในร่ม (INTEGRATED APPROACH TO SIMULTANEOUS LOCALIZATION AND MAPPING WITH PATH PLANNING ALGORITHMS FOR INDOOR MOBILE ROBOTS)
อ. ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ. ดร.มานพ วงศ์สายสุวรรณ, 56 หน้า

วิทยานิพนธ์ฉบับนี้ศึกษาปัญหาการระบุตำแหน่งพร้อมกันกับการทำแผนที่ โดยบูรณาการกับระเบียบวิธีการวางแผนเส้นทางสำหรับหุ่นยนต์เคลื่อนที่ในร่ม โดยใช้การจำลองสถานการณ์ด้วยคอมพิวเตอร์ สิ่งแรกที่สนใจคือวิธีแก้ปัญหาการระบุตำแหน่งพร้อมกันกับการทำแผนที่โดยใช้วิธีการกรองแบบเบย์ (Bayes) ทั้งแบบอิงพารามิเตอร์และไม่อิงพารามิเตอร์ ในส่วนนี้ได้มีการตรวจสอบข้อดีและข้อเสียของระเบียบวิธีแต่ละแบบและความเป็นไปได้ที่จะใช้ระเบียบวิธีเหล่านี้เพื่อบูรณาการกับวิธีที่ได้นำเสนอโดยใช้การจำลองสถานการณ์เปรียบเทียบ ส่วนที่สองคือการศึกษาระเบียบวิธีการวางแผนเส้นทางแบบ A* ร่วมกับการใช้ตัวควบคุมแบบสัดส่วนเพื่อให้หุ่นยนต์สามารถตามรอยเส้นทางที่กำหนดจนถึงจุดหมายโดยไม่มีการชนกับสิ่งกีดขวางตลอดเส้นทาง ส่วนที่สามคือการศึกษาค่าสารสนเทศและค่าการเดินทางต่ำที่สุดที่อิงจากการสำรวจเพื่อประกอบการตัดสินใจของหุ่นยนต์ว่าจะเคลื่อนที่ต่อไปในตำแหน่งใด นอกจากนี้แบบจำลองแผนที่ในรูปแบบของแผนที่แบบอิงคุณลักษณะและแบบกริดกระจายยังมีบทบาทสำคัญในการอธิบายสิ่งแวดล้อมซึ่งได้ถูกพิจารณาในวิทยานิพนธ์ฉบับนี้ด้วย เพื่อรวมเทคนิคทั้งหมดเข้าด้วยกัน เราได้นำเสนอกรอบงานรวมสำหรับหุ่นยนต์เคลื่อนที่อัตโนมัติและประเมินประสิทธิผลโดยการจำลองระบบ

ภาควิชา วิศวกรรมไฟฟ้า	**ลายมือชื่อนิสิต** ....
สาขาวิชา วิศวกรรมไฟฟ้า	**ลายมือชื่อ อ.ที่ปรึกษาหลัก** ....
ปีการศึกษา 2557	

5570532921 : MAJOR ELECTRICAL ENGINEERING

KEYWORDS : MOBILE ROBOTS / SLAM / KALMAN FILTERS / PARTICLE FILTER / PATH PLANNING/ EXPLORATION

HONG KHAC NGUYEN : INTEGRATED APPROACH TO SIMULTANEOUS LOCALIZATION AND MAPPING WITH PATH PLANNING ALGORITHMS FOR INDOOR MOBILE ROBOTS.
ADVISOR : ASST. PROF. MANOP WONGSAISUWAN, Ph.D., 56 pp.

This thesis investigates Simultaneous Localization and Mapping (SLAM) problem in integration with path planning algorithm for an indoor mobile robot implemented in computer simulation. The first focus is on SLAM solutions using both parametric and non-parametric Bayesian filtering approach. Comparative simulations then prove pros and cons of each method as well as the feasibility to be used for the proposed integration. The second part of this work studies A* path planning algorithm combined with a proportional controller to make SLAM robot be able to follow planned paths until reaching the goal without collision along the way. Thirdly, the information gain- and minimized traveling cost-based exploration is studied in order to provide the robot the decision making capability for where to go next. In addition, map representations in forms of feature-based and occupancy grid map play important roles in reasoning the environment, which are also taken into consideration. Putting these techniques together, we introduce the whole framework for an indoor autonomous mobile robot and evaluate its effectiveness via simulations.

Department : Electrical Engineering	Student's Signature ....
Field of Study : Electrical Engineering	Advisor's Signature ....
Academic Year : 2014	

Acknowledgements

First and foremost, I would like to gratefully thank my advisor, Assistant Professor Manop Wongsaisuwan, for giving me true inspiration to conduct the research and clear guidance during the past five semesters. Without his generous helps and supports, I could not manage the thesis to this end. I am also immensely grateful to him for introducing me this research area in which afterwards I found very interested, for numerous discussions, critical comments and constructive suggestions. I am very fortunate to be his student.

My special thanks also go to the committee members, Professor David Banjerdpongchai and Assistant Professor Itthisek Nilkhamhang, who kindly agreed to be the thesis examiners and have provided me with plenty of invaluable comments as well as great encouragement.

I am deeply indebted the instructors of all the EE course works and seminars I enrolled in, especially the instructors from the Division of Control Systems. Their lectures and talks have taught me essential knowledge as well as relevant skills that are hugely beneficial to my current research and future career.

I am gratefully acknowledge the full scholarship of AUN/SEED-Net which has given me a great opportunity to experience high-quality graduate study, professional academic forum and fascinating life abroad. I would like to express my deep sense of gratitude to Chulalongkorn University for providing a systematic education and professional academic environment. In addition, the payment of tuition fee from International School of Engineering for my fifth semester is very much appreciated.

I must thank the students in the Control System Research Laboratory for their immense helps and warm friendship. Many big thanks to all of my lovely friends for being with me, being supportive right in time and for motivational talks we shared one another.

Lastly, deepest in my heart, I would like to thank my family for their endless love and unconditional supports. No words can fully describe what they have done for me throughout my life.

Contents

List of Figures

Figure Page

Figure Page

CHAPTER I

INTRODUCTION

1.1 Motivation

In recent years, there is an enormous attention to truly autonomous mobile robots, which are capable of mapping its current environment, localizing itself relative to the built map and controlling its motion in that environment. Once successfully doing so, robot is able to perform a lot of duties instead of human, such as surveillance, search and rescue, security, transportation, remote exploration, and cleaning. All of three capabilities, i.e. localization, mapping, and motion control or path planning are not independent, but mutually correlated, meaning that we cannot solve each of them separately but concurrently as one entire problem as depicted in Figure 1.1. Each coupling region, denoted by (I), (II), and (III), represents a classified problem in autonomous mobile robots and the common region (IV) indicates the state of *truly autonomous* of a mobile robot. In literature, there are actually numerous different approaches to coping with coupling problems and also the whole one. This thesis will primarily focus on key aspects to solve the integrated problem for an indoor mobile robot.

First of all, we are going to analyze further the characteristics of each coupling ones to see in more detail the whole picture.

Simultaneous Localization and Mapping, mostly known as SLAM problem has received a great deal of attention in both theory and practice since the 1980s, commenced by the seminal paper of Smith and Cheeseman [9]. It brings a solution to a mobile robot moving in a previously unknown place in order to incrementally build a map of that environment while at the same time specifies its

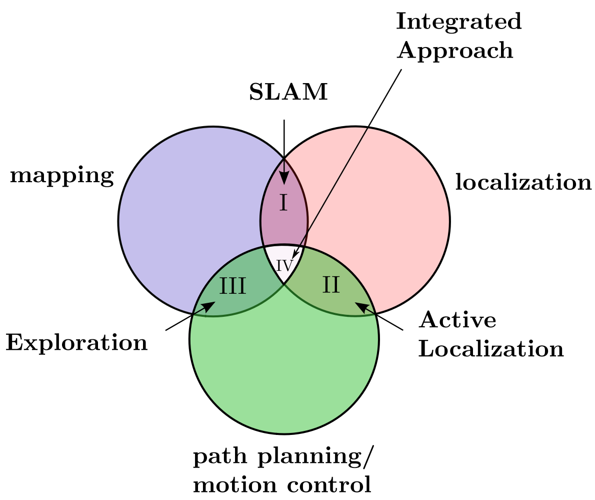


Figure 1.1: Classified tasks in autonomous mobile robotics [10]. The overlapping area (region IV) will be addressed in this work.

position in the map. Additionally, robot is asked to use on-board sensors only to implement its tasks. There are many approaches extensively considered in indoor [2], outdoor [3], underwater [5] and airborne [7] environments. *Active Localization* is another problem in which the map of the environment is provided. Robot then has to determine its pose in that map while performing the planned motion. On the other hand, *Exploration*, by some means given the robot pose, utilizes path planning to obtain an efficient map.

By going through the whole picture, we can realize more clearly that the integrated problem is very essential in the field of autonomous mobile robot. Its complete solution is still open to future research. This work makes an attempt to investigate a possible approach to the whole problem in order to answer these questions

- how to model the problem including sensor, robot motion and environment
- how to deal with unknown data, noise and uncertainties to solve the SLAM problem
- how to integrate path planning into SLAM to obtain a better map

1.2 Literature Review

1.2.1 History of SLAM Problem

The first brick of SLAM problem was constructed at the IEEE Robotics and Automation Conference held in San Francisco, California in 1986 [8]. At that time, researchers had just introduced probabilistic theory into both robotics and artificial intelligence. During the conference, a lot of time was spent on discussing about how to use estimation method to solve mapping and localization problem. Its participants, including Randall C. Smith and Peter Cheeseman, soon afterward published the important papers on this direction. They attempted to represent locational relationship of landmarks in the robot's vicinity and then show that there exists a correlation between them, which can be reduced its uncertainty by the use of some operations.

The landmark research was conducted shortly after that by Smith et. al [13]. This paper described an efficient method based on Kalman filter to acquire the estimate of robot pose and landmark position in case that robot is moving in the environment and taking observations of nearby landmarks. Importantly, these landmark estimates are all correlated with each other due to the common error in estimated vehicle location. This work proposed a solution to SLAM problem that opened many related researches on essential SLAM's aspects later on. Its conclusion emphasized on two respects. First, the correlations between landmarks are kept updated by repeatedly using Kalman filter. Second, the full state composed of robot state and landmark states is maintained and updated following each time of taking observation. However, one drawback of this method is that it carries a large state vector with the order of the number of landmarks in the map and furthermore its computational cost is as expensive as the square of that order [8]. In addition, this work did not take into account the convergence properties of the map or its steady-state behavior. Instead, the method was assumed that the estimated map errors would not converge but exhibit a random-walk behavior with unbounded error growth. For this reason, the following researches was going along with these assumptions that the correlations between landmarks would be eliminated and was only focusing on approximating the consistent map. And then the full-estimation theoretic SLAM problem seemed to be temporarily forgotten until the conceptual breakthrough came with the realization that the combined mapping and localization problem, once formulated as a single problem, was eventually convergent [14]. Above all, the crucial part of SLAM problem turned out to be the correlations between landmarks, which many researchers previously attempted to minimize. Surprisingly, the more these correlations increased, the better the solution.

SLAM problem afterward became an active area where both theory and application aspect have been developed by many research groups around the world, notably at the Massachusetts Institute of Technology, the Australian Centre for Field Robotics, University of Zaragoza in Spain. And also a lot of significant implementations have been carried out in various types of environment, which include indoor, outdoor, aerial and subsea. Besides, many conferences, such as International Symposium on Robotics Research (ISRR), IEEE International Conference on Robotics and Automation (ICRA), Intelligent Robots and Systems (IROS), and SLAM summer schools, held by KTH Stockholm in 2002, LAAS-CNRS Toulouse in 2004, and University of Oxford in 2006 attracted a huge number of researchers coming and bringing a variety of grand ideas into the field. In recent years, researchers have been working on improving computational efficiency and addressing vision-based SLAM in real applications.

1.2.2 Review of Key Literature

In this section, we will highlight key references that have the most important influence on this research's motivation. Following the review of the flow of SLAM problem, we pay attention on SLAM application, in other words, what robot is able to do after it completes SLAM task. In this connection, we focus on how to apply the map obtained from SLAM for navigation problem.

The authors in [11] proposed the same idea of SLAM and navigation with the use of A* and RRT path planning algorithm for navigation after the map is built by SLAM. In addition, a method of extracting both point-landmarks and line-landmarks are introduced by employing an omni-directional camera and laser range finder as the sensing system of the robot. However, it is appear that they combine SLAM with those path planning algorithms in a sequential manner, meaning that path planning is performed after the map is built from SLAM.

In [12], the adaptive method relies on a metric in terms of Fisher information that represents the sum of the areas of the error ellipses of the vehicle and feature estimates in the map. With the objective to maximize the robot's information about its position and all the landmarks' positions, a motion control for the robot at each time step can be chosen.

The work [31] introduced the frontier based method of robot exploration, which defines frontiers between explored and unexplored regions. The nearest frontier will be chosen to be the next destination for the robot. As reaching the goal, the robot perform a new scanning sweep to continue finding new frontiers. This work was then improved in [32] which took localization method into account. However, the tasks of exploration and localization were considered separately.

The integrated SLAM exploration strategy introduced in [10] uses the tight integration between every two elements of the tasks of localization, mapping, and motion control. This approach to exploration calls for a balanced evaluation of alternative motion actions from the point of view of information gain, localization quality, and navigation cost. These qualities will be described by their specific utility function and the total utility will then decide the next destination for the robot. In the same manner, the framework in [33] used Extended Kalman filter for robot localization and mapping while constructing an occupancy grid map for the environment in which the robot is exploring based on frontier method. Moreover, this work also concerns about the multi-robot exploration as well as dynamic A* path planning algorithm, equipping the robot a capability to deal with dynamic objects on the way to goal.

The extensive robotic mapping works in [22] have appeared to be the state-of-art techniques in SLAM and Exploration which effectively exploit particle filter and grid-based map. In addition, active loop closing and information gain are main objectives for the robot while exploring. The research also presented how to face with dynamic environment as well as multi-robot coordination.

Let us take the work in [34] to be a typical example of exploration-based outdoor autonomous mobile robot that was successfully manufactured and implemented. Therefore it covers many aspects of robotics, from vision, state estimation, exploration, planning, control and so on. Robot also owns behavior selection scheme in order to interact with people on the way. The obtained knowledge is then integrated into the on-line reasoning system in order to enhance its autonomy. Overall, this framework extensively exploits probabilistic robotics and Bayesian estimation field.

1.3 Thesis Objectives

1. Study SLAM solutions based on Extended Kalman filter (EKF), Unscented Kalman filter (UKF) and Particle filter (PF) and make a comparison between them.

2. Investigate the convergence and consistency of these SLAM solutions in order to guarantee the feasibility of the SLAM map's application.

3. Develop an integration between SLAM problem with path planning algorithms in order for a mobile robot to autonomously arrive at a prescribed destination.

4. Illustrate the usefulness of the developed method by carrying out MATLAB simulation.

1.4 Scope of Thesis

1. Investigate the most well-known SLAM solutions and make a comparison between them in terms of accuracy.
2. Study on A* path planning algorithm
3. Develop an integration between SLAM and the path planning algorithm.
4. Build a MATLAB simulation to test the proposed integration.

1.5 Methodology

1. Literature review on SLAM problem and existing techniques to solve this problem.
2. Literature review on path planning algorithms in order for robot to reach the desired destination without any collision on the way.
3. Build a simulation tool for SLAM problem in combination with proposed path planning algorithm.

1.6 Expected Outcomes

1. A simulation-based comparison between some existing SLAM solutions.
2. An integration between SLAM and path planning algorithm.
3. A simulation tool for SLAM problem combined with path planning.

1.7 Thesis Outline

The organization of the thesis is as follows. Chapter 2 presents basic background and concepts that will be used throughout this work. Chapter 3 investigates the SLAM problem with three algorithms, EKF-, UKF- and particle filter- based methods. Here we present one of the most important aspects in SLAM, which is loop closing, together with illustrative simulations. The comparative simulations between these solutions are also provided. Chapter 4 considers the occupancy grid map to represent the environment and to be used for navigation. It also takes into account how to evaluate benefit function based on information gain, traveling and steering cost to equip the robot the decision making capability of where to move next. Chapter 5 examines the whole framework by conducting simulations for the robot exploring the previously unknown area. Conclusions and future works are given in Chapter 6.

CHAPTER II

PROBLEM DESCRIPTION

This chapter provides a preliminaries on the configurations of the environment and robotic system. The material presented here will be essential for all the algorithms mentioned in the following chapters.

2.1 Introduction

Following the motivation aforementioned, we need to determine the techniques to be used for our robot. Here in this description chapter, an overview of the entire scheme is given briefly. Throughout the subsequent chapters, we will discover in a row these blocks of specialized knowledge in mobile robotics and then see the connections between them as well as the underlying reasons why we opt to these specializations.

