“I hereby declare that I have read through this report entitle “Investigation on Path Planning for Autonomous Mobile Robots in Partially Observable Environment” and found that it has comply the partial fulfilment for awarding the degree of Bachelor of Mechatronics Engineering.

Signature : ..........................................................................................

Supervisor’s Name : NUR ILYANA BT ANWAR APANDI

Date : ..........................................................................................

# INVESTIGATION ON PATH PLANNING FOR AUTONOMOUS MOBILE ROBOTS IN PARTIALLY OBSERVABLE ENVIRONMENT

**LAW CHENG QUAN**

**A report submitted in partial fulfilment of the requirements for the degree of Bachelor of Mechatronics Engineering**

**Faculty of Electrical Engineering**

**UNIVERSITI TEKNIKAL MALAYSIA MELAKA**

**2017**

I declare that this report entitled “Investigation on Path Planning for Autonomous Mobile Robots in Partially Observable Environment” is the result of my own research except as cited in the references. The report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : ..........................................................................................

Name : LAW CHENG QUAN

Date : ..........................................................................................

To my beloved father and mother

# ACKNOWLEDGEMENT

First and foremost, I would like to express my immeasurable appreciation and deepest gratitude to University of Technical Malacca (UTeM) for providing an opportunity for me to undertake my Final Year Project in partial fulfilment for Bachelor of Mechatronics Engineering.

I am deeply indebted towards my project supervisor, Miss Nur Ilyana bt Anwar Apandi for her patience guidance and keen interest on me at every stage of my project progression. Her prompt encouragements, timely assistance, erudite advice, and warm kindness have motivated me to perform better and widen my research boundaries in the completion of my Final Year Project.

Thanks are also extended to my panels, Ms. Nurul Fatiha bt Johan and Ms. Nurdiana bt Nordin who have assessed my presentation and gave valuable comments for my project.  Also, I would take this opportunity to express my gratitude to my parents for their continuous shower of love, unceasing encouragement and support throughout all these years.

Last but not least, I place on record, my sense of gratitude to one and all who, directly or indirectly, have offered their helping hand during the entire period of Final Year Project.

# ABSTRACT

The autonomous mobile robot uses partially observable Markov decision processes (POMDP) model for the shortest and the best path planning to reach a destination in a partially structured environment. POMDP model is applied to improve computational efficiency of path planning problem. Sensing and information processing is important in autonomous mobile robots. Path planning in the real world is difficult because of partial observability and dynamic changes in the environment. Computational complexity increases when more variables are involved. The Perseus algorithm is investigated and the outcomes such as value function, reward and number of vectors are evaluated on different POMDP problems. Perseus algorithm improves belief point collection and selection to compute for better value functions. The algorithm randomly explores the belief space of an environment and collect a set of reachable belief points which will be fixed throughout the algorithm. Then, new value functions are computed to update the belief points. The algorithm repeats until a convergence criterion is met. Varying number of states and actions have significant effects on value function and number of vectors. While reward depends on the value of reward state and cost state.

# ABSTRAK

Robot mudah alih autonomi menggunakan pemerhatian sebahagian proses keputusan Markov (POMDP) dalam perancangan jalan yang singkat dan terbaik untuk sampai ke destinasi yang dalam persekitaran berstruktur sebahagian. Pemprosesan sensing dan maklumat adalah penting dalam robot mudah alih autonomi. Perancangan laluan dalam dunia sebenar adalah sukar kerana keteramatan separa dan perubahan dinamik dalam persekitaran. Kerumitan pengiraan meningkat apabila lebih pembolehubah yang terlibat. Algoritma Perseus disiasat dan hasil seperti fungsi nilai, ganjaran dan bilangan vektor dinilai pada masalah POMDP berbeza. Algoritma Perseus meningkatkan koleksi titik kepercayaan dan pilihan untuk membuat pengiraan untuk fungsi nilai yang lebih baik. Algoritma secara rawak meneroka ruang kepercayaan alam sekitar dan mengumpul satu set mata kepercayaan dicapai yang akan tetap sepanjang algoritma. Kemudian, fungsi nilai baru dikira untuk mengemaskini mata kepercayaan. Algoritma mengulangi sehingga kriteria penumpuan dipenuhi. nombor yang berbeza-beza negara dan tindakan mempunyai kesan yang besar ke atas fungsi nilai dan bilangan vektor. Walaupun ganjaran bergantung kepada nilai ganjaran negeri dan negeri kos.

