Predictive Autonomous Robot Navigation
PhD Thesis Proposal

Amalia Foka
Department of Computer Science
University of Crete

March 7, 2002

1 Introduction

For humans, the ability to navigate intentionally is eminent. For mobile robotic systems, however, navigation in dynamic real-world environments is an extremely complex and challenging task. Such environments are characterized by their complex structure and the movement of humans and objects in them.

An important problem that a navigating robot is facing in a crowded environment is that it can easily get blocked by moving humans and obstacles. In such cases, the robot becomes immobilized and not able to continue its movement towards its goal position until the moving obstacles and humans free its way. To avoid ever getting in such a situation, many researchers have tried to predict the motion of humans and obstacles. The future motion prediction of humans and obstacles allows the robot to estimate if the way it follows is going to be blocked and thus change direction before it ever faces this situation.

Future motion prediction is an intrinsic behavior of humans. Consider the example of a man trying to cross a street. A man looks at his left and right to see if any vehicles are approaching. If there are no vehicles he is certain that he can cross the street safely and so he does. In the case that a vehicle is approaching, he tries to estimate how long it will take for the vehicle to reach the point that he stands and then decides if he should cross the street. This is in effect a predicitve behavior. It would be desirable for an autonomous robot to develop a similar behavior.

Generally speaking, a robot has to make decisions about the actions it should take each time step considering the information its sensors provide about the environment state. When a robot is given a specific task, it is faced with the question "What should I do now?"[22]. This question has to be answered when:

- "now" is the current state of the robot (e.g. location, orientation)
- "do" is one of the actions the robot can perform (e.g. move forward, move left)
- "should" is maximize a long-run measure of reward
- "I" is an automated planning or learning system (agent)

Actually, the robot faces this question continuously until it reaches the goal position. Thus, the robot has to solve a *sequential decision problem.*

The solution of a sequential decision problem in a completely observable environment, where the robot always knows its state, is called a Markov Decision Process (MDP) [references]. When there is uncertainty or not enough information to determine the state of the robot, the problem is called a Partially Observable Markov Decision Process (POMDP).

In this thesis it is proposed to integrate the future motion prediction of humans or obstacles into the sequential decision problem of navigation modelled by a POMDP. Previous work that made use of future motion prediction has made assumptions about the movement of obstacles and it has been only used in local collision avoidance modules [19, 60, 64, 52, 29, 6, 61, 34]. In this thesis we intend to eliminate or, alternatively, relax such assumptions about the movement of obstacles and also make use of this information in the global navigation model. This will allow for more efficient use of this information. Moreover, the POMDP model is proposed in order to formulate the navigation task because it provides a probabilistic way of representing information. The probabilistic method of representing information in POMDP's also solves the localization problem. POMDP's have been previously used in robotics but only in either simple environments [5, 20] or as a high-level mission planner with a separate module for collision avoidance [54]. This approach proposes the use of a POMDP model that has integrated the localization and collision avoidance with the use of future motion prediction modules for autonomous robot navigation.

The problem of multi-robot navigation will also be considered in this thesis. In applications that many robots operate in the same environment, there is the need for coordination and communication between the robots. This approach will consider the case where there are m robots and when a goal is specified, the robot that will get there easier and in a more efficient manner should eventually reach the goal location. This behavior is extremely useful for service robots where it is of no interest which robot will serve a specific task as long as one of them does.

The multi-robot navigation task will be modelled with the use of a POMDP as in the single robot case. In addition to the localization and collision avoidance with future motion prediction modules, the modules for the communication and coordination of the robots will also be incorporated in the POMDP. To the best of our knowledge this will be a first attempt to tackle the multi-robot navigation, i.e. reach a goal-position, problem.

The rest of this document is organized as follows. In section 2, the necessary background for POMDP's and its application in robotics is given. In the same section, there is an overview of the available techniques for the prediction of the movement of humans or other moving obstacles. Finally, in section 2 there is a survey of the up-to-date methods for multi-robot communication and coordination. In section 3, the modelling of the predictive robot navigation problem using POMDP's is detailed. Some preliminary experimental results are given that highlight our approach. Finally, in section 4 there is the discussion of the proposed approach and the reasons for its choice are justified.

2 Literature Review

The navigation task can be broken down into three parts [16]:

- localization, the process of figuring out where the robot is;
- mapping, the process whereby the robot builds a model of its environment;
- planning, the process of figuring out how the robot can get to other places.

Robots are inherently uncertain about the state of their environments. Uncertainty arises from sensor limitations, noise and the fact that real-world environments are unpredictable. Autonomous robots must act in the face of uncertainty. Uncertainty led researches to use probabilistic approaches in navigation. In probabilistic navigation information is represented through probability densities. In particular, perception and action is performed in a probabilistic manner.

The field of probabilistic localization and mapping has been studied quite extensively. On the other hand, the field of probabilistic planning is much poorer developed [56]. This thesis will be concerned with this problem taking care also of the localization problem by utilizing Partially Observable Markov Decision Processes (POMDP's).