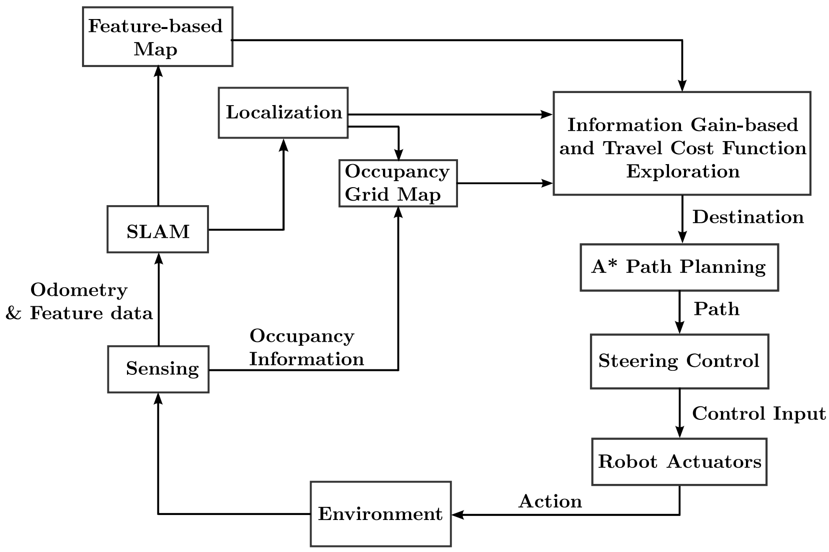


Figure 2.1: Entire reference scheme of mobile robot used in this research

Figure 2.1 depicts the entire reference scheme that we exploit in this research. To classify it in sub-groups, the Sense-Think-Act cycle [24] will be an appropriate representation, as shown in Figure 2.2. In this connection, *Sense* includes Sensing action using robotic sensors to perceive the current environment; *Think* consists of *i,* SLAM problem to concurrently specify robot's position and map, *ii,* a technique to explore the environment and *iii*, a path planning algorithm to generate a path reaching the desired goal; *Act* afterward produces reasonable motion controls to perform movements following the specified path.

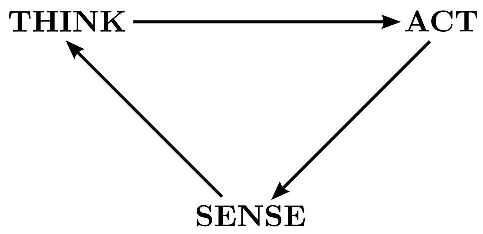


Figure 2.2: Sense - Think - Act cycle

Now with that proposed scheme for our mobile robot, we can somewhat claim it makes the robot *autonomous*. Before embarking upon the journey with each part of the scheme, let us define essential preliminaries.

2.2 Uncertainties in Robotic Operation

In any system, from human to machines, there always exist noises that we cannot neglect. These uncertainties result from, to name but a few, each physical operation of robot, i.e. sensing and moving. Depending on what type of sensor and its price, the measurement ability of our robot varies from large noise to high accuracy. The same applies for robot actuation, where noises arise from wear and tear or mechanical failure. In either case, autonomous robot is supposed to manage itself to handle these disturbances. Unfortunately, they are mostly unpredictable that it is unlikely to arrive at a closed form of noises. The most common way is using probabilistic models to represent the presence of noisy subjects in the system. In this way, we have to take some moments such as mean and covariance of our state into account and overall the problems with robots turn to be recursive estimation problems, which will play the core role in our algorithms presented in this thesis.

We will be going on with notations and concepts in each process step of our robot.

2.3 Robot Perception

Robot perception is a way robot interacts with its environment using sensors to extract useful information. This task plays a vital role in achieving the necessary accuracy of robot actions. If for some reason a sensory device becomes inaccurate while working, the whole system can afterward act in a divergent manner, e.g. move uncontrollably far from the operating area. Therefore, robotic vision has received considerable attention over the past decades and consequently has developed to an advanced level, where machine with affordable cost can autonomously perform dangerous or tedious tasks instead of human.

Today there is a wide range of sensory devices in the robotic field. Generally, these sensors are classified into two groups based on their own functions, which are *proprioceptive sensors* and *exteroceptive sensors*. The first group is used to measure values related to internal actions of robot such as wheel revolution, load weight or battery voltage; whereas the latter is to take measurement

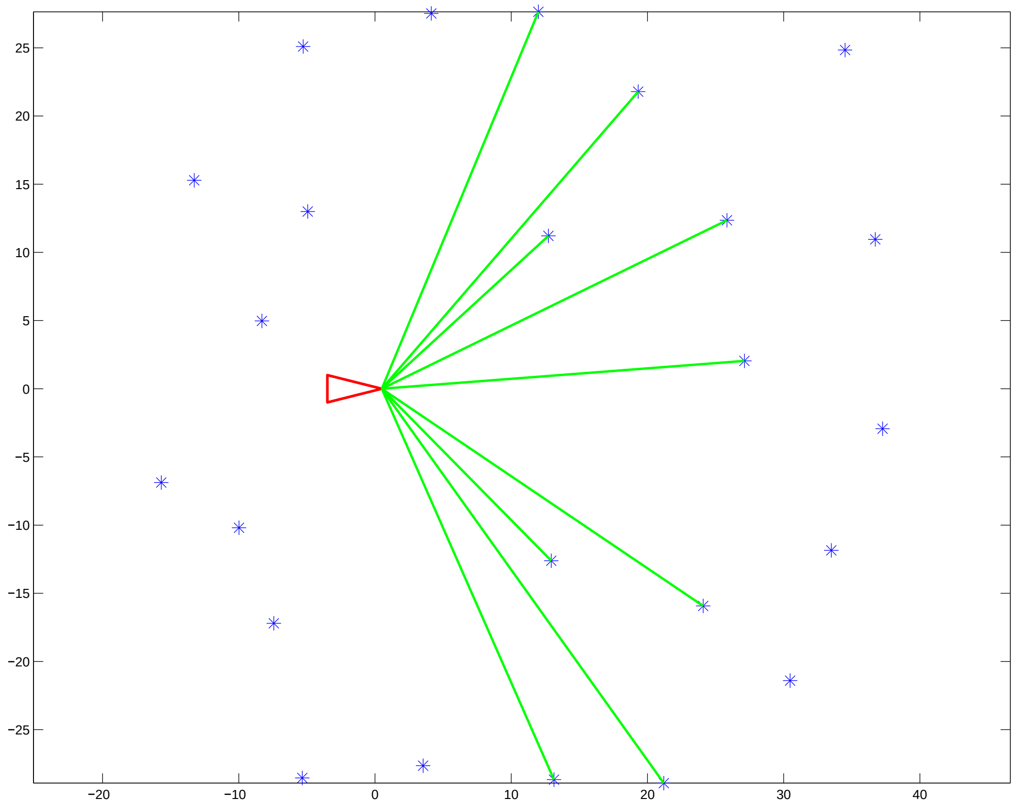


Figure 2.3: A typical setup in a simulated environment, e.g. MATLAB. Robot (triangle) is observing landmarks nearby (stars) using a model of laser range finder where straight lines illustrate its observing actions.

to external objects existing nearby such as distance to the wall in front or detecting the presence of obstacles. We now need to choose appropriate sensors to fulfill particular purposes, usually pertaining to technical objectives and economic constraints.

In our scenario, the robot basically needs not only to handle its motion but to perceive its environment. This is like a person getting lost his way. He then has to be aware of how far or how fast he has been moving while utilizing his vision to observe surrounding entities. Therefore, we need to use both types of sensors mentioned above, two internal and one external, to acquire necessary information. The first one, Encoder, is the most popular device to monitor the robot motion through counting the revolution of the rear wheels by fitting it at the drive shaft. The second internal sensor is Linear Variable Differential Transformer, attached in the steering rack to give a means of reading robot heading. The other sensor, Laser Range finder, plays a role in taking measurements to the things nearby. This is a very common sensor used to measure the distance between itself and an object by projecting a beam into the target and calculating the time taken for the beam to be bounced off the target's surface and return to the instrument. This action is visualized via straight lines in Figure 2.3 where a robot (triangle) is observing landmarks nearby (stars).

2.4 System Configuration

The robot in this thesis is confined to work in planar environments. The pose x of robot thus comprises its location (x_r, y_r) and heading ϕ relative to a global coordinate system. Therefore we have $x = [x_r, y_r, \phi]^T$.

Objects in robot's environment are assumed to be static. In other words, under the scope of this research we have not yet dealt with dynamic environments. In fact, we need to exploit *landmarks*, which are distinct features placed in the environment so that the robot can recognize with high reliability. Additionally, these landmarks will contribute into the system state via the form of their locations relative to global coordinate frame and all landmarks' state that robot has observed at a certain point of time will create the *map* m of the environment.

$$m = [m_1, m_2, \ldots, m_{N_t}]^T = [x_1, y_1, x_2, y_2, \ldots, x_{N_t}, y_{N_t}]^T \tag{2.1}$$

We now can define the *complete system state* X in equation (2.2), which includes the robot pose and the map m

$$X = \begin{bmatrix} x \\ m \end{bmatrix} \tag{2.2}$$

As discussed in Section 2.3, the robot obtains information about its own motion and current environment by using on-board sensors. In this way, the robot and the environment have two interactions

- Control actions: robot is forced to move in the environment which lead to a change in the system state. The control input is denoted by $u = [V, \gamma]^T$, including velocity and steering angle as aforementioned.

- Measurement (aka Observation) actions: Robot uses sensors to acquire measurement values of objects in its environment and increase information it possesses. Each valid measurement $z = [r, \theta]^T$ consists of range r and bearing θ, which are the distance and relative angle between the robot and a landmark.

We also note that the way we often separate control action and measurement due to the nature of filtering algorithms that are discussed in the subsequent chapters. In fact, both actions are taken place at the same time.

2.5 Process Model

In this thesis we are working with a car-like mobile robot in indoor environment, consisting of two fixed standard wheels at the back and two steered standard wheels in front. Rear wheels are actuated by one motor and the steering mechanism of front wheels requires another motor. It is common to model this four-wheeled car-like robot by a bicycle model, which includes one fixed rear

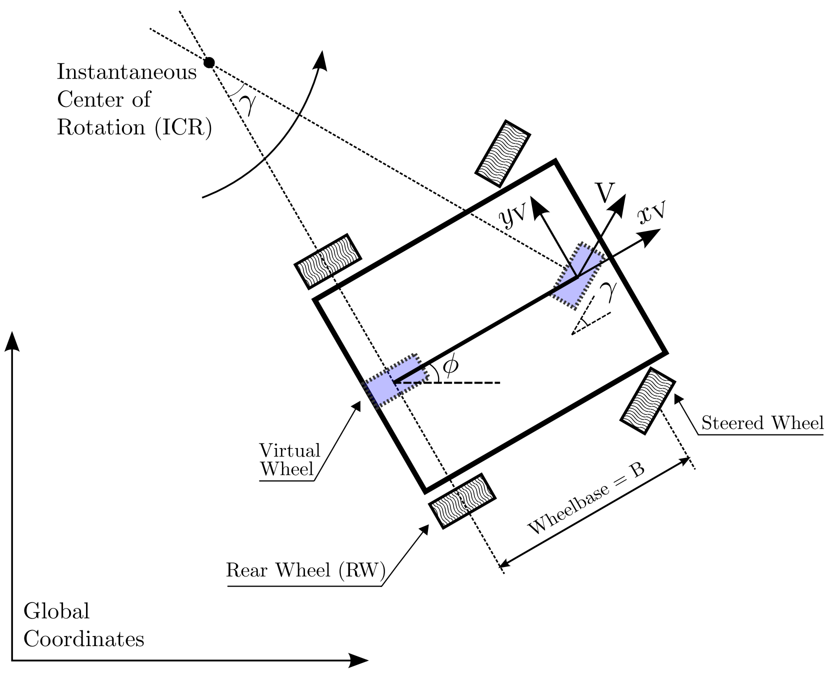


Figure 2.4: The robotic car rotates about its instantaneous center of rotation. The bicycle approximation is depicted in between left and right wheels.

wheel and one front wheel able to rotate about its vertical axis to steer the vehicle, as depicted in Figure 2.4.

The process model investigated here is based on the robot's kinematic model under some assumptions [16]. First, the robot motion is restricted to a two-dimension plane. The second one is that the said plane is fixed into the Earth ground so that we can neglect the effects caused by the Earth's motion. The next assumption, mostly known as nonholonomic constraint, is no slip occurring between all the tyres and the ground so that the robot cannot move laterally along its axle. By these assumptions, the system complexity and its motion characteristics have been relaxed to be:

- The degree of freedom (DOF) of the whole system decreases to two. The initial DOF without constraints includes the drive velocities of all four wheels together with the steer angle of the front wheels, whereas the constrained system has the translational velocity of the whole vehicle and the steering angle, then resulting in 2 DOF.

- The whole system will rotate about the instantaneous center of rotation (ICR), determined by the intersection of two straight lines perpendicular to the front and rear wheels respectively. The whole system now can be modeled as a bicycle through the representation of two virtual wheels as depicted in Figure 2.4.

For these reasons, we opt to use the translational velocity of the robot V and the steering angle γ of the virtual front wheel as the control input without loss of generality.

Let B be the wheelbase of the vehicle, which is the distance between the front and rear wheels, then, based on these assumptions and configurations, the process model in continuous time is written

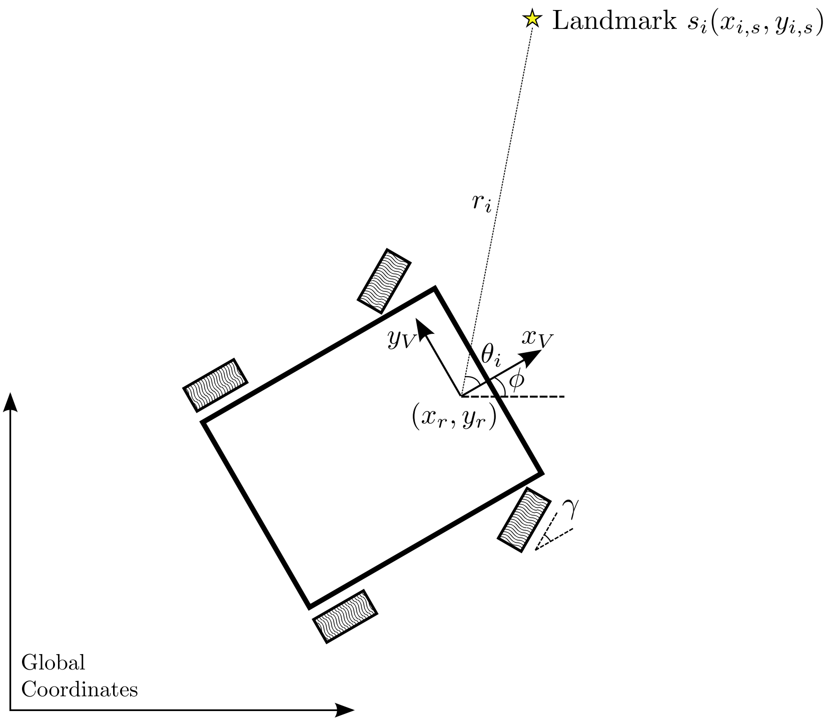


Figure 2.5: Robot is observing a nearby landmark (star) and obtains measurement value $z_i = [r_i, \theta_i]^T$.

as follows

$$\begin{aligned}\dot{x}_r &= V\cos(\phi+\gamma)\\ \dot{y}_r &= V\sin(\phi+\gamma)\\ \dot{\phi} &= \frac{V\sin(\gamma)}{B}\end{aligned} \tag{2.3}$$

This model needs to be discretized in order to be applicable in computer-based execution. Within a very small period of time Δt, assume that the control input, V and γ, is stationary. We now can perform discretization to obtain the robot pose at time step k

$$\begin{aligned}x_{k+1,r} &= x_{k,r} + V_k\Delta t\cos(\phi_k+\gamma_k)\\ y_{k+1,r} &= y_{k,r} + V_k\Delta t\sin(\phi_k+\gamma_k)\\ \phi_{k+1} &= \phi_k + \frac{V_k\Delta t\sin(\gamma_k)}{B}\end{aligned} \tag{2.4}$$

2.6 Measurement Model

A model of robot observing a landmark is depicted as Figure 2.5. For the sake of simplicity, we assume that the sensor is attached at the center of the front axle without loss of generality. In fact, at one time sensor is able to measure many landmarks depending on the density of landmarks in the environment and the maximum length capability of the laser sensor. The range and bearing acquired from sensing *i*-th landmark are computed as

$$\begin{bmatrix} r_i \\ \theta_i \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i-x_r)^2+(y_i-y_r)^2} \\ \arctan(\frac{y_i-y_r}{x_i-x_r}) - \phi \end{bmatrix} \tag{2.5}$$

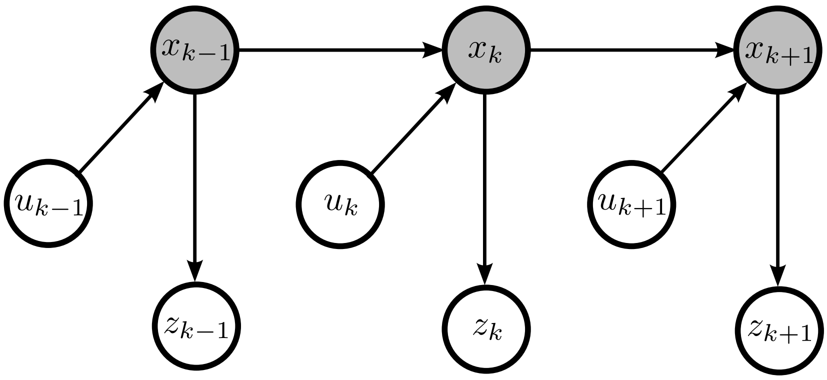


Figure 2.6: Evolution of state, control input and measurement in this robotic system [36].