# TABLE OF CONTENTS

## LIST OF TABLES

# LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

## 1.1 Introduction

Uncertain environment in robotics research is a challenge to the operation of autonomous robotic systems [1]. Autonomous navigation in partially observable domains is an extensive research area in mobile robotics. For this reason, researchers have developed methods for mobile robots to overcome dynamic changes in the environment since the last few decades.

Partially observable Markov decision processes (POMDP) provide a powerful framework for mobile robot planning under uncertain environment. POMDPs generalize Markov decision process (MDP) model and offer a natural and principled framework for sequential programming to allow more variables to be incorporated in the process [7]. The POMDP model contains several qualities such as abstraction, adaptability and robustness [7]. The POMDP model is used for robot navigation [2], exploration tasks [8], machine learning [9] and other purposes. Consider a mobile robot is moving in a real world, which the robot perceives it as a grid world in discrete time, each grid represents a state in which the robot acts. A transition probability function tells the robot what to do by observing the environment through its sensors. The robot receives a reward or cost after performing the action. Thus, the POMDP component will have system state, action, transition probability function, reward function, observations, observation function, belief state and discount factor if it is a finite criterion. The robot has to generate a belief-state space over the underlying state space by using an algorithm to compute the robot’s current location. The robot must be equipped with sensors to detect obstacles such as walls and landmarks in order to update the belief state.

In the principle of mathematics, the complexity of algorithms increases when the number of variables increases. The number of variables especially state space and observation space grow exponentially over time, making computation for exact solution impossible. Over the years, many researches have done to increase the scalability of the algorithms that is able to solve larger problems such as decentralized-POMDP [5], hierarchical-POMDP [3], and dynamic Bayesian networks [10]. Although there are many advance navigation algorithms are introduced, but they are still not ready for dynamic changes of the real world.

## 1.2 Motivation

The primary challenge in implementing POMDPs for navigation is that the robot has to first model the physical environment to state spaces. Due to partial observability of the state, the robot does not know the exact location it is in, so it cannot execute the action recommended for that state [28]. This increases the computational costs of any associated algorithm resulting in high computational complexity [2]. In reality, the state is not always giving the exact information, and the sensor of the robot is not always giving the accurate value.

A significant of research have done on the application of POMDP in mobile robots over the past few decades. The algorithm’s scalability can be improved by decreasing the order of observation functions from exponential to polynomial [5]. Hierarchical decomposition of POMDP enables mobile robot to breakdown large and complex maps to formulate a sequence of sensing and processing suitable for its main objective [3]. The robot needs to plan sequentially one after another to find coordinated trajectories with an adapted version of classical prioritized planning [6]. However, the algorithm becomes impossible to compute when the size of the observation set increases. The motivation of this research is to solve the problem by using an extension to point-based value iteration algorithm, known as Perseus algorithm. The algorithm samples a finite set of reachable belief points and then update the belief points.

## 1.3 Problem Statement

Development of autonomous mobile robot is heavily based on sensing and information processing to a specific task. Path planning in the real-world domain is particularly difficult because partial observability and dynamic changes occur continuously. The existing mobile robot system is built for static environment only, which is prone to sensing error during navigation when it is equipped with sensors [3]. Every action executed by the mobile robot may affect the total reward it will receive. However, the mobile robot may take a considerable amount of time to evaluate the long-term reward from its action. Moreover, human proficiency and time to provide detail and accurate feedback is crucial in designing a mobile robot to navigate in complex domains [3].

It is necessary for a mobile robot to respond quickly to dynamic changes on the environment so that it would not need human intervention during operation. The sensor of a robot is important to provide information on changes of the environment. However, sensor is not reliable because it does not provide accurate information of the real world consistently. The sensor may not work properly due to its physical constraints or when it breaks down. Application of the project is an exploration task specifically for logistics. A mobile robot is deployed with a predefined of an environment. The task of the mobile robot is to transfer an object from a source to a destination. Essentially, it has to plan the best route to transfer the object. The main objective of the mobile robot is to maximize the reward when undergoing each path planning algorithm [4].

## 1.4 Objectives

The objectives of the project are:

1. To investigate path planning for mobile robots by using Perseus algorithm.
2. To evaluate the performance of the algorithm in terms of value function, reward and number of vectors in partially observable environments.

## 1.5 Scopes

The scopes of the project are:

1. Partially observable Markov decision processes is applied to model the path planning problem in a partially observable environment.
2. A predefined map of an enclosed area in an indoor environment is stored in a mobile robot for path planning task.
3. The map is a 2D bounded environment.
4. A single robot will be navigating in the enclosed area.
5. Evaluated factor is the average reward collected by the robot when undergoing each path planning algorithm.