2.1 Partially Observable Markov Decision Processes (POMDP's)

POMDP's are a model of an agent interacting synchronously with its environment. The agent takes as input the state of the environment and generates as output actions, which themselves affect the state of the environment. In the POMDP framework, a system acting in the world is not guaranteed at any time to know the state of the world, and the system may not even know its own state with respect to the world (e.g. its exact position may be only partially observable).

A POMDP is a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{Z}, \mathcal{O} \rangle$, where

- $\mathcal{S}$, is a finite set of states of the environment that are not observable
- $\mathcal{A}$, is a finite set of actions
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \Pi(\mathcal{S})$ is the *state transition function*, giving for each state and agent action, a probability distribution over states. $\mathcal{T}(s, a, s')$ is the probability of ending in state s', given that the agent starts in state s and takes action a, $p(s'|s, a)$.
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the *reward function*, giving the expected immediate reward gained by the agent for taking each action in each state, $\mathcal{R}(s, a)$
- $\mathcal{Z}$, is a finite set of observations
- $\mathcal{O} : \mathcal{A} \times \mathcal{S} \rightarrow \Pi(\mathcal{Z})$ is the *observation function* giving for each state and agent action, a probability distribution over observations. $\mathcal{O}(s', a, z)$ is the probability of observing z, in state s' after taking action a, $p(z|s', a)$.
- an initial belief state, $p_0(s : s \in \mathcal{S})$, a discrete probability distribution over the set of environment states, $\mathcal{S}$, representing for each state the agent's belief that is currently occupying that state.

In mobile robot navigation, each state is a location in a world and each observation is a discrete percept that can be produced by the robot sensors. If the objective of the robot is to reach a particular goal state, then the reward will be maximized when the robot has reached this particular state. The POMDP is controlled by executing one of the available actions at each time step. The result of executing an action will be to transfer to a new unobservable state and perceive a new observation. The POMDP solves the robot's localization problem by maintaining at all times a belief distribution over all possible states it can be in, that is updated according to the actions the robot takes and the percepts it observes.

2.1.1 Solving POMDP's

Solving a POMDP amounts to choosing the actions that will maximize the expected cumulative reward. A *policy* π is a mapping from $\mathcal{S}$ to $\mathcal{A}$, specifying an action to be taken in each situation. Given a policy π and a reward function $\mathcal{R}$, the *value* of state $s \in \mathcal{S}$, $V_\pi(s)$, is the expected value of the sum of rewards to be received at each future time step, discounted by how far into the future they occur. That is,

$$V_\pi(s) = E(\sum_{t=0}^{\infty} \beta^t \mathcal{R}_t) \tag{1}$$

where R_t is the reward received at the tth step of executing policy π after starting at state s and β is a discount factor.

The goal of the agent is to compute the optimal policy, $\pi^*(s)$, that is to maximize the quantity $E(\sum_{t=0}^{\infty} \beta^t \mathcal{R}_t)$.

It has been shown that finding an exact solution of a POMDP is intractable [23]. The general case of an infinite-horizon, stochastic POMDP is EXPTIME-hard for boolean rewards and is not known to be decidable for general (but bounded) rewards.

A number of techniques have been proposed for finding approximations to the value function or policy that give satisfactory results. Many of these techniques transform the POMDP into a fully observable MDP, whose states are the belief states of the original POMDP [5], and is referred to as a "belief MDP". The optimal policy is then the solution of this "belief MDP".

The most commonly method for solving MDP's is the *value iteration* [22]. This method starts by determining the value function for a horizon length of 1. This is simply choosing the action with the highest reward for each state, since there is only the need to make a single decision. The next step will be to determine the value function for a horizon of 2. That is, to add to the value function for each state, computed in the previous state, the reward for each of the possible actions. The algorithm iterates until the value function for the desired horizon length is found. Value iteration is also the basis for many other techniques for solving MDP's and POMDP's. One of the most well-known and commonly used method for approximating the optimal policy of a POMDP, based on value iteration, is the *Witness algorithm* [17]. Other value function approximation algorithms have been proposed in [36, 55, 62, 63, 65] .

Another class of methods for solving POMDPs use heuristic control strategies. The *belief replanning* [5, 32] algorithm starts by assuming it is in the most likely state and generates a path to the goal according to that state. It also generates a sequence of predicted states that the robot will reach following that path. If at any time the robot's most likely state is not the predicted one, then the algorithm replans the path to the goal, according to the current most likely state.

There is a family of policies that operate in partially observable worlds, but sidestep the computational intractability of finding an optimal solution by using greedy heuristics. The *most likely state* (MLS) [31] policy finds the world state with the highest probability and executes the action that would be optimal for that state. The *voting method* [45] chooses the action with highest probability mass in the belief vector.

A number of heuristics attempt to model the effect of actions that push the belief state to the edges of the belief space, or *information gathering* actions. A heuristic like this is the Q_{MDP} [21], that is similar to the voting method. Two more information gathering heuristics have been presented in [5] that make use of the entropy of the belief state. Recently, it has been proposed to concatenate the MLS with the entropy [40] to form a set of jointly statistics that represents the belief state and thus be able to recover the true model.