We also note that noises indeed affect both process and measurement model, which will arise later when we discuss the solution to SLAM problem.

2.7 Markov Assumptions and Resulting Probabilistic Models

We now dive into probabilistic concepts for our robotic system by using probability theory. The evolution of controls, states and measurements in the perturbed system are reflected by probability distribution functions, as illustrated in Figure 2.6 where shaded nodes are what we want to estimate and the others are given.

Let us first explore two Markov assumptions which will be presented as follows. The first Markov property states that our state is complete. This means if we know the previous state x_{k-1} then past states $x_{0:k-2}$, measurements $z_{1:k-1}$ and control inputs $u_{0:k-2}$ provide no information for the current state x_k. In addition, the previous state x_{k-1} is conditionally independent of future states $x_{k+1:k+\tau}, \tau \geq 1$ given the current state x_k. Therefore we obtain equations for probability distributions as follows

$$p(x_k|x_{0:k-1}, z_{1:k-1}, u_{0:k-1}) = p(x_k|x_{k-1}, u_{k-1}) \tag{2.6}$$

$$p(x_{k-1}|x_{k:k+\tau}, z_{k:k+\tau}) = p(x_{k-1}|x_k) \tag{2.7}$$

The second assumption accounts for the independence of the current measurement z_k conditioned on the current state

$$p(z_k|x_{0:k}, z_{1:k-1}) = p(z_k|x_k) \tag{2.8}$$

These main resulting models, $p(x_k|x_{k-1}, u_{k-1})$ and $p(z_k|x_k)$, in the world of uncertainties corrupted by noises indeed interpret probabilistic models of the robot's state transition and measurement processes. The probability distribution $p(x_k|x_{k-1}, u_{k-1})$ shows how the state evolves when robot receives a new control input u_{k-1} at the state x_{k-1}. Whereas the measurement probability $p(z_k|x_k)$ stochastically manifests the resulting measurement z_k at the current state x_k.

2.8 Conclusions

This chapter highlights the key background and, especially, the whole framework description that will be discussed in more detail in the following chapters. Two basic models of the robotic system have been built together with their assumptions. We also introduce the nature of inevitable uncertainty existing in both models, which then is governed by probabilistic theory.

CHAPTER III

SIMULTANEOUS LOCALIZATION AND MAPPING

3.1 Introduction

This section investigates the problem of Simultaneous Localization and Mapping, mostly known as SLAM. This addresses both localization and mapping for mobile robot at the same time. Localization determines robot pose in a given map. Mapping, on the other hand, is going to construct a map of the environment in which a robot with known pose is maneuvering. In autonomous mobile robotics, SLAM has been a major research topic attracting numerous extensive works for more than two decades and also famous as a big motivation in approaching to probabilistic robotics. Since this is a large problem, SLAM community has developed many insightful research directions. However, there are still open challenges to continue on the journey [19].

In general, to re-emphasize, a mobile robot equipped with sensors to perceive the ambient objects in its working area inevitably encounters noisy measurements. It also receives uncertainties from its control inputs as well as dynamic environment. Unfortunately, these quantities cannot be governed by concrete formulation. Instead, we treat these uncertainties like random variables by using probability theory in a systematic and efficient manner.

There are two main forms of SLAM, which are known as the online SLAM and the full SLAM problem. The first one attempts to estimate the posterior over the current pose and map

$$p(x_k, m|z_{1:k}, u_{0:k-1}) \tag{3.1}$$

The full SLAM problem aims to estimate the entire poses of the robot together with the map

$$p(x_{0:k}, m|z_{1:k}, u_{0:k-1}) \tag{3.2}$$

Because of the high dimensionality of the full SLAM, especially when the environment owns a huge number of landmarks, in most cases people try to come up with the online SLAM problem and approximate the full problem if needed. In this work, we will investigate both two problems as a base for the further research.

The following parts present key solutions to SLAM, comprising the use of Gaussian filters and Nonparametric filter. Along the way, we will see how these approaches can obtain relatively good estimates for robot pose and map among ambiguous inputs. At the end, various simulations show effectiveness of these techniques as well as the most important properties in this problem.

3.2 EKF-SLAM

EKF-SLAM provides a solution to this problem based on Extended Kalman filter [25], which linearizes the system models in order to apply the well-known Kalman filter. Historically speaking, this technique is the earliest and most influential in the field.

As an estimation problem, importantly, EKF-SLAM take into account the first two moments of probability distributions, recursively estimating the mean of the system state and its associated covariance.

The EKF-SLAM maintains a feature-based map which governs each landmark via its Cartesian location. In addition, each landmark will be treated as a point in the coordinate frame.

We now start the main procedure of SLAM problem that is aligned with the Bayesian filtering manner in association with other specific techniques.

3.2.1 Prediction

The process model of the state will be considered separately corresponding to its structure as in Equation (2.2), taken at time step k when the robot receives a new control input $u_k = [V_k, \gamma_k]^T$ to achieve the next state at $k+1$.

$$x_{k+1} = \begin{bmatrix} x_{r,k+1} \\ y_{r,k+1} \\ \phi_{k+1} \end{bmatrix} = \begin{bmatrix} x_{r,k} + (V_k + \delta V_k)\Delta t \cos[\phi_k + (\gamma_k + \delta\gamma_k)] \\ y_{r,k} + (V_k + \delta V_k)\Delta t \sin[\phi_k + (\gamma_k + \delta\gamma_k)] \\ \phi_k + \frac{(V_k + \delta V_k)\Delta t \sin(\gamma_k + \delta\gamma_k)}{B} \end{bmatrix} \tag{3.3}$$

$$m_{k+1} = m_k \tag{3.4}$$

There always exists disturbance in sensor data, which is assumed to be additive zero-mean Gaussian in this model, denoted by $v_k = [\delta V_k, \delta\gamma_k]^T$. Thus, equation (3.3) is necessarily extended from equation 2.4. As a simplest formulation of disturbance, each element of v_k is added directly into each control input. Finally, the process model can be written in general form as

$$X_{k+1} = f(X_k, u_k, v_k) \tag{3.5}$$

Let $\hat{X}_{k|k}$ and $P_{k|k}$ denote the state estimate at time k and its associated covariance matrix. The EKF linearizes the state equation using Taylor series expansion evaluated at $\hat{X}_{k|k}$. The prediction step of EKF-SLAM is then given by

$$\hat{X}_{k+1|k} = f(\hat{X}_{k|k}, u_k, 0) \tag{3.6}$$

$$P_{k+1|k} = \nabla f_X P_{k|k} \nabla f_X^T + \nabla f_u Q_k \nabla f_u^T \tag{3.7}$$

where Q_k is the 2-by-2 covariance matrix of the control input noise and ∇f_X, ∇f_u are Jacobian matrices given by

$$
\begin{aligned}
\nabla f_X &= \frac{\partial f}{\partial X}(\hat{X}_{k|k}, u_k) = \begin{bmatrix} \nabla g_{X_r} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \\
\nabla g_x &= \frac{\partial g}{\partial x}(\hat{x}_{k|k}, u_k) = \begin{bmatrix} 1 & 0 & -V_k \Delta t \sin(\hat{\phi}_{k|k} + \gamma_k) \\ 0 & 1 & V_k \Delta t \cos(\hat{\phi}_{k|k} + \gamma_k) \\ 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{3.8}
$$

$$
\begin{aligned}
\nabla f_u &= \frac{\partial f}{\partial u}(\hat{X}_{k|k}, u_k) = \begin{bmatrix} \nabla g_u & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\
\nabla g_u &= \frac{\partial g}{\partial u}(\hat{X}_{k|k}, u_k) = \begin{bmatrix} \Delta t \cos(\phi_{k|k} + \gamma_k) & -V_k \Delta t \sin(\phi_{k|k} + \gamma_k) \\ \Delta t \sin(\phi_{k|k} + \gamma_k) & V_k \Delta t \cos(\phi_{k|k} + \gamma_k) \end{bmatrix}
\end{aligned}
\tag{3.9}
$$

where **I** and **0** are respectively identity and zero matrices with the corresponding size.

3.2.2 Measurement Update

We assume that robot receives a control input before observing landmarks. At time step $k+1$ robot takes a new measurement to the i-th landmark. The measurement model corrupted by noise is given as follows

$$
\begin{bmatrix} r_{i,k+1} \\ \theta_{i,k+1} \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i - x_{r,k+1})^2 + (y_i - y_{r,k+1})^2} \\ \arctan(\frac{y_i - y_{r,k+1}}{x_i - x_{r,k+1}}) - \phi_{k+1} \end{bmatrix} + \begin{bmatrix} \delta r_{i,k+1} \\ \delta \theta_{i,k+1} \end{bmatrix} \tag{3.10}
$$

This model can be rewritten in the general form

$$
z_{i,k+1} = \begin{bmatrix} r_{i,k+1} \\ \theta_{i,k+1} \end{bmatrix} = h_i(X_{k+1}, m_i) + w_{i,k+1} \tag{3.11}
$$

where the noise w_i is assumed to be Gaussian with zero mean and covariance matrix R.

Innovation term is the difference between the true and the predicted measurement, together with its associated covariance, which are both given by

$$
\nu_{k+1} = z_{i,k+1} - h_i(\hat{X}_{k+1|k}, m_i) \tag{3.12}
$$

$$
S_{k+1} = \nabla h_{i,k+1} P_{k+1|k} \nabla h_{i,k+1}^T + R_{k+1} \tag{3.13}
$$

The estimate of the state vector and its associated covariance can be updated as follows

$$
\hat{X}_{k+1|k+1} = \hat{X}_{k+1|k} + K_{k+1} \nu_{k+1} \tag{3.14}
$$

$$
P_{k+1|k+1} = (\mathbf{I} - K_{k+1} \nabla h_{i,k+1}) P_{k+1|k} \tag{3.15}
$$

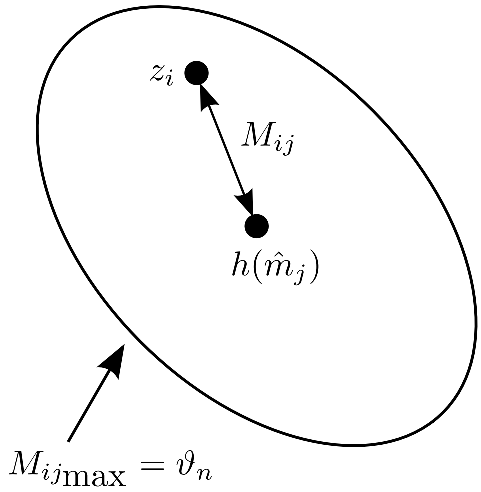


Figure 3.1: The validation gate is defined by ϑ_n, which is formed a gating ellipse. Any measurement z_i that falls inside this gate will be considered as a valid measurement associated to the predicted measurement $h(\hat{m}_j)$

where K_{k+1} is the Kalman gain which determines the influence of the innovation on the updated estimate given by

$$K_{k+1} = P_{k+1|k} \nabla h_{i,k+1}^T S_{k+1}^{-1} \tag{3.16}$$

and $\nabla h_{i,k+1}$ is a Jacobian matrix of the measurement model with respect to the state and obtained by

$$\nabla h_{i,k+1} = \frac{\partial h_i}{\partial X_{k+1}}\bigg|_{\hat{X}_{k+1|k}} = \begin{bmatrix} -\frac{\Delta x}{d} & -\frac{\Delta y}{d} & 0 & \frac{\Delta x}{d} & \frac{\Delta y}{d} \\ \frac{\Delta y}{d^2} & -\frac{\Delta x}{d^2} & -1 & -\frac{\Delta y}{d^2} & \frac{\Delta x}{d^2} \end{bmatrix} \tag{3.17}$$

$$\begin{aligned} \Delta x &= x_{i,m} - x_{r,k+1|k} \\ \Delta y &= y_{i,m} - y_{r,k+1|k} \\ d &= \sqrt{\Delta x^2 + \Delta y^2} \end{aligned} \tag{3.18}$$

We note that, in Equation (3.17), all the columns of landmarks that are not currently observed are neglected.

3.2.3 Data Association

It is often difficult to state which measurement corresponding to which landmark since, for instance, the similar characteristics of landmarks in the environment. Data association therefore addresses the problem of matching a measurement to its correct landmark, which plays a very crucial role in the SLAM implementations. In literature, there exist several algorithms as well as their extensions to address this problem. We will first come up with the most common technique which has been used for our scenario, namely *validation gating* [3]. This technique defines a validation region for a predicted measurement $h(\hat{m}_j)$ of landmark m_j in order to find a valid measurement based on the *normalized innovation squared* (NIS), and also termed as *Mahalanobis distance*. Recall the innovation term in equation (3.12), $\nu_{ij} = z_i - h(\hat{m}_j)$ and its corresponding covariance in equation (3.13),

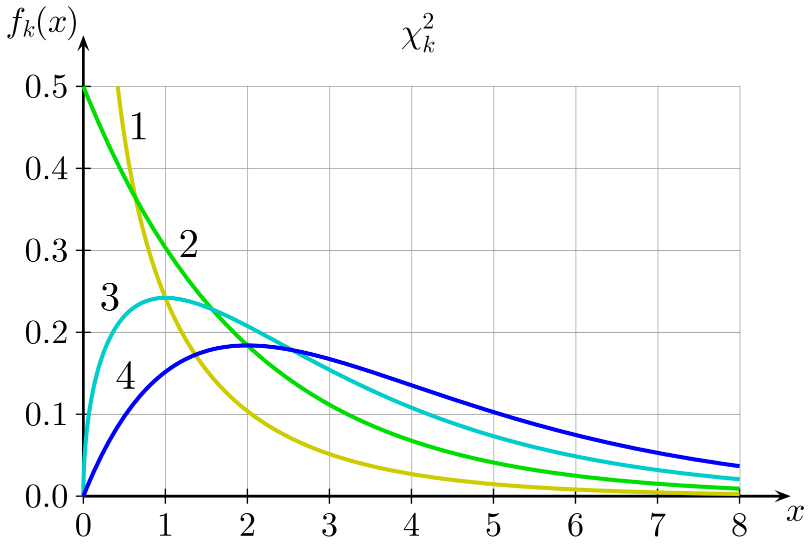


Figure 3.2: χ^2 distribution with 1, 2, 3 and 4 degrees of freedom

S_{ij}, then the NIS is stated as follows

$$M_{ij} = \nu_{ij}^T S_{ij}^{-1} \nu_{ij} \tag{3.19}$$

The validation gate is employed as the maximum threshold ϑ_n for the NIS, given by

$$M_{ij} < \vartheta_n \tag{3.20}$$

where n is the dimension of vector ν_{ij}. Any measurement which satisfies the condition (3.20) will be a valid measurement as illustrated in Figure 3.1.

In order to obtain the value of validate gate, we exploit the χ^2 (chi-squared) distribution whose shape changes according to the degree of freedom, or in this specified case, the dimension of n of the innovation vector ν, as shown in Figure 3.2. The integral of χ^2 distribution from 0 to the chosen threshold ϑ_n gives the probability of how likely the observation z_i associates with the target m_j.

3.2.4 Landmark Augmentation

Once the robot discovers a new landmark which satisfies to be incorporated into the whole state vector, the landmark augmentation will be taken place. As a result, the size of the state vector and its covariance will increase considerably. First the state vector and covariance matrix are augmented as follows [3]

$$\hat{X}_{\text{aug}} = \begin{bmatrix} \hat{X} \\ z \end{bmatrix} \tag{3.21}$$

$$P_{\text{aug}} = \begin{bmatrix} P_{xx} & P_{xm} & \mathbf{0} \\ P_{xm}^T & P_{mm} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & R \end{bmatrix} \tag{3.22}$$

A transformation to convert the polar measurement z into the global Cartesian location is given by

$$g(x,z) = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} x_r + r_i \cos(\phi + \theta_i) \\ y_r + r_i \sin(\phi + \theta_i) \end{bmatrix} \tag{3.23}$$

The augmented state can then be corrected by linearizing function $\mathbf{f}$ using Taylor series as follows

$$\hat{X}^{+} = \mathbf{f}(\hat{X}_{\text{aug}}) = \begin{bmatrix} \hat{X} \\ g(\hat{x}, z) \end{bmatrix} \tag{3.24}$$

$$P^{+} = \nabla\mathbf{f}_{X_{\text{aug}}} P_{\text{aug}} \nabla\mathbf{f}^{T}_{X_{\text{aug}}} \tag{3.25}$$

where the Jacobian $\nabla\mathbf{f}_{X_{\text{aug}}}$ is given by

$$\nabla\mathbf{f}_{X_{\text{aug}}} = \left.\frac{\partial\mathbf{f}}{\partial X_{\text{aug}}}\right|_{\hat{X}_{\text{aug}}} \tag{3.26}$$

3.2.5 EKF-SLAM: Simulation Results

We first introduce the environment settings that the simulation test is conducted, as depicted in Figure 3.3. It is necessary to previously define

- A global coordinate frame
- Landmarks' Cartesian locations in the environment as drawn by many stars (*).
- Initial pose at $[0, 0, \pi/2]^T$ and its associated covariance is $P = \mathbf{0}$.
- Pure SLAM is passive in the sense of robot motion. We need to provide a set of waypoints to guide the robot where to go. In practice, for example, to implement SLAM for a car in outdoor environment a person will manually drive the car around. The car controller performing SLAM is then able to tell us where it is and also returns the map of the environment.