## 1.6 Thesis Organization

The thesis is organized as follows. The next chapter presents the literature review on POMDPs and other well-known methods for solving POMDP problems. Chapter 3 describes the methodology and application of the algorithm in a mobile robot travelling in a partially observable area. Chapter 4 compares and discusses the result using Perseus algorithm between several well-known POMDP problems. Chapter 5 concludes the overall work and proposes recommendation for future research based on the outcomes of the research.

# CHAPTER 2

# LITERATURE REVIEW

## 2.1 Path planning

Path planning of autonomous mobile robot is a challenge in the real world. In this project, the robot already has a grid-based world of the predefined map of an enclosed area. The grid world defines each state of the world that allows the robot to plan a path over it. Every action on a state is offered with a specific amount of cost or reward.

When there are changes in the domain map due to changes in object arrangements, the robot automatically updates the map and recalculates the path to a destination [12]. The path planning algorithm such as modified pulse-coupled neural network (MPCNN) [13] uses a simple neural network by first collecting the robot's location, destination and obstacles that plan the shortest collision-free path so that the robot moves to the grid cell containing the highest reward [12]. Energy consumed by the robot can be reduced when planning an optimal path [13]. However, MPCNN do not include a learning function that lets the robot to learn.

In recent years, new path planning methods have been introduced such as improved Q-learning (IQL) and heuristic searching techniques for mobile robots [14]. The methods limit the belief space and variation range of the mobile robot. There are two path planning methods that are used in a static environment. Global path planning is finding a path before execution in a static environment. This planning method is computationally intractable in more complex environment. Local path planning is used in partially observable environment. The IQL algorithm is combined with several exploration strategies to reduce computational time [14]. However, IQL method is still having difficulty in path planning in a dynamic environment with a large number of state spaces.

## 2.2 Partially observable Markov decision processes (POMDPs)

When a mobile robot is moving in a partially observable environment, the mobile robot is often difficult to make the best decision to achieve task objective. Hence, this kind of problem has to be modeled as POMDPs. This method is gaining popularity in the modern autonomous mobile robot application though it is computationally complex. POMDPs planning is capable of predicting the future by studying the history for a finite time bound. However, applying the general POMDP model to perform simple tasks such as navigation only is PSPACE-hard [6], which means it is very complex and not efficient to compute [28]. Hence, many research focus on improving the navigation algorithm [1], [2] or reducing the number of variables for computation [19, 20].

An exploring autonomous mobile robot has to update state of the environment every time the state changes. Dynamic Bayesian Network (DBN) is used to monitor the environment changes and update the belief state at each node in the network [23]. However, new nodes are generated when updating the DBN and may reach high computational complexity after a short time for large problems.

Task planning and motion planning level are a sequence of operations that take the robot to reach the task objectives. Motion planning level are executed only after the task planning level has computed, and this may lead to undesired motion planning solutions when the proposed task plans are too difficult to be computed by the motion planning level. This problem is modeled as the simultaneous task and motion planning (STAMP) problem. The STAMP problem is integrated with task motion multigraph (TMM) algorithm to increase the efficiency in solving STAMP problem. TMM-based algorithm is able to solve problems that demonstrates Markov Decision Processes (MDPs) [24]. However, due to the nature of MDP problems, the integration with TMM often unable to accurately solve the problem and it takes longer time to compute, though this paper focuses on solving POMDP problems, which is an extension of MDPs.

## 2.3 Point-based value iteration (PBVI) algorithm

The problem is that the complexity of the process becomes more difficult because the number of state space increases exponentially as time goes by. The introduction of the PBVI algorithm has been able to approximately solve large POMDPs rapidly. This section discusses about the PBVI algorithm.

The PBVI algorithm samples a representative set of points from the belief space and use it to represent the space approximately. Recent algorithms are more efficient by sampling a set of reachable under arbitrary sequences of actions. Approximate POMDPs solutions can be obtained efficiently by using PBVI algorithm [19]. Ideally, the algorithm selects belief points that are spread evenly across the reachable belief space to cover as much reachable space as possible within a given horizon.

Successive Approximations of the Reachable Space under Optimal Policies (SARSOP) is another algorithm that computes the optimal policy on a range of optimally reachable belief space. SARSOP is proven to improve computational efficiency when performing simple robotic tasks including navigation [19]. Finding a range of reachable belief space is the key for this algorithm to solve for an optimal policy. However, finding the range that is close to the optimal value function is difficult even the size of the belief space is polynomial.