Reinforcement learning methods such as Q-learning have also been used to solve POMDPs [37, 50, 49, 48, 51, 22, 13]. Finally, evolutionary algorithms for reinforcement learning can be used in conjunction with the traditional reinforcement learning techniques [27].

2.1.2 Learning POMDP's

Learning POMDPs involves determining the structure of the POMDP, initialize the probability matrices of that structure and iteratively adjust them so as to maximize the likelihood that the training data was generated by the model. The probability matrices are comprised by the transition probabilities, $\mathcal{T}(s, a, s')$, and the observation probabilities, $\mathcal{O}(s', a, z)$.

Learning POMDPs can be performed by means of general algorithms for learning probabilistic networks from data [67, 41, 66, 7, 28, 57, 15]. This approach for learning POMDPs has been used until now in applications for speech recognition and dialogue systems.

The Baum-Welch learning rule, based on the expectation-maximization (EM) algorithm, has also been applied to learning POMDP's in [8, 25, 20] for robotics applications. Baum-Welch in general does not converge to the true models that generated the training data [30]. To overcome this problem, algorithms that are based on state merging for learning POMDP's were developed. Such algorithms are the *best-first model merging* (BFMM) [46] and the *State Merging by Trajectory Clustering* (SMTC) [31].

2.2 Prediction Techniques

Early research in autonomous robot navigation assumed that robots operate in an environment with only stationary obstacles. Obviously, this assumption is unrealistic for real-world applications of robot navigation. Navigation methods that deal with moving obstacles and humans can be broadly classified into two major categories. In the first category, belong the navigation methods that when they detect moving obstacles avoid them using local planning techniques [12, 14, 43, 9, 35, 59, 26]. In the second category, methods that attempt to model and hence predict the future motion of obstacles and humans can be placed.

A first attempt to predict future positions of moving obstacles was to use an autoregressive (AR) model [19, 60]. Zhu [64], used a Hidden Markov Model (HMM) to describe and predict the motions of obstacles in dynamic environments. HMM's have also been used to model the movement of humans [52].Other stochastic models, such as random walk process with Gaussian distribution [29] have been used to model the motion of obstacles. Artificial Neural Networks (ANN's) have also been used to predict the future motion of obstacles [6, 61]. All the previously mentioned methods have made assumptions about the movement of obstacles and humans. Neural networks though have been successfully used even without any assumption about the movement of obstacles [34].

In this thesis, it is desirable to use a method that will predict the future motion of obstacles and humans without any pre-assumptions and in addition make use of this prediction in the global navigation method. Moreover, it is desirable to obtain an estimate of the velocity of the moving obstacles or humans. A proposed approach for future motion and velocity prediction is to use a neural network that can model highly non-linear behaviors such as movement. Furthermore, it is proposed that the topology and weights of the neural network to be determined via an evolutionary method. This approach allows to model satisfactory very complex behaviors [10]. The application of other methods (e.g.HMM's, non-linear stochastic models) will also be investigated to tackle the future motion and velocity prediction problem .

Prediction methods discussed so far give satisfactory results for one-step ahead prediction. For the robot navigation task it would be more useful to have many-steps ahead prediction. This would give the robot sufficient time and space to perform the necessary manoeuvres to avoid obstacles and more importantly change its route towards its destination position. It is desirable for the robot to develop a behavior that will prefer routes that are not crowded and thus avoid ever getting stuck, as any human would do.

It is unlikely that any available prediction method will give satisfactory results for many-steps ahead prediction, given the complexity of the movement behavior. To achieve this kind of behavior, it is proposed to make a long-term prediction. The long-term prediction refers to the prediction of a human's or an obstacle's final destination position. It is logical to assume that humans mostly do not just move around but with the intention of reaching a specific location. A possible approach for performing long-term prediction is to define "hot" points (points of interest) in the environment, where people would have interest in visiting them. For example, in an office environment desks, doors and chairs are objects that people have interest in reaching them and could be defined as points of interest. In a museum, the points of interest can be defined as the various exhibits that are present. Moreover, other features of the environment as the corner points, passages, e.t.c, can be defined as points of interest. If the points of interest of an environment are defined, then the long-term prediction could refer to the prediction of which "hot" point a human is going to approach.

Such a prediction can be accomplished in various ways. The "hot" points could be fully connected by e.g. straight line segments, and then the long-term prediction would refer to deciding which connection approximates the movement of a human. Another way would be to extrapolate the movement of a human by using appropriate line segments (straight lines or higher order curves) and estimate the "hot" point that is pursuing. Independently from the method that will be employed to perform long-term prediction, the past movement of a human is required to be tracked. Tracking the past movement of a human involves solving the correspondence problem in sensor readings at different time steps between the various moving obstacles and humans that are present. In order to effectively approach this task, the fusion of vision and range readings will be attempted.

It cannot be overlooked though that there will be cases that the long-term predictor will not be able to make decisions with great certainty or that a human moves randomly with no intention. In such cases, the one-step ahead prediction should prevail over other predictions made.