Before we go, also note that we use the solid line to depict the estimated path and dashed line to the true path of the robot. Ellipsoid drawn in each object based on its covariance shows the uncertainty of its estimate. Now let us get the car moved and see through characteristics and results of EKF-SLAM.

One of the most important property of SLAM is *loop closing*, where the robot re-visit and recognize the place it has passed by before. From starting point to the moment before the loop, state error increase according to the length of the loop, as we can see in Figure 3.4. Right after closing the loop, uncertainties of both robot's pose and landmarks' locations collapse as in Figure 3.6. The same thing happens when the robot runs to another loop towards the letters R and L shown in Figure 3.7 and 3.8, uncertainties keep increasing along with the run until closing the loop, which abruptly shrinks all the ellipses.

Our remarks are confirmed when actual errors in robot pose estimates in x and y axis are plotted by solid lines in Figure 3.9 together with the $\pm 2\sigma$ (95%) confidence bounds (dotted lines). Along the way, there are two loop closures where exist sudden changes of error towards zeros.

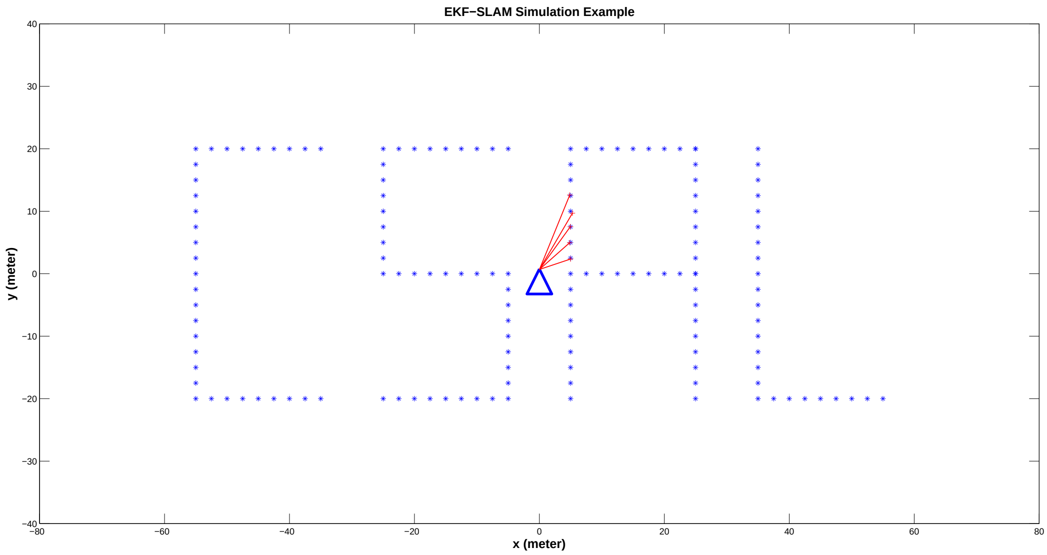


Figure 3.3: SLAM simulated environment setup. Landmarks are many small stars (*). Robot is drawn by the triangle, which has initial pose at $[0, 0, \pi/2]^T$. Straight lines originated from robot show observation actions of robot's laser sensor.

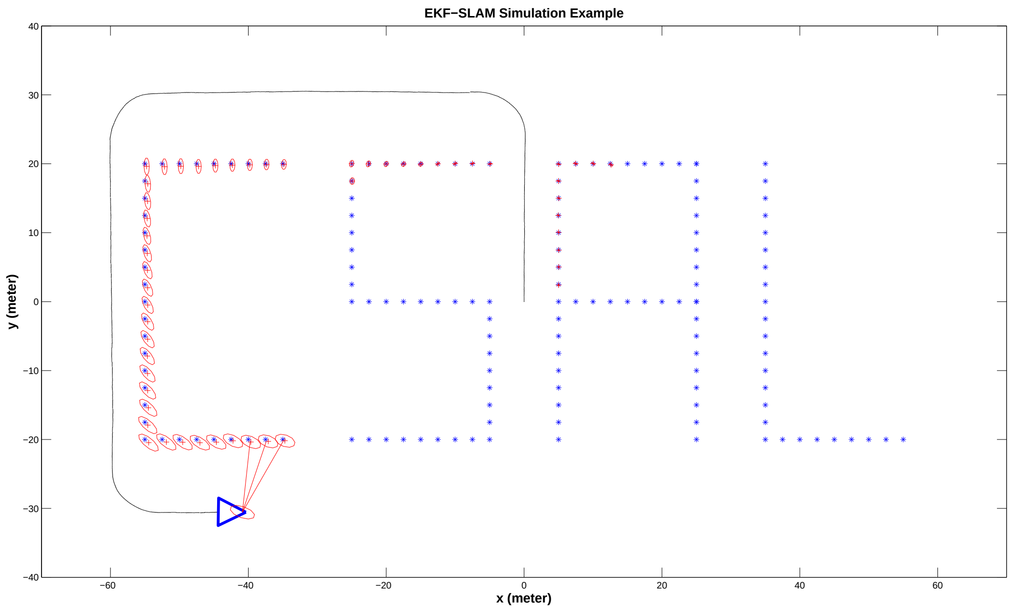


Figure 3.4: Robot starts to move and the uncertainties are increasing as illustrated by covariance ellipses on the observed landmarks and the robot.

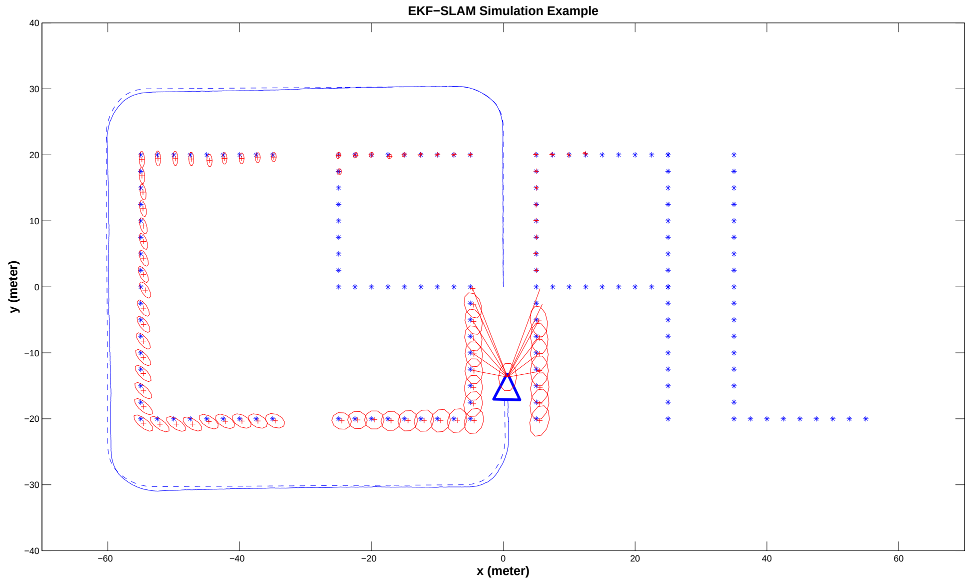


Figure 3.5: Robot continues moving and uncertainties continue going up. The later recognized landmarks have bigger uncertainties. As can be noticed, robot is about to close its first loop.

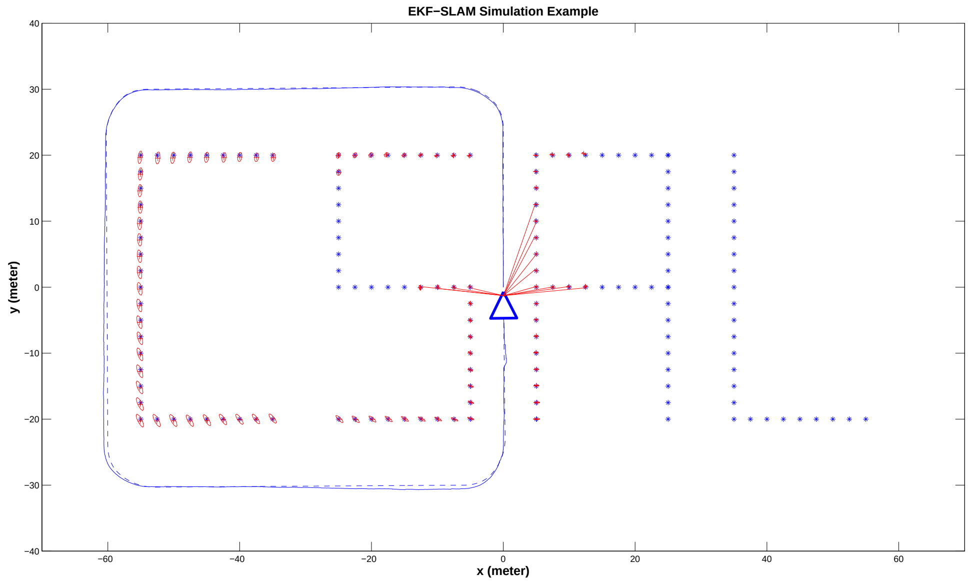


Figure 3.6: After closing the loop, all uncertainties suddenly shrink. Robot continues running to the second loop.

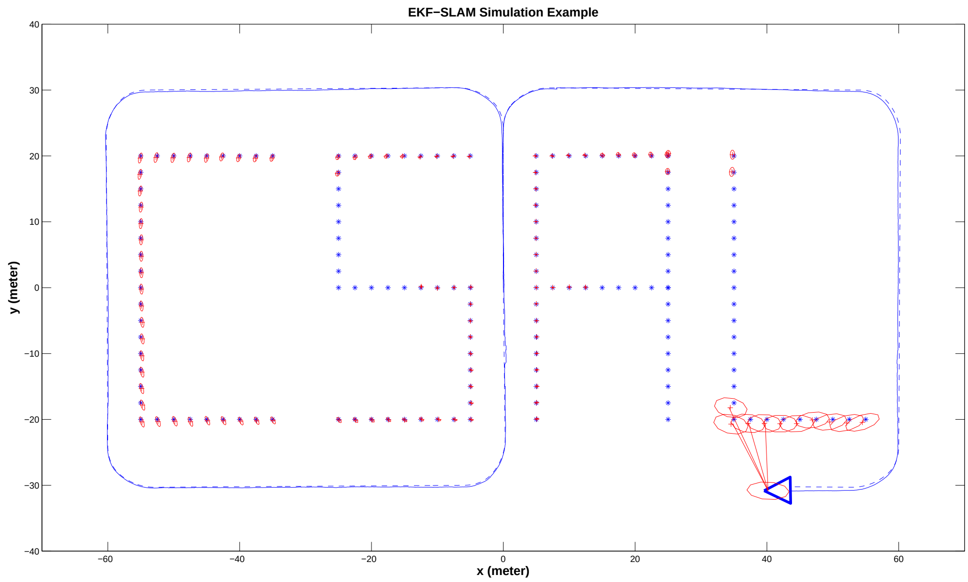


Figure 3.7: Before closing the second loop. The same results obtained with increasing uncertainties.

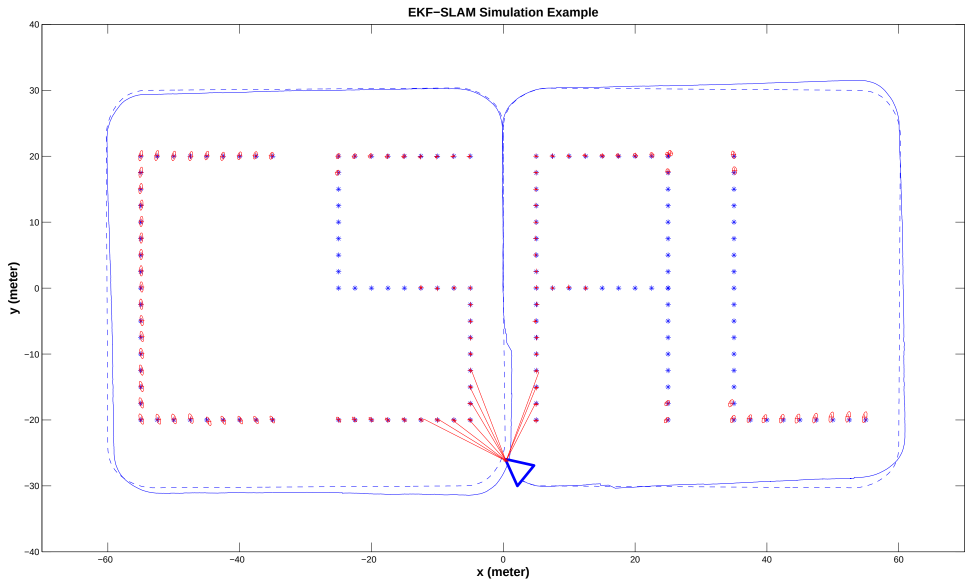
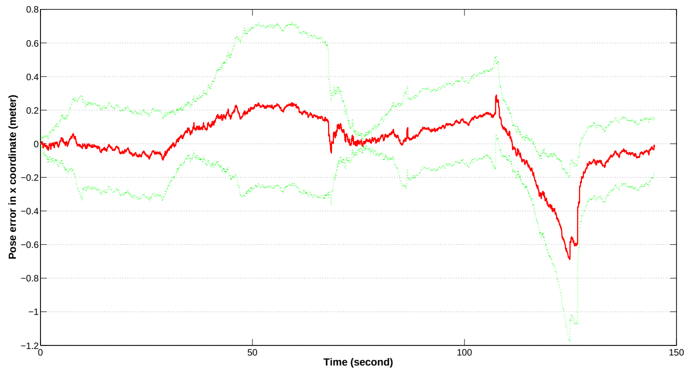
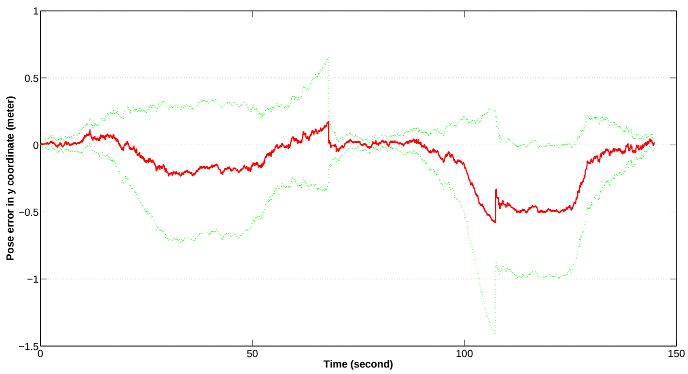


Figure 3.8: After closing the loop, all uncertainties collapse, which is shown by the sudden change in size of all ellipses explored during the second loop.



(a) Pose error in x-axis



(b) Pose error in y-axis

Figure 3.9: Actual error in robot pose estimate in x and y axis (solid lines) together with two 95% confidence bounds (dotted lines) derived from the estimated covariance

3.3 UKF-SLAM

Another extension of the Kalman filter, Unscented Kalman filter (UKF) [26], which utilizes Unscented transformation (UT) resulting the estimation more accurate than the EKF and at the same complexity. A number of the UKF approaches to SLAM problem have also appeared in the literature [27, 28]. Nevertheless the UKF is an approximation which can also lead to inconsistency.

3.3.1 Initialization

Start with storing the state vector and the noises into one augmented state vector as follows

$$X_k^a = \begin{bmatrix} X_k^T & v_k^T & w_k^T \end{bmatrix}^T \tag{3.27}$$

$$P_k^a = \begin{bmatrix} P_k & 0 & 0 \\ 0 & Q_k & 0 \\ 0 & 0 & R_k \end{bmatrix} \tag{3.28}$$

Perform the Unscented Transformation (UT) to obtain a set of so-called sigma points and accompanying weights

$$\chi_{k-1}^a = \begin{bmatrix} X_{k-1}^a & X_{k-1}^a + \eta\sqrt{P_{k-1}^a} & X_{k-1}^a - \eta\sqrt{P_{k-1}^a} \end{bmatrix} \tag{3.29}$$

$$\begin{aligned} W_0^{(m)} &= \frac{\lambda}{L+\lambda} \\ W_0^{(c)} &= \frac{\lambda}{L+\lambda} + 1 - \alpha^2 + \beta \\ W_i^{(m)} &= W_i^{(c)} = \frac{1}{2(L+\lambda)}, \quad i = 1, \ldots, 2L. \end{aligned} \tag{3.30}$$

where $\eta = \sqrt{L+\lambda}$; $\lambda = \alpha^2(L+\kappa) - L$ is a scaling parameter; L is the dimension of X_k^a; κ is a scaling parameter which is usually set to be $3 - L$; α is a positive scaling parameter which determines the spread of the sigma points around $\hat{X}^a$, usually set to be $1 \geq \alpha \geq 10^{-4}$; β is a constant used to incorporate prior knowledge of the distribution of X^a. $\beta = 2$ is optimal in case of Gaussian distributions [29]; $\sqrt{P^a}$ gives matrix square root of P^a where lower-triangular Cholesky factorization is the most widely used method.