Several works have conducted on separated POMDP model into a hierarchy of processes to achieve much simpler computations. Hierarchical POMDP (H-POMDP) is capable for collaboration of human and mobile robot to achieve task objectives together. Multi robot collaboration is also achievable using the H-POMDP formulation by adding another layer for communication between robots. H-POMDP consists of three levels which is high-level for visual sensing, intermediate-level for information selection and low-level for information processing [3]. However, a significant amount of data and modelling algorithms have to be coded manually.

## 2.4 Perseus algorithm

One of the extension to the PBVI algorithm is the Perseus algorithm [32]. The Perseus algorithm performs random exploration in the belief space, then samples an action and observation to update the belief state by running several trials. The trials continue to get a large number of points over the belief space. For each successive iteration, Perseus improves the approximation of value function by performing a one-step backup of each belief. During each iteration, Perseus improves standard PBVI by omitting the improved beliefs by another backup.

## 2.5 Summary of Path Planning technique

In this chapter, autonomous mobile robot is more preferred for the industry because of the automaticity and absence of human operator. The POMDP model provides a powerful framework for modelling uncertainty and also predicting for close future. The PBVI algorithm can effectively compute belief points over a belief space to achieve near optimal outcome. This thesis discusses POMDP model based on Perseus algorithm to achieve the objective effectively in terms of execution time and success probability. The method is applied in path planning and navigation of autonomous mobile robot in an enclosed area such as a warehouse.

# CHAPTER 3

# METHODOLOGY

The aim of the POMDP model is to solve autonomous mobile robot navigation uncertainty in a partially-structured environment. The mobile robot has to plan the best route from the start to a specified destination. Besides planning the shortest distance to a destination, the logical path is also considered. The robot has to continue to the next destination without returning to the starting point, until it has achieved the objectives. For autonomous mobile robot, the robot is always waiting at the starting point for new instructions.

When the state of the mobile robot is too large, solving for a policy requires tremendous time and computational power. Perseus is an extension to PBVI algorithm that is able to solve large state spaces without compromising time and computational power. This chapter discusses POMDP model, PBVI algorithm and Perseus to be applied in a mobile robot.

## 3.1 The POMDP model

In the real world where mobile robot navigation is concerned, decision-making is the fundamental problem for the robot. The mobile robot has to determine the best action during the decision-making processes to achieve an optimal reward or accomplish the main objectives. When the environment around the robot changes dynamically, observations need to be included into decision-making process. The problem is, the mobile robot has to consider the rewards after a sequence of action, which gives rise to sequential planning in a stochastic environment. POMDP serves as a powerful framework developed to account for the problem. A mobile robot will be represented as an agent from here onwards.

The objective of POMDP planning is to discover a policy $\pi$ to select an action for the agent. The policy defines how the agent should act in order to maximize the rewards. There are several types of policies, history-dependent or Markov, stochastic or deterministic [27]. In POMDP formulation, the observation often depends only on the current state of the process, regardless of the history. Also, including histories into the process can be an exhaustive task, so the belief state is used in the space of probability distributions over states. POMDP models with belief states can be generalized into a belief-space MDP models. This formalism is widely used in the POMDP to model the agent's navigation. We focus on applying discrete and finite state space and action space. Continuous state space is also used in POMDP to scale up the algorithm to simulate a near actual environment [18].

POMDPs provide a framework for sequential planning to allow more forms of uncertainty into the process. The system states of the POMDP model is represented by belief states that are used for decision-making [16]. A POMDP [1-3, 15-17] is formally denoted as a tuple <$S$, $A$, $T$, $Z$, $O$, $R$, $b$, $\gamma$>. Variables in capital letter is the complete set variable in an environment, small letter denotes a certain set of variable used for calculation. The variables are defined in Table 3.1.

When an agent does not know the exact state, the agent can only act depending on observations it can perceive. However, the sensor of the agent may not give accurate observations of the state. The agent has to assign a probability distribution over the state known as the belief state $b$. The probability of the belief state assigned to an actual state is written as $b(s)$. The agent must update the current belief state to a new belief state for the actions taken and observations made so far. A technique called recursive function is used to calculate the new belief state $b'$ from the previous belief state and new observation. The new belief state is given by

$$b'(s') = \alpha P(o|s') \sum_{s} T(s'|s,a)b(s), \tag{3.1}$$

Table 3.1: Variables of a general POMDP model.