2.3 Multi-Robot Navigation

Using multiple robots to solve certain tasks can provide great benefits. Multi-robot systems though, add complexity to the navigation task because of the interference between robots, the communication and coordination cost.

There are several different forms of multi-robot systems. In *merely coexisting* systems, multiple robots coexist in a shared environment, but do not even recognize each other, treating other robots merely as obstacles. This is the simplest form, because there is no need for coordination, but also the least efficient: the more robots there are, the less effective the system is, since the robots must avoid collisions with each other.

In more sophisticated multi-robot systems, multiple (i.e., two or more) robots co-exist in the same environment and are aware of each other, but are *loosely coupled* in that they do not depend on each other for completing the task. This means they can react to each other in more interesting ways than just avoidance, but they do not directly help each other.

In even more sophisticated multi-robot systems, multiple robots actively cooperate with each other. If the robots depend on each other, their organization can be said to be *tightly coupled*.

Finally, multi-robot systems can be *competitive*. Competitive multi-robot systems are usually found in game scenarios as robot soccer. There is always though a kind of competition in multi-robot systems. There can be competition for physical resources as space and task resources as objects and goals.

There are two basic ways that multi-robot systems can be controlled: *centralized* and *distributed*. The centralized control approach makes use of a single, centralized, controller, which takes the information about all of the robots as input, and outputs the actions for all of them. In the distributed control approach, each robot can use its own controller to decide what to do.

The centralized approach has the major advantage that allows the computation of optimal solutions at the group level. This of course comes at the price of the great cost of gathering information and requiring global communication, that becomes extremely slow for a large number of robots.

On the other hand, the distributed control approach does not need central information gathering and the communication can be minimized or avoided. It also allows for the group size to change and is not affected if a single robot or a sub-group of robots fail to accomplish their task. The key disadvantage of the distributed approach is the difficulty in designing individual behaviors for each robot which will result in the desired group behavior.

Research in multi-robot systems so far has concentrated in learning group behaviors using behavior-based approaches(e.g. foraging, flocking, schooling) [24, 2, 3, 4, 47, 58] as well as in localization [11],mapping [44] and exploration [39, 44, 42]. Methods for local path-planning have also been proposed [33, 18, 1]. There seems to be no previous work in the multi-robot decision making problem for goal reaching in a global manner.

In this thesis, the multi-robot navigation problem will be approached for the case of having multiple robots in the same environment with one goal position, where only one robot is required to reach it. This problem will be tackled in a distributed and loosely coupled manner. The implementation is proposed to be in a probabilistic manner using POMDP's as in the single robot case although the application of Markov games to this problem will be also investigated. The use of Markov games is expected to provide a more elegant method for defining how the robots should coordinate so that one of them achieves the specified goal.

3 Proposed Approach

In this thesis it is proposed to use prediction of the future motion of obstacles and humans for safe and efficient autonomous robot navigation. The future motion prediction will be incorporated into the robot navigation model. The robot navigation model is proposed to be probabilistic and specifically a POMDP to deal with uncertainty. In addition, the coordination and communication between robots that operate in the same world and have the same goal will also be included in this model.

A POMDP is characterized by the tuple $M = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{Z}, \mathcal{O} \rangle$. To model the problem of predictive robot navigation with a POMDP:

- $\mathcal{S}$, the set of finite states, will consist of location, orientation and velocity. The location will correspond to a discrete square area of the occupancy grid map. Much of the criticism for the use of POMDP is that they are computationally inefficient when there is a large state space involved. Implementations so far for robot navigation with POMDP's used coarse discretization of the occupancy grid map to maintain the state space small [5, 45]. These implementations used one-square-meter discretization, and it is desirable to use a much finer discretization without large computational cost. For that reason, it is proposed to use a hierarchical representation of the POMDP model [53]. The hierarchical POMDP will have a tree structure where going from the top layer to the bottom, the resolution will change from coarser to finer.

- $\mathcal{A}$, the set of finite actions, will consist of the five basic actions: *move-forward, turn-left, turn-right, no-operation* and *declare-goal*. The *declare-*

goal action will be used by the robot to indicate that it has achieved its objective. The *no-operation* action will be used to indicate that the robot will perform no action. This finite set of actions could be further refined to define the angle of the turn that the robot should perform or how further it should move.

- $\mathcal{Z}$, the finite set of observations will contain features of the environment. Since observations are only used by the POMDP to determine the robot's true state these features should be reliable and facilitate the robot localization and velocity estimation. Such features could be straight line segments and corner points. A learning procedure could also be used to extract those features that are most suitable to form the set of observations.

- the state transition function, $\mathcal{T}$, and observation function, $\mathcal{O}$, will be determined from a learning procedure.

- the reward function, $\mathcal{R}$, is the element of the POMDP that this approach gives emphasis. It is the element of the POMDP that will actually control the movement of the robot and allows the POMDP to be used for global planning instead for a high-level mission planner as in previous approaches [5, 20, 54]. It will ensure that the robot avoids moving obstacles, humans and other robots and also that all the robots communicate and coordinate to achieve a specific goal.