3.3.2 Time Update

All the sigma points are propagated through the motion model $f(.)$ of robot

$$\chi_{k|k-1}^X = f(\chi_{k-1}^X, u_{k-1}, \chi_{k-1}^w) \tag{3.31}$$

The mean and covariance of the predicted state is calculated as follows

$$\hat{X}_{k|k-1} = \sum_{i=0}^{2L} W_i^{(m)} \chi_{i,k|k-1}^X \tag{3.32}$$

$$P_{k|k-1} = \sum_{i=0}^{2L} W_i^{(c)} (\chi_{i,k|k-1}^X - \hat{X}_{k|k-1})(\chi_{i,k|k-1}^X - \hat{X}_{k|k-1})^T \tag{3.33}$$

All the sigma points are propagated through the observation model $h(.)$ of robot

$$\mathcal{Z}_{k|k-1} = h(\chi_{k|k-1}^X, \chi_{k-1}^v) \tag{3.34}$$

The mean of the predicted observation is then computed as shown below

$$\hat{z}_{k|k-1} = \sum_{i=0}^{2L} W_i^{(m)} \mathcal{Z}_{i,k|k-1} \tag{3.35}$$

3.3.3 Measurement Update

Compute the covariance of the predicted observation and the cross-covariance of the predicted state and the predicted observation

$$P_{\tilde{z}_k \tilde{z}_k} = \sum_{i=0}^{2L} W_i^{(c)} (\mathcal{Z}_{i,k|k-1} - \hat{z}_{k|k-1})(\mathcal{Z}_{i,k|k-1} - \hat{z}_{k|k-1})^T \tag{3.36}$$

$$P_{X_k \tilde{z}_k} = \sum_{i=0}^{2L} W_i^{(c)} (\chi_{i,k|k-1} - \hat{X}_{k|k-1})(\mathcal{Z}_{i,k|k-1} - \hat{z}_{k|k-1})^T \tag{3.37}$$

The state estimate and and its error covariance are now able to be updated

$$\mathcal{K}_k = P_{X_k \tilde{z}_k} P_{\tilde{z}_k \tilde{z}_k}^{-1} \tag{3.38}$$

$$\hat{X}_k = \hat{X}_{k|k-1} + \mathcal{K}_k (z_k - \hat{z}_{k|k-1}) \tag{3.39}$$

$$P_k = P_{k|k-1} - \mathcal{K}_k P_{\tilde{z}_k \tilde{z}_k} \mathcal{K}_k^T \tag{3.40}$$

3.3.4 Landmark Augmentation

In this subsection, we will focus on how to perform landmark augmentation [30]. Generally, this technique has the similar manner as EKF-SLAM's landmark augmentation presented in Section 3.2.4. First, the position of a newly observed landmark z_k and its covariance are simply added as follows

$$\hat{X}_k^+ = \begin{bmatrix} \hat{X}_k \\ z_k \end{bmatrix}, P_k^+ = \begin{bmatrix} P_k & 0 \\ 0 & \mathcal{L}_k \end{bmatrix} \tag{3.41}$$

The dimension of the augmented state vector is now equal to $L^* = L+2$. Using equation (3.29) to perform the Unscented Transformation for the pair in equation (3.41) and acquire the following set of sigma points

$$\chi_k^+ = \left[\hat{X}_k^+ \quad \hat{X}_k^+ + \eta^* \sqrt{P_k^+} \quad \hat{X}_k^+ - \eta^* \sqrt{P_k^+} \right] \tag{3.42}$$

Each column of this matrix contains the transformed state vector, including the robot's pose and the already known landmarks, and the transformed newly-observed landmark

$$\chi_{i,k}^{+} = \begin{bmatrix} \chi_k^X \\ \chi_k^z \end{bmatrix}_i \tag{3.43}$$

Take a coordinate transformation to switch the new landmark position in the polar coordinate to the global Cartesian coordinate by using a function $g(.)$

$$\chi_{i,k} = \begin{bmatrix} \chi_k^X \\ g(\chi_k^X, \chi_k^z) \end{bmatrix}_i \tag{3.44}$$

With the sigma points available, we come to the end of this stage by computing the weighted mean and covariance

$$\hat{X}_k = \sum_{i=0}^{2L} W_i^{*(m)} \chi_{i,k} \tag{3.45}$$

$$P_k = \sum_{i=0}^{2L} W_i^{*(c)} (\chi_{i,k} - \hat{X}_k)(\chi_{i,k} - \hat{X}_k)^T \tag{3.46}$$

3.3.5 UKF-SLAM: Simulation Results

We run the simulation for UKF-SLAM in exactly the same environment and initial setups as implementing with EKF-SLAM in previous section. We highlight that the loop closing occurs in the same manner, there is a sudden change in uncertainties when robot is passing by a loop closure as can be seen in Figure 3.10. This is reaffirmed in Figure 3.11, showing the actual error of robot pose in x and y axis together with its 95% confidence bounds.

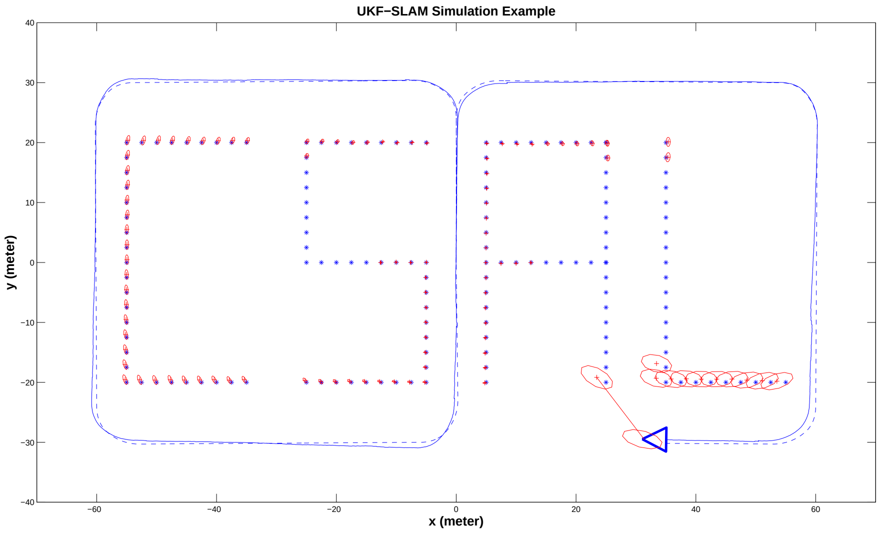
3.4 FastSLAM

So far, we have focused on Kalman filters in solving SLAM problem, producing a recursively closed-form approximation of state and covariance. Now we come to nonparametric method with the use of particle filter, which has become a preferable choice in today SLAM [37].

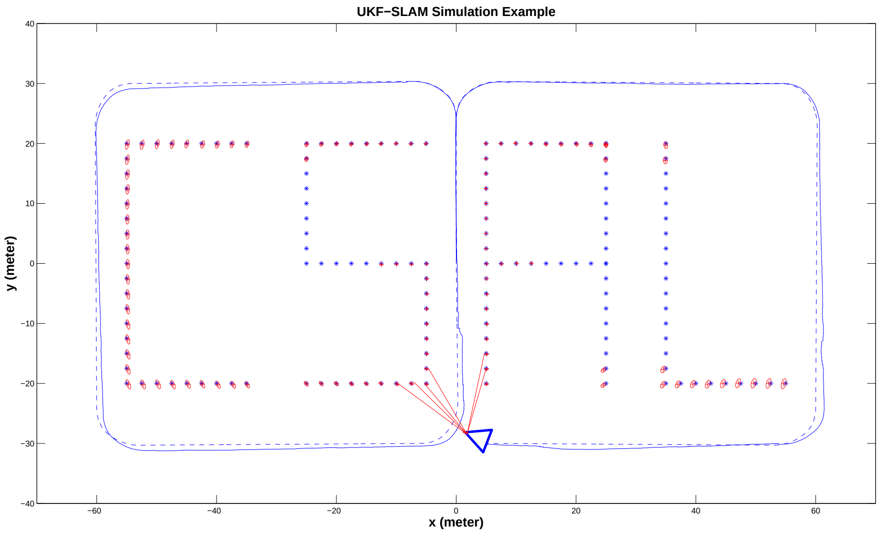
3.4.1 FastSLAM Factorization

In EKF-SLAM and UKF-SLAM, robot is required to estimate the on-line posterior distribution $p(x_k, m|z_{1:k}, u_{0:k-1})$, which is the probability of the whole robot trajectory and the map up to this point in time given all measurements and control inputs received from on-board sensors.

We need to take advantage of SLAM property in regard to dynamic Bayesian network in order to factorize the posterior, which in turn reduces the computational complexity and enable to apply

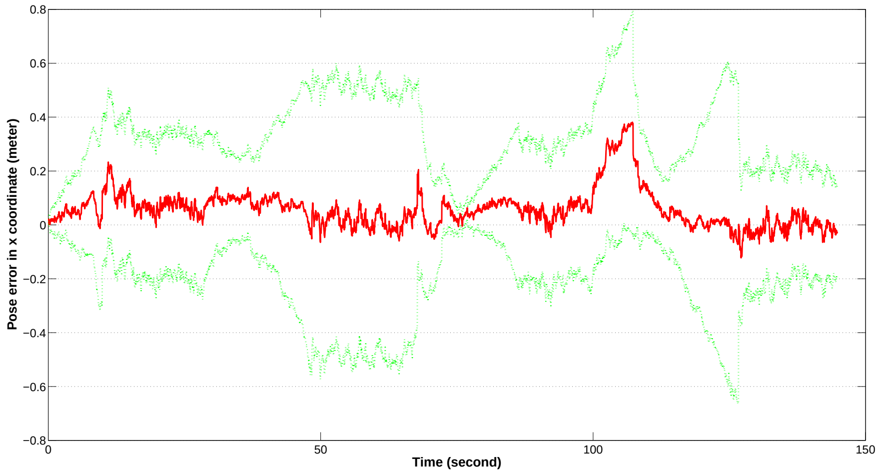


(a) Before closing the second loop

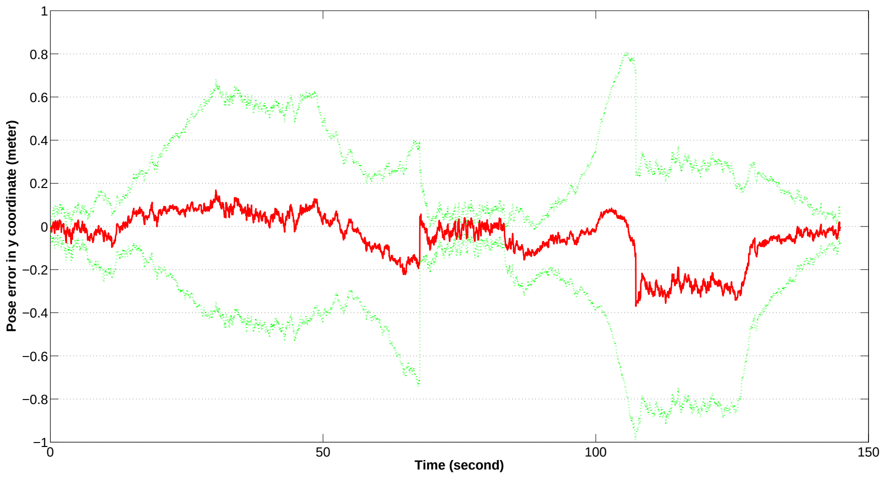


(b) After closing the second loop

Figure 3.10: The loop closing problem in UKF-SLAM which appears to be closely similar to EKF-SLAM



(a) Pose error in x-axis



(b) Pose error in y-axis

Figure 3.11: Actual error in robot pose estimate in x and y axis (solid lines) together with two 95% confidence bounds (dotted lines) in UKF-SLAM

particle filter [35]. In this connection, FastSLAM estimates the full SLAM problem over the robot path from the beginning to current time and the map

$$p(x_{0:k}, m|z_{1:k}, u_{0:k-1}) \tag{3.47}$$

This term can be derived as follows based on the conditional probability definition

$$p(x_{0:k}, m|z_{1:k}, u_{0:k-1}) = p(x_{0:k}|z_{1:k}, u_{0:k-1})p(m|x_{0:k}, z_{1:k}, u_{0:k-1}) \tag{3.48}$$

The first term of the above equation is the robot path posterior whereas the second term is the map estimator given the entire robot path up to this time. From the evolution network of SLAM problem, we observe that all the landmarks are conditionally independent if the robot trajectory is given. Based on this observation and using product rule, the posterior over robot pose and map can be factorized as follows

$$p(m|x_{0:k}, z_{1:k}, u_{0:k-1}) = \prod_{i=1}^{N} p(m_i|x_{0:k}, z_{1:k}, u_{0:k-1}) \tag{3.49}$$

Then equation 3.47 becomes

$$p(x_{0:k}, m|z_{1:k}, u_{0:k-1}) = p(x_{0:k}|z_{1:k}, u_{0:k-1}) \prod_{i=1}^{N} p(m_i|x_{0:k}, z_{1:k}, u_{0:k-1}) \tag{3.50}$$

The interpretation is that SLAM posterior can be estimated via the product of robot path and N landmarks posterior given the robot path.

This interpretation suggests that we can estimate the posterior of FastSLAM using a separate manner. Each term in (3.50) will be governed by one filter, i.e. the first term will apply particle filter whereas the N landmark posteriors will use N different EKF filters. The particle filter needs to maintain a set of M particles and note that the posterior of each landmark is conditioned on the robot path. Therefore each particle is required to possess both the robot pose estimate and a set of N EKFs of N landmarks given by

$$\mathcal{Y}_k^{[i]} = \{x_{0:k}^{[i]}, \mu_{1,k}^{[i]}, \Sigma_{1,k}^{[i]}, \mu_{2,k}^{[i]}, \Sigma_{2,k}^{[i]}, \ldots, \mu_{N,k}^{[i]}, \Sigma_{N,k}^{[i]}\} \tag{3.51}$$

where i indicates i-th particle in the set of M particles at time k. $x_{0:k}$ is the robot estimated path. A pair $\{\mu_j, \Sigma_j\}$ describes mean and covariance of the Gaussian estimating the position of the j-th landmark. Particle filter, in the same manner of the Bayesian filters, will update the particle set $\mathcal{Y}_k$ to $\mathcal{Y}_{k+1}$ when receiving new control signal and measurement. This procedure comprises four key steps which are outlined as follows [35].

3.4.2 Sampling a New Pose

It is often intractable to draw a sample directly from the posterior since it may possess any form. Therefore we need to do this sampling step via a so-called proposal distribution, which is chosen from the probabilistic motion model

$$x_k^{[i]} \sim p(x_k|u_k, x_{k-1}^{[i]}) \tag{3.52}$$

Here $x_{k-1}^{[i]}$ describes the robot pose estimate at time $k-1$ belonging to the i-th particle. We temporarily put these samples into a set that will be used for the next steps.

3.4.3 Calculate Importance Weight

This step corrects the mismatch between proposal and target distribution which is represented by importance weight. Intuitively, these weights will attempt to push back samples drawn in Section 3.4.2 to the posterior distribution.

The proposal distribution of the full path particles:

$$p(x_{0:k}^{[i]}|z_{1:k-1}, u_{0:k-1}) = p(x_k^{[i]}|x_{k-1}^{[i]}, u_{k-1})p(x_{0:k-1}^{[i]}|z_{1:k-1}, u_{0:k-2}) \tag{3.53}$$

The target distribution incorporates the measurement at time k

$$p(x_{0:k}^{[i]}|z_{1:k}, u_{0:k-1}) \tag{3.54}$$

The importance weight is calculated by the ratio between the target and proposal distribution

$$\begin{aligned} w_k^{[i]} &= \frac{\text{target distribution}}{\text{proposal distribution}} = \eta p(z_k|x_k^{[i]}) \\ &\approx \eta|2\pi S_k^{[i]}|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(z_k - \hat{z}_k^i)^T S_k^{[i]-1}(z_k - \hat{z}_k^i)\} \end{aligned} \tag{3.55}$$

where $z_k - \hat{z}_k^i$ and $S_k^{[i]}$ are innovation term and its associated covariance as defined in Equation 3.12 and 3.13.

3.4.4 Measurement Update

This update step will base on successfully observed landmarks while the others remain unchanged. Each observed landmark will be updated using the standard EKF equations

$$\hat{z}_k = h(x_k^{[i]}, \mu_{j,k-1}) \tag{3.56}$$

$$G_{m_j,k} = \nabla_{m_j} h(x_k, m_j)|_{x_k = x_k^{[i]}; m_j = \mu_{j,k-1}^{[i]}} \tag{3.57}$$

$$K_k = \Sigma_{m_j,k-1}^{[i]} G_{m_j,k}(G_{m_j,k}^T \Sigma_{m_j,k-1}^{[i]} G_{m_j,k} + R_k)^{-1} \tag{3.58}$$

$$\mu_{j,k}^{[k]} = \mu_{j,k}^{[i]} + K_k(z_k - \hat{z}_k^{[i]}) \tag{3.59}$$

$$\Sigma_{m_j,k}^{[i]} = (I - K_k G_{m_j,k})\Sigma_{m_j,k-1}^{[i]} \tag{3.60}$$

3.4.5 Adaptive Resampling

Adaptive resampling technique examines which particles best represent the true posterior to remain and delete the other particles. The most common procedure is computing the so-called *effective number of particles* or *effective sample size*, denoted by N_{eff}. This quantity indicates how well the current set of particles characterize the true posterior, which is calculated as follows

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N}(w_k^{[i]})^2} \tag{3.61}$$

where all the weights must be normalized before taking this computation.