| Variable | Name | Description |
|---|---|---|
| $S$ | State space | A discrete and finite set of all system states, which are observable and unobservable, that represents the environment where the robot acts. |
| $A$ | Action space | A discrete and finite set of actions at each time instant. |
| $T$ | State transition | A probability function that passes the current state to the next state. The value is within an interval [0,1]. It is defined as $T: S \times A \times S' \rightarrow [0,1]$. Also, $\sum_{s \in S'} T(s, a, s') = 1, \forall (s, a)$. The notation $S'$ is the subsequent state $S$. Notation $T(s, a, s')$ is the state transition in current state $s$ given action $a$ moving into next state $s'$. |
| $Z$ | Observation space | A finite set of observations. The observations include noisy inputs of the true state of the environment through the robot sensors. |
| $O$ | Observation function | A function that represents the conditional probability given the action and the subsequent state. The function depends on the triplet $(z, a, s')$. |
| $R$ | Reward function | Immediate reward function that assigns a real value executing action $A$ in state $S$. Negative reward represents a cost. The function directs an agent towards the goal location. Also defined as $R$: $S \times A$. |
| $b$ | Belief state | The agent's knowledge or belief of the state of the environment. It is a probability distribution over all possible states $S$. |
| γ | Discount factor | A real value within the interval [0,1). An infinite sequence becomes finite ensures the algorithm converges to a final value. |

where $\alpha$ is a normalizing constant that makes belief state sum to 1. The subsequent belief state takes the summation for $s = 0, 1, \dots, S$ in an environment the agent is exploring. The agent does action according to its current belief state, not the actual state. This means that the optimal policy $\pi^*(b)$ maps belief states to actions. In fact, the action changes subsequent belief state when the agent observed the outcome of its action. Hence, action can be considered as one of the performance of the agent.

Given the current belief state $b$ and action $a$, we can calculate the probability the agent would reach in the subsequent belief state $b'$. We do not know the subsequent observation yet, so the agent might reach in one of several possible belief states $b'$. The probability of observation $o$ given that action $a$ was performed in belief $b$ is given by

$$\begin{aligned} P(o|a,b) &= \sum_{s'} P(o|a,s',b)P(s'|a,b) \\ &= \sum_{s'} P(o|s') \sum_{s'} P(s'|s,a)\, b(s). \qquad (3.2) \end{aligned}$$

To find the transition probability of mapping $b$ to $b'$ given action $a$ as $T(b'|b,a)$, we get

$$\begin{aligned} T(b'|b,a) = T(b'|a,b) &= \sum_{s'} T(b'|o,a,b)T(o|a,b) \qquad (3.3) \\ &= \sum_{o} T(b'|o,a,b) \sum_{s'} T(o|s') \sum_{o} T(s'|s,a)b(s). \qquad (3.4) \end{aligned}$$

Equation (3.4) can be used as the transition model for the belief state. The reward function for belief states is

$$\rho(b) = \sum_{s} b(s)R(s). \qquad (3.5)$$

The probability $T(b'|b,a)$ from Equation (3.3) and (3.4) and reward function $\rho(b)$ from Equation (3.5) can represent an observable MDP on the space of belief states. The optimal policy for this MDP, $\pi^*(b)$, is also the optimal policy for the original POMDP. Hence, POMDP in the physical state space can be generalized into an observable MDP on the corresponding belief-state space. This is because we assume the belief states are fully observable to the agent. Figure 3.1 illustrates the process of a general POMDP algorithm. The algorithm will stop after it reaches a terminating condition, usually a convergence criterion or within a limited time. Though, in mobile robot, it is usually programmed to do other task after reaching the goal state, such as placing down objects.

## 3.2 Value function

Value function is one of the characteristic of Markov decision processes. Finding an optimal policy can be immediately transformed into an optimization problem in terms of value functions. This results in a less complex optimality equations resolution than exploring the whole set of policies. We can use the Bellman equation for the belief-space MDP to generate the value function, *V*:

$$V = \max_{a \in A} R(b, a) + \gamma \sum_{b' \in B} \tau(b, a, b')V(b') \tag{3.6}$$

$$= \max_{a \in A} R(b, a) + \gamma \sum_{o \in O} T(o|b, a)V(b^{a,o}). \tag{3.7}$$

In both the finite and infinite horizon case, the value function $V$ can be modeled almost closely as the upper envelope of a finite set of linear functions, known as $\alpha$-vectors. Now, the value function can be written as $V = \{\alpha_1, \dots, \alpha_n\}$ to define over the full belief of the process. The value at a given belief can be computed as:

$$V(b) = \max_{\alpha \in V} b \cdot \alpha, \tag{3.8}$$

where $b \cdot \alpha = \sum_{s \in S} b(s) \cdot \alpha(s)$ is the standard inner product operation in vector space.

The Bellman equation from Equation (3.6) serves as an important aspect in value iteration algorithm to solve POMDPs. If there are *n* states, then there are *n* Bellman equations corresponding to each state. However, the Bellman equation is not linear, because the “max” operator is not a linear operator.