Each square area in the occupancy grid map is characterized by an associated value. This value is the reward the robot receives for ending up in this state. Thus, it would be desirable that this value gives a description of the state of this square area in the environment as

- how far it is from the goal position

- whether it is occupied by a static obstacle

- whether it is occupied by a moving obstacle, human or other robot

- whether it will be occupied and how soon by a moving obstacle, human or other robot

It is proposed that the values of each cell in the occupancy grid map are determined in a similar manner to the approaches that use artificial potential fields. To save computation time it is also proposed that there are two types of artificial potential fields: a static and a dynamic potential field.

The static potential field will be evaluated once and will include the first two sources of information concerning the goal position and static obstacles. Various potential field functions have been proposed to determine the values of the occupancy grid map; the "numeric" potential field function [38] seems appropriate for our purposes. This potential field function labels all fully or partially occupied cells as occupied. The free cells are assigned the value that is determined from the city block distance between this cell and the goal. The goal cell is given the value of zero. This approach ensures that there are no other minima except at the goal position.

It is more interesting to see how the information about moving obstacles will be included in the dynamic potential field. The information whether a specific region of the occupancy grid map is occupied by a moving obstacle can be provided if at every time new sensor readings are obtained, they are superimposed with the initial map of the environment. Thus, objects that were not initially present can be regarded as moving obstacles. In addition, the information provided from the short-term and long-term prediction modules should be included in the dynamic potential field. Hence, the value of cell, p, in the dynamic potential field, DPF, can be given by a function

$$DPF(p) = extent^{\gamma},$$

where *extent* is a constant that controls how far the robot should stay from obstacles and γ is the factor that determines how much the value of *extent* should be discounted. The value of γ depends on the estimate of how far in the future this cell will be occupied and the relative velocity between the robot the obstacle that is estimated to occupy this cell. The dynamic potential field will be continuously updated and will be added with the static potential field to obtain the final reward function.

As previously mentioned, the POMDP model for predictive robot navigation will be extended for multi-robot navigation. To achieve multi-robot navigation there has to be a way for the robots to communicate and coordinate.

Communication of robots will be achieved by making some additions to the reward function of the POMDP model described previously. Each robot will regard any other robot as a regular moving obstacle and thus update the occupancy grid map that it holds in the way that was described before. This is an implicit method of communication between the robots regarding their position in the occupancy grid map. An explicit method of communication would be for each robot to inform all other robots at all times of its location.

For the coordination of the robots there is the need for a separate controller for each robot that will compare its excepted cumulative reward with the expected cumulative reward of all other robots.As mentioned before, when a goal is specified, many robots could start moving towards the goal position but finally only one should reach it. A robot should start moving towards the goal if its expected sum of rewards is higher than a threshold, that could be specified as a percentage of the highest expected sum of rewards. While a robot is in motion, it should stop its attempt to reach the goal position if its expected sum of rewards falls below a threshold, that could be again expressed as a percentage of the highest expected sum of rewards. This approach ensures that only the robot than can follow the easiest and most efficient way to the goal position will finally reach it.

3.1 Preliminary Results

A preliminary implementation of the proposed approach has been completed and results obtained are presented in this section for demonstration purposes. The current implementation constitutes a skeleton of the final implementation.

The workspace of the robot is an L-shaped corridor without any static obstacles. It is assumed that there is a single moving obstacle with its course of motion being a straight line. It is also assumed that the end point of the obstacle's straight line movement is the long-term prediction and at each time step the next point on this line is the short-term prediction. Finally, it is regarded that the moving obstacle and the robot move with the same constant velocity.

A POMDP is utilized to model the world that the robot operates and make a decision at each time step, in a global and probabilistic manner, regarding the action it should perform. The reward function of the POMDP, giving the reward for executing an action at a state, is initially defined by the Euclidean distance of the resulting state from the goal position. The reward function is updated at each time step according to the short-term and long-term prediction of the moving obstacle's movement as described in the previous section. The POMDP is solved by converting it to a "belief MDP" and then using the value iteration algorithm.

In the figures shown, the blue and red points represent the start and goal points of the robot, respectively. The green and orange points are the moving obstacle's start and end points, respectively. The path followed, for the same configuration of robot and obstacle positions, is shown in the case that the POMDP model was used to obtain the path and the case where a decision about the locally best-next action was made.

When local decisions are made, the robot moves in the NE direction, because the reward is the Euclidean distance from the goal point, until it reaches the column that the goal point lies and then moves in the N direction. In this example, shown in Figure 1, the movement of the obstacle did not affect the robot's movement because it could override the obstacle region before they meet.

On the other hand, when decisions are made by the POMDP, the robot's two initial movements are in the N direction and the remaining in the NE direction, as shown in Figure 2. This is because, in this example the reward the robot gains is the value function determined for a finite horizon length. Thus, the value function is the future reward for a finite sequence of actions. Therefore, the path obtained does not direct the robot to the obstacle region.