This effective number is a way to estimate the variance of importance weights. The higher the weight variance, the worse the approximation of the current particle set. Resampling will be executed when this quantity is considerably less than the number of particles, for instance, $N/10$. This technique presents a significant advantage in reducing depletion of good samples.

3.5 Comparison of EKF-SLAM, UKF-SLAM and FastSLAM

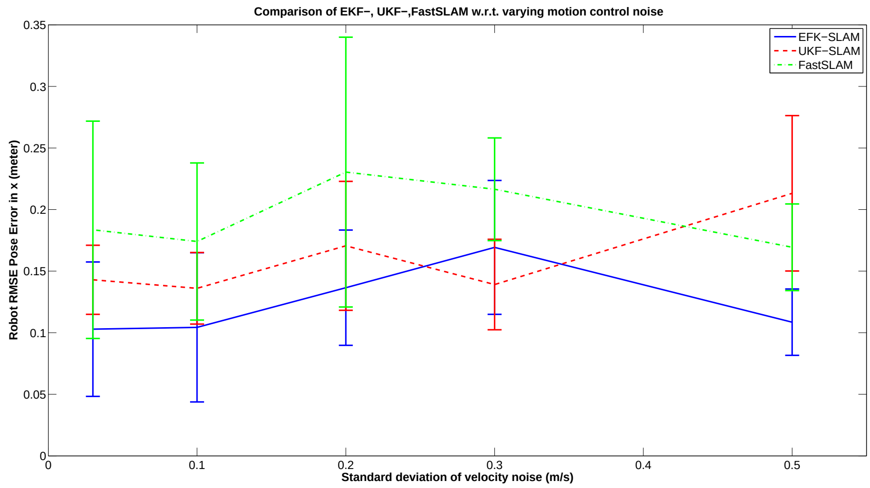
Now, with these solutions derived clearly, we attempt to make comparisons between these algorithms in cases that one of the control noises is varied. We run the simulation of three aforementioned algorithms of SLAM in the same environment with 150 landmarks and initial setups. The steering control noise, range and bearing measurement noise are kept at 3^0, 0.1 m, and 1^0. Robot is moving at 3 m/s and its sensor has the maximum range of 30 m. The first comparative simulation is varying the noise of a control signal, translational velocity as can be seen in Figure 3.12. Each algorithm was run 10 times at each noise level and different random seed were used at each run. Thus the results shown here are the mean over those runs together with associated standard deviation as depicted by vertical bars. In our scenario, it is appeared that the EKF-SLAM tends to be the preferable choise since it gives smaller errors except one case at 0.3. FastSLAM tends to reduce its errors when increasing noise. Overall, we would say all the algorithms return pretty good results with small errors.

The second comparative simulation is varying the steering control noise and record RMSE errors of there algorithms. The final result are shown in Figure 3.13. Here all the errors are small and tend to increase as rising the noise.

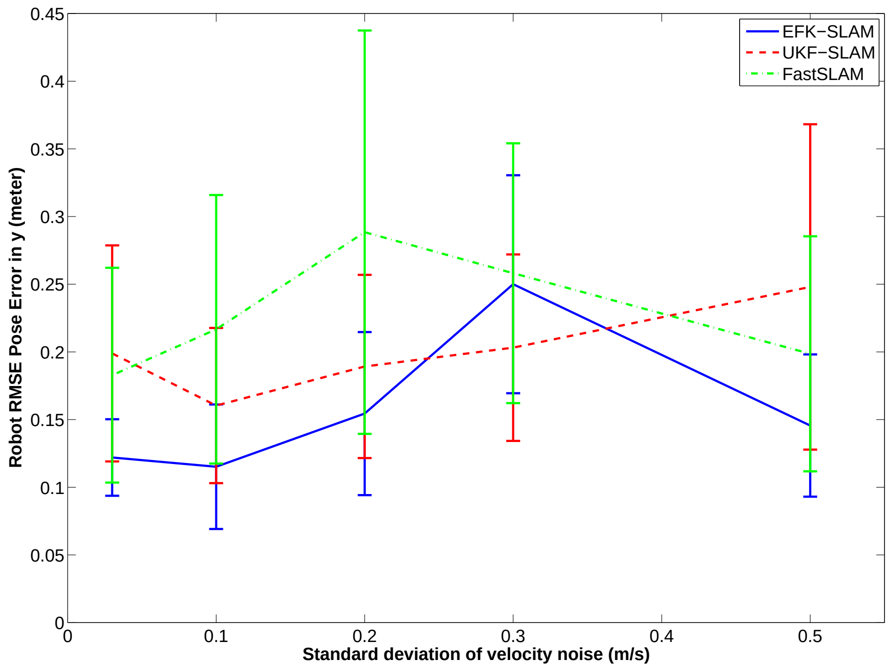
By these simulations, we can notice that the accuracy of the algorithm depends on many factors, not only the control noise itself since we still receive better accuracy when increasing the noise. However, the good thing is the errors we observe in these implementations still remain at low level even the noise is considerably large (e.g. 0.5 m). Therefore we suppose that these algorithms are applicable in integration with other tasks in mobile robotics.

3.6 Conclusion

This chapter consider the SLAM problem together with its well-known solutions. The simulation suggests the key aspect in closing the loop for SLAM mobile robot since it considerably decrease the uncertainties of the whole state. In addition, we also investigate a comparison between those algorithms in terms of root mean square error and receive small, acceptable error levels. To some extent we claim the feasibility of these techniques for our integrated framework.

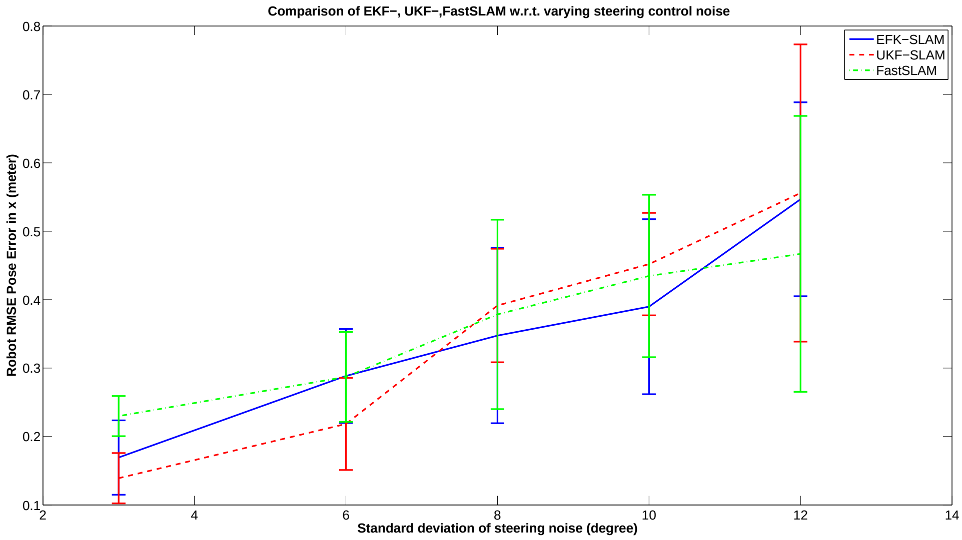


(a) Pose error in x-axis

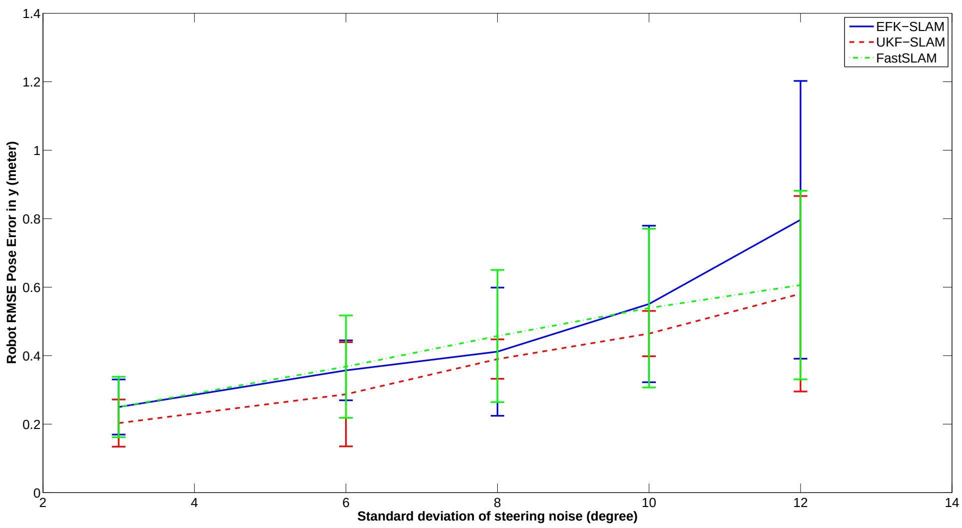


(b) Pose error in y-axis

Figure 3.12: RMSE pose error of robot when varying translational velocity noise



(a) Pose error in x-axis



(b) Pose error in y-axis

Figure 3.13: RMSE pose error of robot when varying steering control noise

CHAPTER IV

EXPLORATION ALGORITHM

This chapter investigates the exploration-based problem for our mobile robot. In this connection, we need to construct an occupancy grid map in order to implement A* path planning algorithm, which both will be presented here. In addition, to perform the framework, as mentioned in Figure 2.1, we also introduce utility and cost functions to define a new destination for robot.

4.1 Occupancy Grid Map

In previous chapter, we have used the feature-based map to represent the vicinity of the robot, which can be efficiently maintained and updated. However, this type of map still owns some drawbacks since we do not know the space between these features. Therefore it would be risky if we let the robot move freely within these partially undefined areas. For this reason, it is necessary to exploit another representation, occupancy grid map [36], to support the navigation part presented in the subsequent sections.

Occupancy grid map is the way in which the world is discretized into many small cells each of which contains its probabilistic occupancy information. This means we have to compute for each cell a posterior probability if the corresponding area of that cell is *occupied* by an obstacle, which also implies the probability of the cell is *empty* that can be traveled through. One advantage of this approach is that it does not require any predefined landmarks and allows the constant time access to the map. Importantly, this map provides the information of the environment, whether a certain place, as small as the size of a cell, is occupied or empty, whether a place has already visited or unexplored. Thus this technique become popular for the use in path planning and exploration tasks.

At first we define the interpretation that we can claim corresponding to occupancy probability

$$p(m_i) = \begin{cases} 1 & \text{occupied} \\ 0 & \text{unoccupied} \\ 0.5 & \text{no knowledge} \end{cases} \tag{4.1}$$

In addition, we assume that the cells $\{m_i\}_{i=1:L}$ of the occupancy map m are independent as shown by

$$p(m|z_{1:k}, x_{0:k}) = \prod_{i=1}^{L} p(m_i|z_{1:k}, x_{0:k}) \tag{4.2}$$

By this assumption, if we know the probability of one cell, we do not know any information of its adjacent cells. In this way, given a new measurement and current robot's pose, we can estimate the

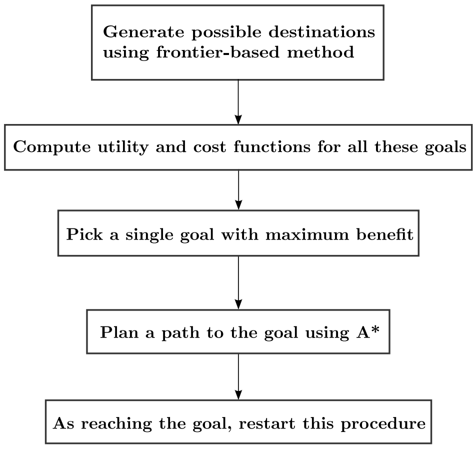


Figure 4.1: Exploration Reference Procedure for Robot

updated probability of cells individually as follows [17]

$$p(m_i|z_{1:k}, x_{0:k}) = [1 + \frac{1 - p(m_i|z_k, x_k)}{p(m_i|z_k, x_k)} \frac{1 - p(m_i|z_{1:k-1}, x_{0:k-1})}{p(m_i|z_{1:k-1}, x_{0:k-1})} \frac{p(m_i)}{1 - p(m_i)}]^{-1} \tag{4.3}$$

4.2 Frontier-based Potential Destinations

Now we can start the exploration algorithm. The reference procedure is then provided in Figure 4.1, which is a part in the whole framework.

When the robot reach its goal, it will perform a step called *frontier-based* approach [31] in order to propose new potential destinations. Frontiers are defined as the cells of the grid map that in between already explored and unexplored regions. In this way, the robot will be guided towards unexplored region to discover as much as area as possible. Of course, there frontiers lie in the edge of the explored region, which is also that of the unexplored region.

4.3 Information Gain and Cost Functions

This section defines function for robot to make its own decision of where to go next by evaluating all frontiers and pick up the "best" one. After the robot determines a list of new frontiers, it will evaluate each single frontier by using these utility and cost function [33]

- **Utility of Information Gain** The purpose of exploration is to collect the information of the world. Therefore we will exploit the information gain utility to define the amount of information that robot will obtain at one specific frontier. Let p_c be the occupancy probability of one cell m_i, the amount of information it possesses is measured by entropy

$$H(m_i) = -p(m_i) \log(p(m_i)) \tag{4.4}$$

To evaluate the information if the robot moves to that frontier cell, we first determine the sensed region W of that cell's point of view corresponding to the maximum range of the laser sensor capability. The second step is to compute the total information of all the cells in W as follows

$$I = \sum_{m_i \in W} H(m_i) \tag{4.5}$$

We note that the higher entropy the more uncertain that region is and the more attractive it is to explore. In addition, the type of sensor used for robot also play a crucial role in this task since it will decide the shape and size of the sensed region W.

- **Cost to Reach Goal** Robot also takes into account the function that approximately evaluate the cost to move from the current pose to the goal. In this work, it is simply determined by the Euclidean distance between these two points, which is also the shortest distance. Hence, this cost will be exact in the case that robot moves along the straight path connecting starting and target point.

$$C_G = \sqrt{(x_{\text{goal}} - x_{\text{start}})^2 + (y_{\text{goal}} - y_{\text{start}})^2} \tag{4.6}$$

- **Steering Cost** Generally, the initial orientation of the robot is different from that of the reference path. It takes robot a cost for steering to the desired direction. The steering cost function C_S defined here to describe this action, which is simply proportional to the different angle between these orientations.

- **Total Benefit** is the weighted utility function subtracted by weighted cost functions. Let w_I, w_G and w_S be the weights of the functions I, C_G, C_S respectively

$$\Sigma_B = w_I I - (w_G C_G + w_S C_S) \tag{4.7}$$

- **Selecting the Destination** Finally, the frontier with maximum benefit will be opted for the new destination.

4.4 A* Path Planning

In order for the robot to navigate to a goal, A* search algorithm has been widely used to find a path answering the question of path planning: "How can I reach there?". Given a graph $G(V, E)$ with nodes and edges, A* will find the shortest path between any pair of a start position and a goal defined in the graph beforehand, as illustrated in Figure 4.2. In particular, under some conditions this path will be optimal meaning that A* eventually returns the minimum cost path.

A* was invented by Hart et al. [21] as an extension of the Dijkstra's algorithm, which both have had a widespread use in real applications to present day such as self-driving cars, game programming, Google maps. By incorporating heuristics in its search process, A* attains better results in terms of time performance.

In our scenario, after its decision process, the robot will perceive the goal where it has to reach at that moment. Therefore, with the map and current pose gained from SLAM, and the goal returned

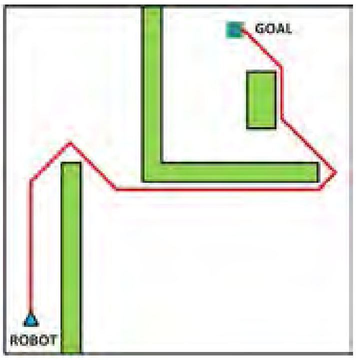


Figure 4.2: Illustration of the A* path planning algorithm

from decision making, we now can apply A* algorithm to get the robot to the said destination. The algorithm is described in more detail as follows and in Algorithm 1.