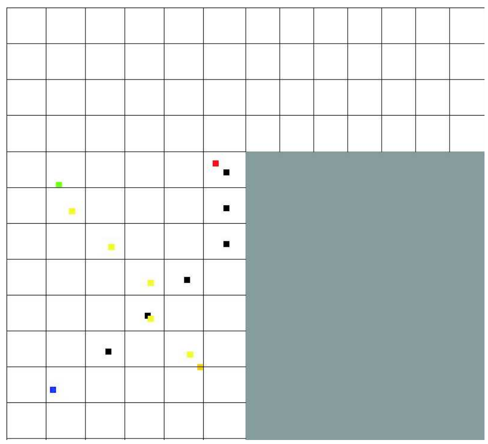

Figure 1: Example a - The path generated by local decisions

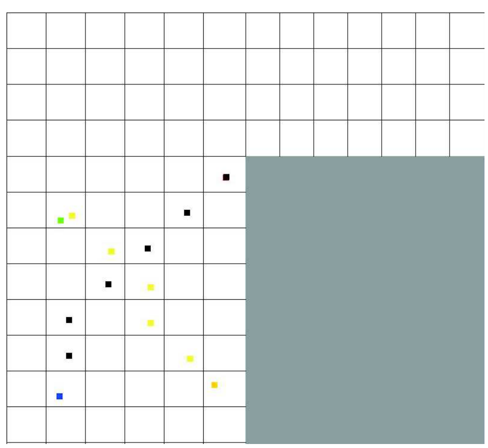

Figure 2: Example a - The path generated by the POMDP

In the second example shown in Figure 3, when local decisions are made, the robot again starts moving in the NE direction. It's third move, thought, is in the N direction since in the NE there is the short-term prediction of the obstacle's movement. Then, it follows a path similar with the one in the previous example since it has passed the obstacle region.

The path generated by the POMDP for the second example, shown in Figure 4, is equivalent to the path generated for the first example. The robot initially moves in the N direction and then to the NE direction until it reaches the goal position.

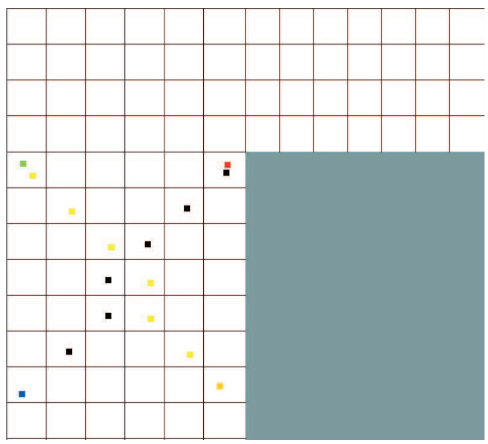

Figure 3: Example b - The path generated by local decisions

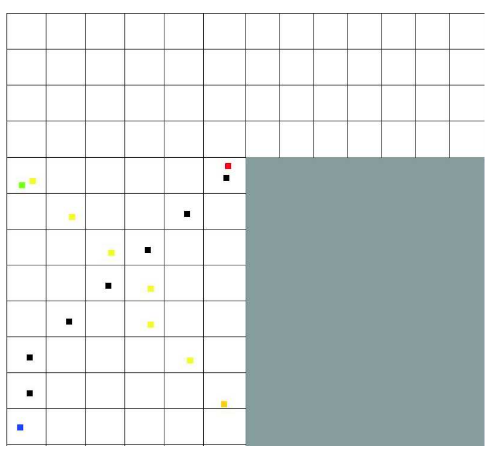

Figure 4: Example b - The path generated by the POMDP

The previous two examples demonstrate the intuition of the proposed approach, although the POMDP was solved for both examples, for a small horizon of length 3. That is, the robot from the beginning of its motion considers the prediction made about the future position of the obstacle. Therefore, it chooses to move towards the goal position by performing a sub-optimal local action (for the previous examples the action in the N direction), that will eventually lead to a global optimal path.

4 Conclusions

In this thesis it is proposed to use prediction techniques within a probabilistic framework for autonomous robot navigation in highly dynamic environments.

The prediction of the future motion and velocity of obstacles and humans will be performed without making any strong assumptions about their movement and this information will be integrated in the global navigation model. The future motion prediction is obtained by a short-term and a long-term prediction. The short-term prediction will refer to the next-time position of the obstacle. The long-term prediction will refer to the prediction of the final destination point of the obstacle's movement.

To deal with the uncertainty of the world and robot sensors, a probabilistic framework, expressed by a POMDP, is utilized. This thesis introduces the use of POMDP's for sequential decision making for goal reaching in a global manner in contrast to their use so far as high-level mission planners. This is achieved by integrating into POMDP's artificial potential fields. The integrated artificial potential field contains the information obtained from the short-term and long-term prediction.

Finally, this thesis provides a first treatment of the multi-robot decision making problem for goal reaching in a global manner that is still unexplored.

As such, the proposed approach provides a way to incorporate into the navigation model

- the intrinsic uncertainty of the robot, since the sensory and state information are modelled in a probabilistic way
- the prediction of the movement of humans or other moving obstacles or other robots
- the localization of the robot
- the communication and coordination with other robots

The chosen formulation thus results in a unified treatment of the predictive robot navigation problem and also to the multi-robot navigation problem.

References

[1] R. Alami. A multi-robot cooperation scheme based on incremental plan-merging. In *Seventh International Symposium on Robotics Research*. Springer, 1995.

[2] Tucker Balch and Ronald C. Arkin. Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation*, 1999.

[3] Tucker Balch and Maria Hybinette. Social potentials for scalable multi-robot formations. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA)*, 2000.

[4] David C. Brogan and Jessica K. Hodgins. Group behaviors for systems with significant dynamics. *Autonomous Robots*, 4:137–153, 1997.

[5] Anthony R. Cassandra, Leslie Pack Kaelbling, and James A. Kurien. Acting under uncertainty: Discrete bayesian models for mobile-robot navigation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996.

[6] Charles C. Chang and Kai-Tai Song. Dynamic motion planning based on real-time obstacle prediction. In *Proceeding of the 1996 IEEE International Conference on Robotics and Automation*, pages 2402–2407, 1996.

[7] Eugene Charniak. Bayesian networks without tears. *AI Magazine*, 12(4):50–63, 1991.

[8] Lonnie Chrisman. Reinforcement learning with perceptual aliasing: The perceptual distinctions approach. In *National Conference on Artificial Intelligence*, pages 183–188, 1992.

[9] P. Fiorini and Z. Shiller. Motion planning in dynamic environments using the relative velocity paradigm. In Lisa Werner, Robert; O'Conner, editor, *Proceedings of the 1993 IEEE International Conference on Robotics and Automation: Volume 1*, pages 560–565, Atlanta, GE, May 1993. IEEE Computer Society Press.

[10] Amalia Foka. Time series prediction using evolving polynomial neural networks. Master's thesis, University of Manchester Institute of Science and Technology, 1999.

[11] Dieter Fox, Wolfam Burgard, Hannes Kruppa, and Sebastian Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, 8(3), 2000.

[12] Kae Fujisawa, Soichiro Hayakawa, and Takeshi Aoki. Real time motion planning for control of autonomous mobile robot. In *IEEE/RSJ International Conference of Intelligent Robots and Systems*, pages 606–611, 1998.

[13] Mance E. Harmon and Stephanie S. Harmon. Reinforcement learning: a tutorial, 1996.

[14] C.S. Hong, S.M. Chun, J.S. Lee, and K.S. Hong. A vision-guided object tracking and prediction algorithm for soccer robots. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, pages 346–351, 1997.

[15] E. J. Horvitz, J. S. Breese, and M. Henrion. Decision theory in expert systems and artificial intelligence. *International Journal of Approximate Reasoning*, 2:247–302, 1988.

[16] Andrew Howard. *Probabilistic Navigation: Coping with Uncertainty in Robot Navigation Tasks*. PhD thesis, Department of Computer Science and Software Engineering, University of Melbourne, 1999.

[17] Leslie Pack Kaebling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134, 1998.

[18] S. Kato, S. Nishiyama, and J. Takeno. Coordinating mobile robots by applying traffic rules. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 1992.

[19] N. Kehtarnavaz and S. Li. A collision-free navigation scheme in the presence of moving obstacles. In *CVPR'88 (IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Ann Arbor, MI, June 5–9, 1988)*, pages 808–813, Washington, DC., June 1988. Computer Society Press.

[20] Sven Koenig and Reid G. Simmons. Unsupervised learning of probabilistic models for robot navigation. In *Proceedings of the International Conference on Robotics and Automation*, pages 2301–2308, 1996.

[21] Michael Littman, Anthony Cassandra, and Leslie Kaebling. Learning policies for partially observable environments: Scaling up. In *Proceeding of the 12th International Conference on Machine Learning*, pages 362–370, 1995.

[22] Michael L. Littman. *Algorithms for Sequential Decision Making*. PhD thesis, Department of Computer Science, Brown University, 1996.

[23] Michael L. Littman, Judy Goldsmith, and Martin Mundhenk. The computational complexity of probabilistic planning. *Journal of Artificial Intelligence Research*, 9:1–36, 1998.

[24] Maja J. Mataric. Reinforcement learning in the multi-robot domain. *Autonomous Robots*, 4(1):73–83, 1997.

[25] Andrew McCallum. Overcoming incomplete perception with utile distinction memory. In *International Conference on Machine Learning*, pages 190–196, 1993.

[26] Jun Miura, Hiroshi Uozumi, and Yoshiaki Shirai. Mobile robot motion planning considering the motion uncertainty of moving obstacles. In *IEEE SMC '99 Conference Proceedings in Systems, Man and Cybernetics*, pages 692–697, 1999.

[27] David E. Moriarty, Alan C. Schultz, and John J. Grefenstette. Evolutionary algorithms for reinforcement learning. *Journal of Artificial Intelligence Research*, 11:199–229, 1999. http://www.ib3.gmu.edu/gref/papers/moriarty-jair99.html.