4.4.1 Algorithm Description

We are first going to explain notations of the A* algorithm. Throughout the search, A* maintains two lists: $OPEN$ and $CLOSED$. The first list contains nodes in which we choose the next one to be explored. The second list includes nodes that have been already considered and nodes seen as obstacles added from the beginning. This means robot will never revisit its past nodes and are not allowed to arrive at unreachable nodes, which both result in the $CLOSED$ list. Other notations are

- N_S, N_G, N_{OBS}, N^* are respectively start node, goal , set of nodes occupied by obstacles, and currently explored node
- $d(N_1, N_2)$ is the distance of two nodes N_1 and N_2
- $\mathcal{G}(N)$ is the function indicating how far the robot has moved from N_S to node N
- $\mathcal{H}(N)$ is the heuristic function estimating the distance cost of the shortest path from node N to N_G
- $\mathcal{F}(N) = \mathcal{G}(N) + \mathcal{H}(N)$ is the total estimated cost of the shortest path from N_S to N_G containing node N

Starting by having N_S become N^* and N_{OBS} stored in $CLOSED$. The main loop consists of checking if each of the adjacent nodes of N^* is feasible to be added into $OPEN$ by the condition that the node is inside the given map and have not yet been in $CLOSED$. The winning nodes will either be added into $OPEN$ if it does not match with any node of $OPEN$ or will be updated the accompanied cost if otherwise and the said one has smaller cost than the one already in OPEN. Note that we need to compute $\mathcal{G}$, $\mathcal{H}$, and $\mathcal{F}$ of each node that, together with the information of its parent, will be always accompanied with that node. Next in the loop, the priority check will be performed,

Algorithm 1 Pseudo code of A* algorithm

Given(N_S, N_G, N_{OBS}, $\mathcal{G}$, $\mathcal{H}$, d)
Add N_S into $OPEN$
Add N_{OBS} into $CLOSED$
while $OPEN$ list is not empty **do**
    find a node with the smallest $\mathcal{F}$ in $OPEN$, call it N^*
    put N^* in $CLOSED$
    if $N^* = N_G$ **then**
        complete the search, exit while loop
    end if
    for each node N' adjacent to N^* and not in $CLOSED$ **do**
        mark N^* as their parent
        $\mathcal{G}(N') = \mathcal{G}(N^*) + d(N^*, N')$
        $\mathcal{H}(N') = d(N^*, N')$
        $\mathcal{F}(N') = \mathcal{G}(N') + \mathcal{H}(N')$
        if $N' \notin OPEN$ **then**
            insert N' to $OPEN$
        else if $\mathcal{F}(N'_{old}) > \mathcal{F}(N'_{new})$ **then**
            update parent of N'_{old} in $OPEN$ to be N^*
            update $\mathcal{G}$, $\mathcal{F}$ of N'_{old} to be N'_{new}'s
        end if
    end for
end while
return the minimum-cost path by tracing back the pointer from the N_G to N_S

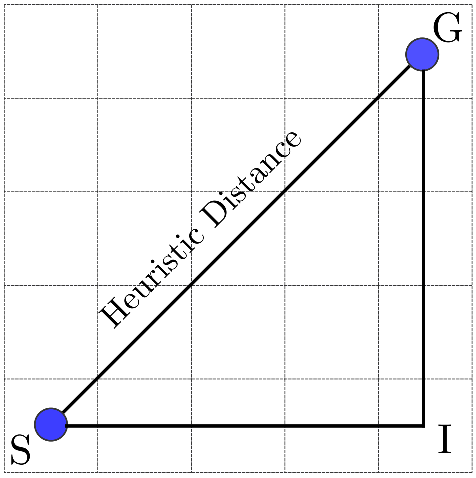


Figure 4.3: Illustration of heuristic and actual distance in a grid map. Heuristic distance = Euclidean distance = SG. Actual distance = Manhattan distance = $SI + IG$

where the node possessing the smallest cost in $OPEN$ will be chosen to be the new N^*, which is then removed from $OPEN$ and added to $CLOSED$. The loop continues on and terminates if N^* is N_G, returning the optimal path, or $OPEN$ is empty, meaning no solution for the problem.

4.4.2 Optimality of A*

A* algorithm will generate an optimal path from N_S to N_G, i.e. lowest-cost path, if the heuristic function is admissible [20], which happens providing that the estimated cost between any node in the graph to the goal given by the function $\mathcal{H}$ is always less than or equal the actual cost.

In the grid map, the actual cost path is the Manhattan cost, measuring the total absolute difference between two points in every Cartesian dimension, whereas the heuristic cost is opted to be the Euclidean distance, as depicted in Figure 4.3. That's why we claim that using A* in our scenario ensure the optimal solution.

4.4.3 Example of A* Path Planning

We conduct a simulation for A* algorithm presented in this section. Provided are an initial robot location (square), a goal (polygon) and obstacles (many circles) depicted in Figure 4.3. As we can see, robot can find the optimal path to reach the goal.

4.5 Control Algorithm for Robotic Vehicle

The path generated from A* includes a set of points that guide the robot to the goal. In fact, it has a number of turning angle along the way so that the robot definitely needs a controller keeping track to the planned path.

In this work, we use two proportional controllers with separate objectives. The first one aims to minimize the robot's distance d to the line whereas the second controller adjusts the orientation of

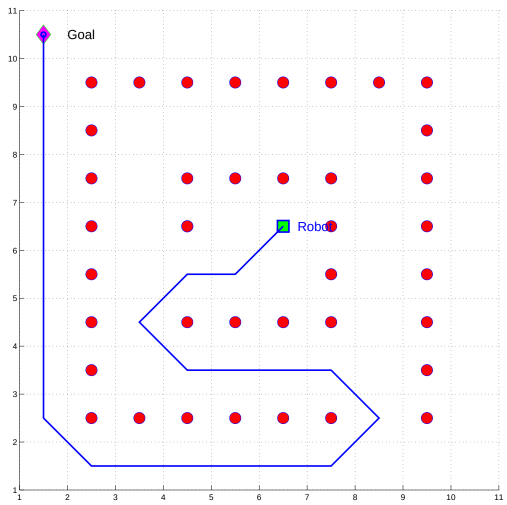


Figure 4.4: An simulation example of A* algorithm

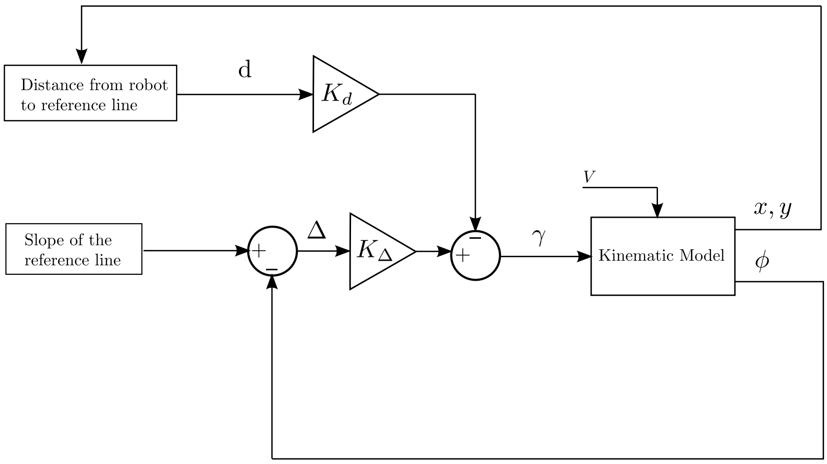


Figure 4.5: Diagram of the tracking controller

the robot to be parallel to the reference line. Therefore we obtain the equation for the steering angle

$$\gamma = -K_d d + K_\Delta \Delta = -K_d d + K_\Delta(\phi_r - \phi) \tag{4.8}$$

where $K_d, K_\Delta > 0$ are controller gains. ϕ_r is the orientation of the reference line and ϕ is the current orientation of the robot.

4.6 Conclusions

This chapter has investigated the procedure for integrated approach by using exploration algorithm. To apply path planning and exploration, it is necessary to maintain an occupancy grid map where robot can perceive both empty space and occupied-by-obstacle space. The exploration algorithm first let the robot define the frontier region between the unexplored and explored in the map.

By using the total benefit function, based on information gain, cost to reach goal and steering cost functions, the robot is able to decide the best destination to go by picking up the frontier with maximum benefit. After that, A* path planning algorithm will generate the path for robot to reach the destination. Finally, a tracking control assist robot to track the planned path.

CHAPTER V

SIMULATION RESULTS

This chapter provides the main simulation results of the integrated algorithm on which we will draw significant conclusions and discussions.

5.1 Integrated Approach

Revisit the whole framework in Figure 2.1, we are now able to perform the entire integrated approach.

- SLAM is used for localization and feature-based mapping in which robot will have a means to know its location.
- Building Occupancy grid map is essential for path planning and exploration. This map will be updated and maintained in parallel with SLAM feature-based map. In general, two these maps are complimentary to each other.
- The decision making process bases on proposing new frontiers whenever robot reaches its current destination and applying benefit function in order to pick the best frontier (new destination).
- A* algorithm to plan path and a steering controller to keep track to that path.

Putting all algorithms together, we now can proceed the simulation test for the integrated approach.

5.2 Simulation Setup

We setup an environment with the size of 100cells x 100cells containing many landmarks in which

- Each landmark takes up 1 cell of the map. In practice, the object smaller than the cell size is also considered occupying one cell.
- Robot is assumed to have the size of one cell.
- Objects in the map except the robot are static.
- To visualize, the gray area indicates unexplored region (occupancy probability $p = 0.5$) whereas the white color shows already explored region ($p \to 0$) and the occupied area (landmarks) is shown by small squares ($p \to 1$) as can be seen in Figure 5.1. The frontiers will be shown by

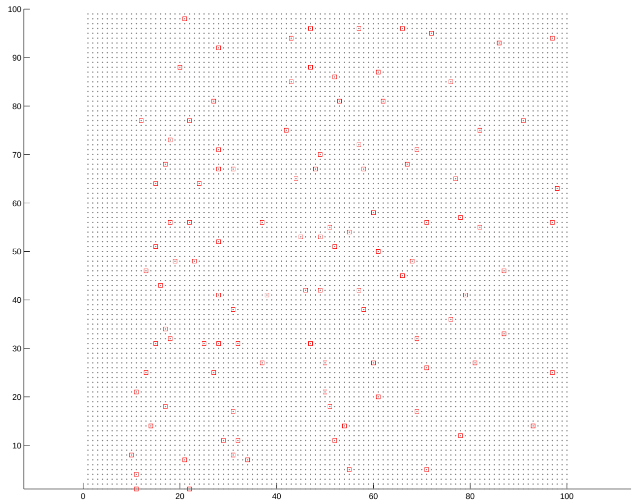


Figure 5.1: Initial occupancy grid map of the simulated environment

small circles while the path generated by A* is a solid line (usually, this is a line with many turning angles)

The parameters for robot are set up as follows

- Robot is visualized by a triangle. Its velocity is 1cell/s. And the time interval is 0.05s.
- Robot wheelbase is 1cell.
- Maximum range of sensor is 15m. The shape of sensed region is a half circle.
- Controller parameters: $K_d = 0.8, K_\Delta = 1$

5.3 Simulation Results

The robot now starts exploring! During this simulation, we have captured videos of exploration action that will be enclosed with this thesis.

The first scene in Figure 5.2 shows the frontiers between the explored (white) and the gray regions. The best-benefit frontier will be chosen and a path to that frontier will afterward be generated as shown by solid line. In fact, at this moment robot is steering to track the path. We also note that only the landmarks (small squares) are already perceives by robot while the others remain unexplored.

When the robot reach the goal, it immediately defines a new list of frontiers and continues the same procedure as can be seen in Figure 5.3.

Figure 5.4 shows the observation action is still taking place at the same time robot is moving to the goal. That is the reason why we have the white area outside the closed area defined by the

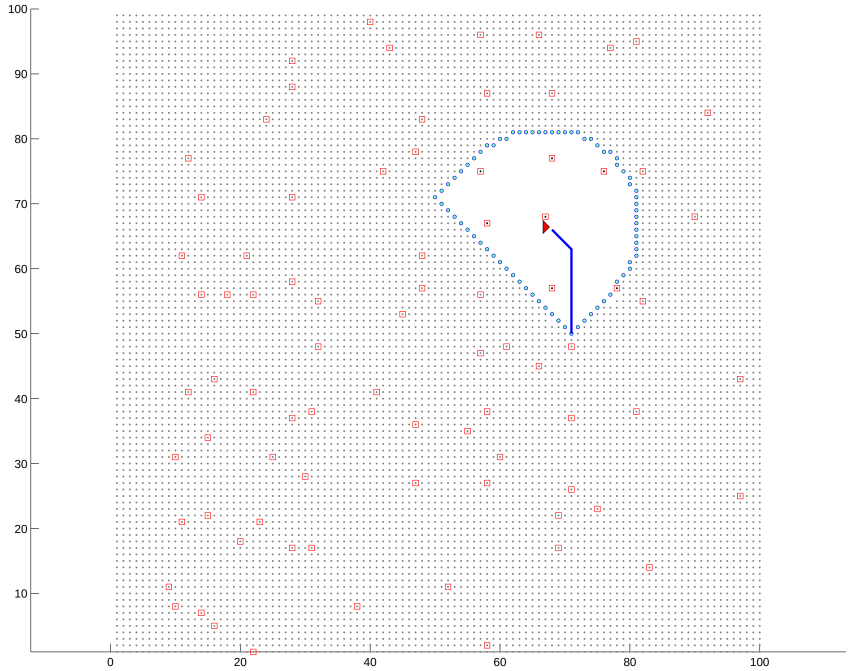


Figure 5.2: Robot defined the frontiers and picked a destination. Now robot is starting moving to the planned path

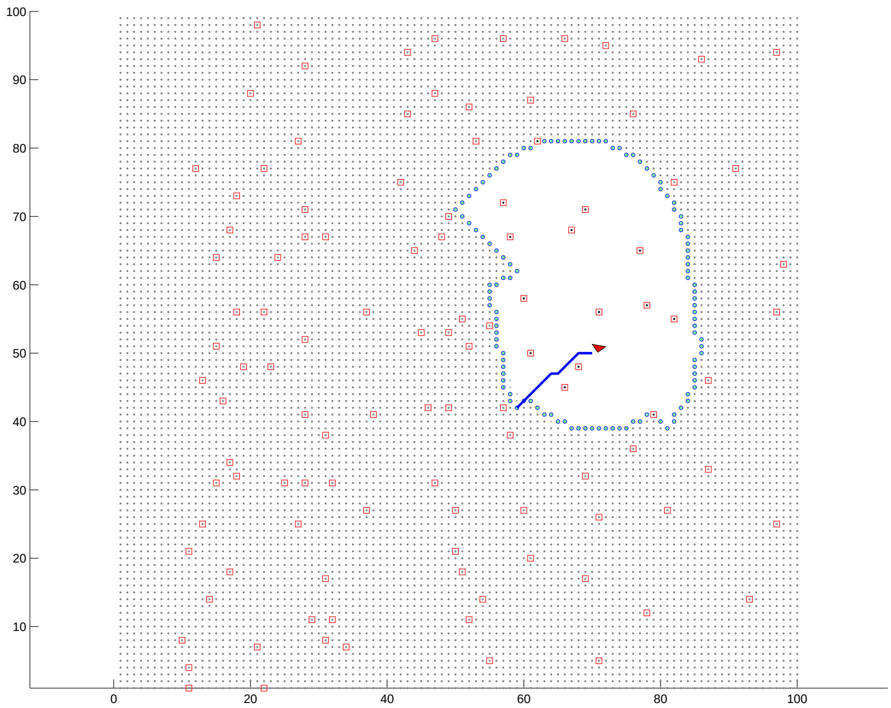


Figure 5.3: Robot reached the goal and continue the same procedure

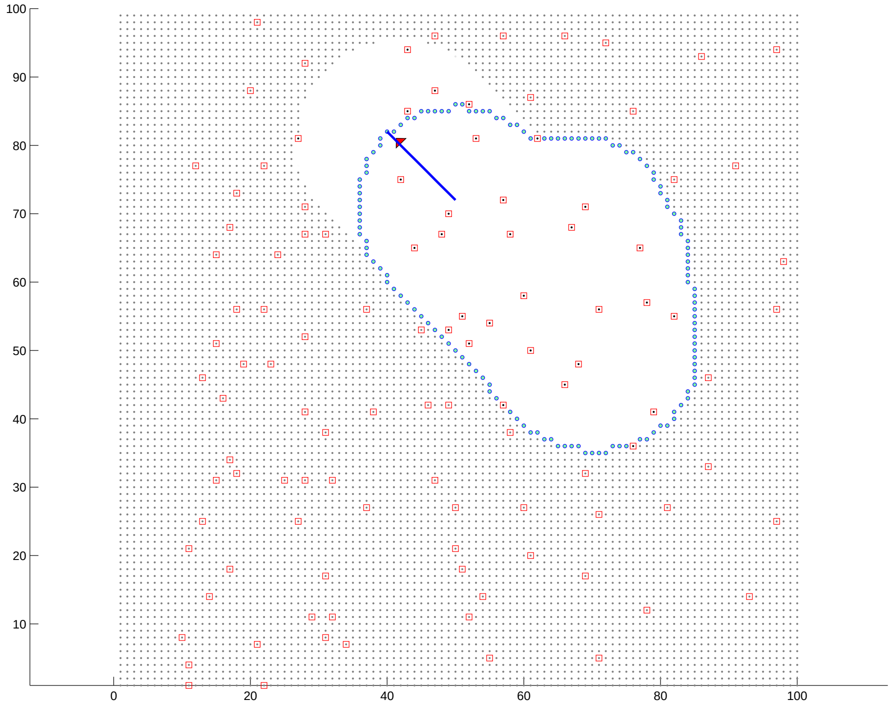


Figure 5.4: An example to illustrate the sensed region of robot corresponding to its sensor's capability

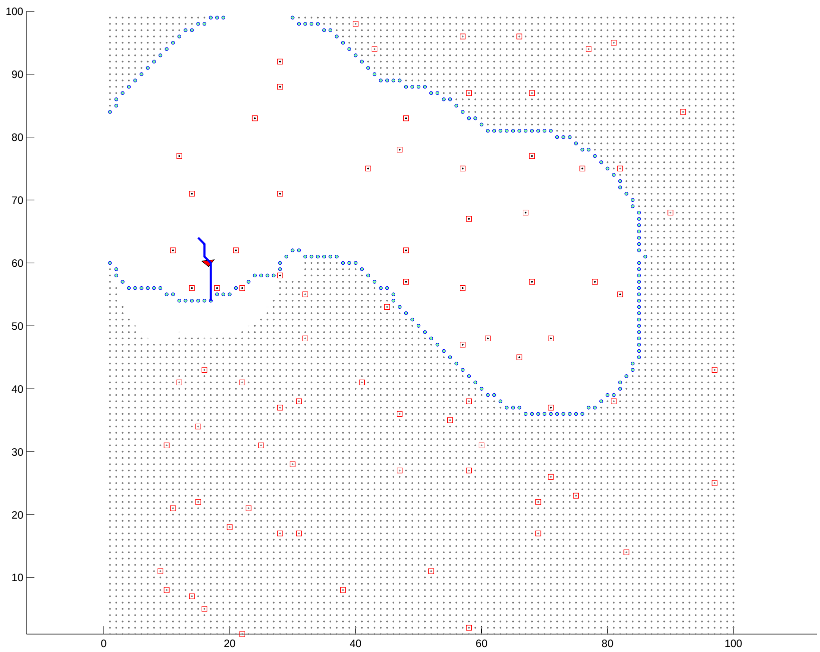


Figure 5.5: A snapshot showing the exploration process after some time. Robot is now moving to the goal.

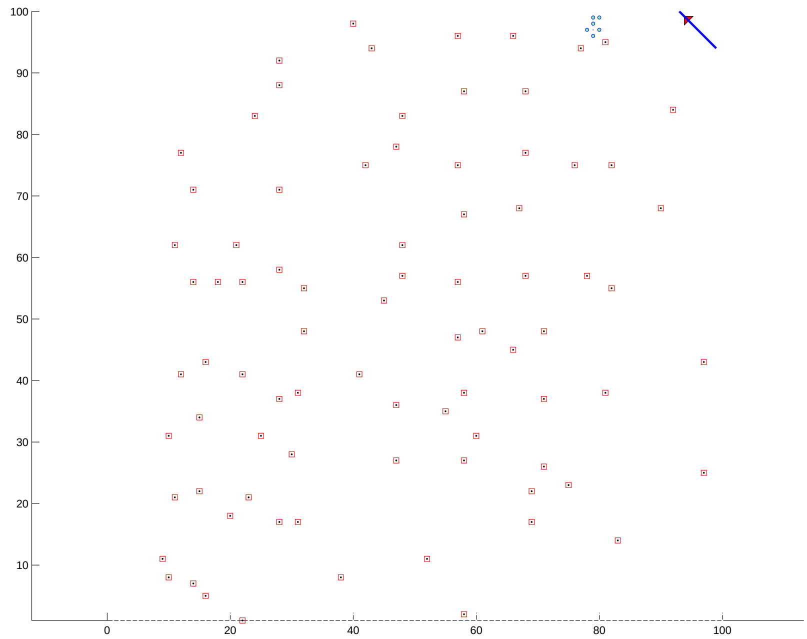


Figure 5.6: A snapshot showing robot completed its exploration. The whole area have been fully explored.

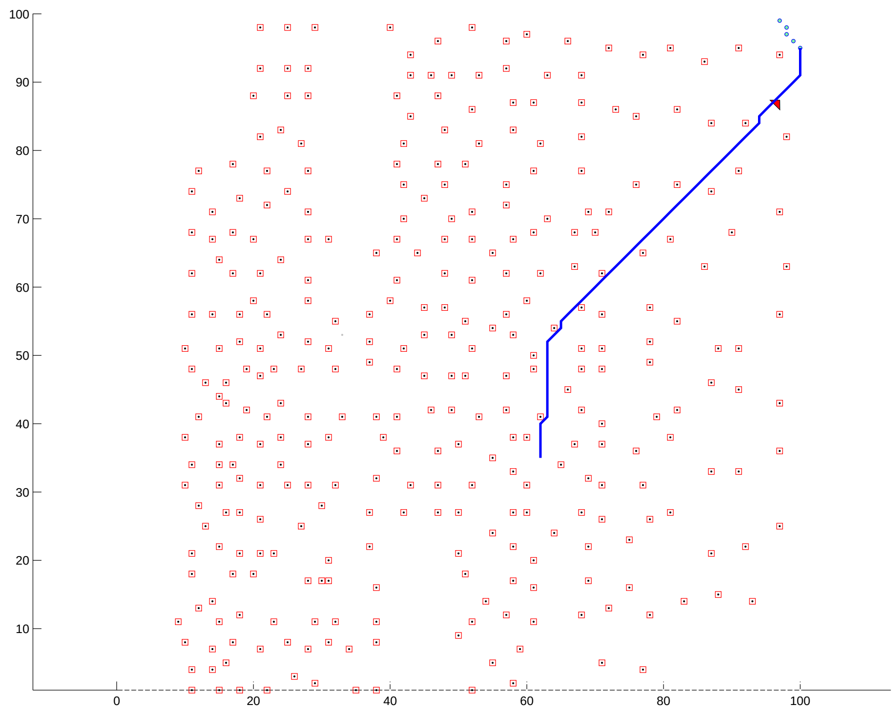


Figure 5.7: The second implementation in dense environment. Robot eventually fully explores the whole area

frontiers. As we can notice, the half-circle shape of robot's sensor is demonstrated by the curved edge between explored and unexplored regions.

Figure 5.5 depicts the scene after some time and the generated path has some turning angles. The exploration procedure stops when the whole area is fully explored as in Figure 5.6, every place has been whitened. Because robot is moving and observing at the same time so that in the last action it even did not need to complete the planned path in order to explore the remaining area. That is why some trace is still left in the top right corner of the figure.

We attempt to implement the second simulation in which we create a lot more landmarks as in Figure 5.7. Robot eventually succeeded the exploration procedure with the whole area fully explored.

5.4 Summary and Discussion

Through simulation results we conclude that the integrated approach to SLAM and A* path planning algorithm works well since robot is able to fully perform the required tasks shown by the whole area totally explored. There are, however, several drawbacks in this technique

- There is a trade-off between the accurate representation of grid map and the velocity of the robot. If the resolution of the map is small, then we have to slow down the robot to make sure it is able to follow the path that generated in cell-to-cell basis. This issue can be improved by using an smoothing algorithm to eliminate many turning points of the pure A* path.
- The benefit function that we used in this work only gives the local maxima. We need to consider other approaches that further optimize the exploration algorithm.
- The robot still needs the aid from the landmarks, meaning that some information has to be predefined which then decrease the concept of *truly autonomous robots*.

CHAPTER VI

CONCLUSIONS

6.1 Concluding Remarks

This work has been motivated by the framework that to some extent make the robot truly autonomous. We then start to deal with tasks needed for mobile robot basically including localization, mapping and path planning. We decide to study an approach that is able to integrate these tasks.

This work starts by investigating SLAM problem that provide robot the capability to construct the map and localize itself in that environment. By exploiting two main branches of Bayesian filtering, which are parametric and non-parametric approach, we have selected three solutions for SLAM to take into account. We also show the key aspect of SLAM problem by simulating the affect of loop closing without which SLAM will turn to be infeasible. The comparison between these algorithms also give a sense in the accuracy to be able to apply SLAM in other works.

The bridge to connect SLAM with path planning algorithm is introduced based on the purpose of autonomous mobile robot moving and collecting information of the world. The information utility will guide the robot to reach the region with richer uncertainties to explore. On the other hand, we also need to take into account the travelling cost function and steering cost in order to minimize the time of motion. This approach also requires the frontier-based technique to determine the preferred region to calculate the benefit and select the destination.

We also investigate the A* path planning algorithm to generate the path from the current point of robot to its goal. Making sure the robot tracks the generated path requires a controller to generate steering action, which has also been introduced.

The simulation results at the end prove the feasibility of our proposed frameworks. Robot performs many required tasks as planned and eventually reach the state that the robot's world is fully discovered. There exist some drawbacks in this approach that are presented in the same chapter.

6.2 Future Works

This work consists of many basic tasks in robotics that open very many extensions if one wishes. We now outline our own ideas and intentions that we will study after this thesis

- We wish to deal with challenging dynamic environment that is closer to what really happen in real life. The obstacles now will be set moving with uncertainties!
- As mentioned earlier, the benefit function is not optimal, which will be a way for us to come.

- We also would like to deal with dynamic programming for planning a path and being able to deal with unpredictable objects appearing on the way.

- Regarding SLAM problem, handling the huge number of landmarks or big-size maps can be extended to improve existing drawbacks of SLAM in this work. Moreover, 3D-SLAM with camera will be very beneficial in our life.

REFERENCES

[1] M. Julià, A. Gil, and O. Reinoso, "A comparison of path planning strategies for autonomous exploration and mapping of unknown environments," *Auton. Robot.*, 33, 4, Springer US, (2012): 427–444.

[2] P. Jensfelt, *Approaches to Mobile Robot Localization in Indoor Environments,* PhD Thesis, Department of Signals, Sensors and Systems, Royal Institute of Technology, Sweden, 2001.

[3] T. Bailey, *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments,* PhD Thesis, Australian Centre for Field Robotics, The University of Sydney, 2001.

[4] T. Bailey, *SLAM Toolboxes*, [Online] Available at http://www-personal.acfr.usyd.edu.au/tbailey/

[5] J. J. Leonard, R. N. Carpenter, and H. J. S. Feder, "Stochastic mapping using forward look sonar," *Robotica*, 19, 5, (Aug. 2001): 467–480.

[6] J. T. Wooden, *Graph-based Path Planning for Mobile Robots,* PhD Thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, USA, 2006.

[7] J. Kim, *Autonomous Navigation for Airborne Applications,* PhD Thesis, Australian Centre for Field Robotics, The University of Sydney, 2004.

[8] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics and Automation Magazine*, 13, 2, (June 2006): 99–110.

[9] R. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Int. J. Robotics Research*, 5, 4, (Dec. 1986): 56–68.

[10] A. A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte, "An experiment in integrated exploration," *IEEE Int. Conference on Intel. Robots and Systems*, 2002.

[11] S.-Y. Lin and Y.-C. Chen, "SLAM and navigation in indoor environments,", in *Advances in Image and Video Technology*, ed. Y.-S. Ho, Springer Berlin Heidelberg, (2012): 48–60.

[12] H. J. S. Feder, J. J. Leonard, and C. M. Smith, "Adaptive mobile robot navigation and mapping," *Int. J. Robotics Research*, 18, 7, (1999): 650–668.

[13] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, New York: Springer-Verlag, (1990): 167–193.

[14] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. on Robotics and Automation*, 17, 3, (June 2001): 229–241.

[15] C. Stachniss, G. Grisetti, and W. Burgard, “Information gain-based exploration using Rao-Blackwellized particle filters,” *Proc. of Robotics: Science and Systems (RSS)*, Cambridge, MA, USA, (2005): 65–72.

[16] S. J. Julier, *Process Model for the Navigation of High-Speed Land Vehicles,* PhD Thesis, University of Oxford, 1997.

[17] W. Burgard and M. Hebert, “World modeling,” in *Springer Handbook of Robotics*, Springer Berlin Heidelberg, (2008): 853–869.

[18] U. Frese, R. Wagner, and T. Röfer, “A SLAM overview from a user’s perspective,” *KI - Künstliche Intelligenz*, Springer-Verlag, 24, 3, (2010): 191–198.

[19] U. Frese, “Interview: Is SLAM Solved?,” *KI - Künstliche Intelligenz*, Springer-Verlag, 24, 3, (2010): 255–257.

[20] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, MIT Press, Boston, 2005.

[21] E. P. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost path,” *IEEE Transactions on System Science and Cybernetics*, 4, 2, (1968): 100–107.

[22] C. Stachniss, *Robotic Mapping and Exploration. Springer Tracts in Advanced Robotics*, Springer Berlin Heidelberg, 55, 2009.

[23] J. Solà, *Simultaneous localization and mapping with the extended Kalman filter*, [Online] Available at http://www.joansola.eu/JoanSola/eng/JoanSola.html

[24] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, 2nd edition, MIT Press, 2011.

[25] A. J. Haug, *Bayesian Estimation and Tracking: A Practical Guide*, Wiley, 2012.

[26] S. Julier and J. Uhlmann, “Unscented filtering and nonlinear estimation,” in *Proc. of the IEEE*, (2004): 401–422.

[27] R. Martínez-Cantin and J. Castellanois, “Unscented SLAM for large-scale outdoor environments,” *IEEE Int. Conference on Intel. Robots and Systems*, (2005): 3427–3432.

[28] G. P. Huang, A. I. Mourikis and S. I.Roumeliotis, “On the complexity and consistency of UKF-based SLAM,” *Proc. of IEEE Int. Conf. on Robot. and Automat., (2009): 4401–4408.*

[29] E. A. Wan and R. van der Merwe, “The unscented Kalman filter,” in *Kalman Filtering and Neural Networks*, ed. S. Haykin, John Wiley& Sons, (2001): 221–280.

[30] J. Zhu, N. Zheng, Z. Yuan, Q. Zhang, X. Zhang, and Y. He, "A SLAM algorithm based on the central difference Kalman filter," *IEEE Intelligent Vehicles Symposium*, China, (2009): 123–128.

[31] B. Yamauchi, "A frontier based approach for autonomous exploration," in *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, CA, USA, 1997.

[32] A. Schultz, W. Adams, and B. Yamauchi, "Integrating exploration. localization, navigation and planning with a common representation," *Autonomous Robots*, 31, 3, (1999): 293–308.

[33] H. N. Pham, *A Comprehensive Architecture for the Cooperative Guidance and Control of Autonomous Ground and Air Vehicles*, Master Thesis, Australian Centre for Field Robotics, The University of Sydney, 2007.

[34] G. Lidoris, *State Estimation, Planning, and Behavior Selection Under Uncertainty for Autonomous Robotic Exploration in Dynamic Environments*, Dr.-Ing. Thesis, University of Kassel, Germany, 2011.

[35] M. Montemerlo and S. Thrun, *FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*, Springer-Verlag Berlin Heidelberg, 2007.

[36] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, MIT Press, Cambridge, 2005.

[37] S. Thrun, "Simultaneous localization and mapping," in *Robotics and Cognitive Approaches to Spatial Mapping*, Springer Berlin Heidelberg, (2008): 13–41.

APPENDIX

BAYES FILTER ALGORITHM

The aim of the algorithm is to estimate the state x given sets of measurements z and controls u by taking into account the posterior distribution, or also known as *belief distribution* [36]. Let us denote the belief distribution of our state x_k at time k to be $bel(x_k)$ which is given as follows

$$bel(x_k) = p(x_k|z_{1:k}, u_{0:k-1}) \tag{7.1}$$

The generalization of Bayesian filtering includes two main steps, *Prediction* and *Measurement Update*, which are widely applied in the estimation problems. All the solutions of SLAM in this work employ this procedure

- **Prediction** This step is the convolution of the previous posterior $bel(x_{k-1})$ and the motion model $p(x_k|x_{k-1}, u_{k-1})$ of the robot

$$\overline{bel}(x_{k-1}) = p(x_k|z_{1:k-1}, u_{0:k-1}) = \int p(x_k|x_{k-1}, u_{k-1})\, bel(x_{k-1}) dx_{k-1} \tag{7.2}$$

- **Measurement Update** This update step is the product of the prediction and the measurement model

$$bel(x_k) = \eta\, p(z_k|x_k)\, \overline{bel}(x_{k-1}) \tag{7.3}$$

RANGE LASER SENSOR MODEL

Sensor model for a laser range finder [22] used for updating the occupancy grid map in Equation 4.3

$$p(m_i|z_{k,n}, x_k) = \begin{cases} p_{prior} & m_i \text{ is not covered by } z_{k,n} \\ p_{occ} & |z_{k,n} - dist(x_k, m_i)| < r \\ p_{free} & z_{k,n} \geq dist(x_k, m_i) \end{cases} \tag{7.4}$$

where $dist(x_k, m_i)$ is the distance between the sensor and the cell m_i; r is the resolution of the grid map; and n indicates n-th beam of the sensor.

The cell from which the sensor receives measurement $z_{k,n}$ is marked as *occupied*. The cells in between the sensor and the occupied cell are *free*. In addition, the assigned probabilities must hold

$$0 \leq p_{free} < p_{prior} < p_{occ} \leq 1 \tag{7.5}$$

Biography

Hong Khac Nguyen was born in Dong Nai Province, Vietnam, in 1988. He received his Bachelor's degree in Mechatronics from Ho Chi Minh City University of Technology, Vietnam, in 2011. He has been granted a scholarship by the AUN/SEED-Net to pursue his Master's degree in Electrical Engineering at Chulalongkorn University, Thailand since 2012. He conducted his graduate study with the Control Systems Research Laboratory, Department of Electrical Engineering, Faculty of Engineering, Chulalongkorn University. His research interests include Simultaneous Localization and Mapping problem (SLAM), Bayesian Filtering, Path Planning, and Robot Exploration.

List of Publications

1. H. K. Nguyen and M. Wongsaisuwan, "A Study on Unscented SLAM with Path Planning Algorithm Integration," *11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, Nakhon Ratchasima, Thailand, (2014): 1–5.

2. H. K. Nguyen and M. Wongsaisuwan, "A Case Study of EKF-SLAM Application on Mobile Robot," *Proc. of the 5th AUN/SEED-Net Regional Conference in Electrical and Electronics Engineering*, Bangkok, Thailand, (2013): 1–4.