[28] R. Charles Murray, Kurt VanLehn, and Jack Mostow. A decision-theoretic approach for selecting tutorial discourse actions. In *Proceedings of the NAACL Workshop on Adaptation in Dialogue Systems*, pages 41–48, 2001.

[29] Yun Seok Nam, Bum Hee Lee, and Moon Sang Kim. View-time based moving obstacle avoidance using stochastic prediction of obstacle motion. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, pages 1081–1086, 1996.

[30] Daniel Nikovski. Learning stationary temporal probabilistic networks. In *Conference on Automated Learning and Discovery*, 1998.

[31] Daniel Nikovski and Illah Nourbakhsh. Learning probabilistic models for decision-theoretic navigation of mobile robots. In *Proc. 17th International Conf. on Machine Learning*, pages 671–678. Morgan Kaufmann, San Francisco, CA, 2000.

[32] I. Nourbakhsh, R. Powers, and S. Birchfield. Dervish: An office-navigating robot. *AI Magazine*, 16(1):53–60, 1995.

[33] Spence Oliver, Mahesh Saptharishi, John Dolan, Ashitey Trebi-Ollennu, and Pradeep Khosla. Multi-robot path planning by predicting structure in a dynamic environment. In *Proceedings of the First IFAC Conference on Mechatronic Systems*, volume II, pages 593–598, September 2000.

[34] J. Gomez Ortega and E. F. Camacho. Mobile robot navigation in a partially structured static environment, using neural predictive control. *Control Engineering Practice*, 4(12):1669–1679, 1996.

[35] Erwin A. Pabler and Evangelos E. Milios. Motion planning amongst arbitrarily moving unknown obstacles. In *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, 1994.

[36] R. Parr and S. Russell. Approximating optimal policies for partially observable stochastic domains. In *Proceedings of the 14th International Joint Conferences on Artificial Intelligence*, 1995.

[37] Leonid M. Peshkin. Architectures for policy search. Proposal for PhD thesis, 2001.

[38] Steven Ratering and Maria Gini. Robot navigation in a knownn environment with unknown moving obstacles. *Autonomous Robots*, 1(2):149–165, 1995.

[39] Ioannis M. Rekletis, Gregory Dudek, and Evangelos E. Milios. Graph-based exploration using multiple robots. In *5th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, pages 241–250, 2000.

[40] Nicholas Roy. Finding approximate POMDP solutions through belief compression. PhD Thesis Proposal, Carnegie Mellon University, 2000.

[41] Stuart Russell, John Binder, and Daphne Koller. Adaptive probabilistic networks. Technical Report CSD-94-824, University of California, Berkeley, July 1994.

[42] Antonio Sgorbissa and Ronald C. Arkin. Local navigation strategies for a team of robots.

[43] Rajeev Sharma. Locally efficient path planning in an uncertain, dynamic enviroment using a probabilistic model. *IEEE Transactions on Robotics & Automation, Feb. 1992*, 8(1), 1992.

[44] Reid Simmons, David Apfelbaum, Wolfram Burgard, Dieter Fox, Mark Moors, Sebastian Thrun, and Håkan Younes. Coordination for multi-robot exploration and mapping. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, 2000.

[45] Reid Simmons and Sven Koenig. Probabilistic robot navigation in partially observable environments. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1080–1087, 1995.

[46] Andreas Stolcke and Stephen M. Omohundro. Best-first model merging for hidden Markov model induction. Technical Report TR-94-003, Computer Science Division, University of California at Berkeley and International Computer Science Institute, 1994.

[47] Peter Stone and Manuela Veloso. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence*, 100.(2), 1999.

[48] Ron Sun. Introduction to sequence learning. In Ron Sun and Lee Giles, editors, *Sequence Learning: Paradigms, Algorithms, and Applications*, pages 1–10. Springer-Verlag, 2000.

[49] Ron Sun and Todd Peterson. Autonomous learning of sequential tasks: Experiments and analyzes. *IEEE Transactions on Neural Networks*, 9(6):1217, 1998.

[50] Ron Sun and Chad Sessions. Learning to plan probabilistically from neural networks. In *Proceedings of IEEE International Conference on Neural Networks*, pages 1–6, 1998.

[51] Ron Sun and Chad Sessions. Learning plans without a priori knowledge. *Adaptive Behavior*, 2001.

[52] Satoshi Tadokoro, Masaki Hayashi, and Yasushiro Manabe. On motion planning of mobile robots which coexist and cooperate with human. In *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 518–523, 1995.

[53] Georgios Theocharous, Khashayar Rohanimanesh, and Sridhar Mahadevan. Learning hierarchical partially observable markov decision processes for robot navigation. In *IEEE Conference on Robotics and Automation*, 2001.

[54] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive musuem tour-guide robot minerva. *International Journal of Robotics Research*, 19(11):972–999, 2000.

[55] Sebastian Thrun. Monte carlo POMDPs. In *Proceedings of Conference on Neural Information Processing Systems (NIPS)*, 1999.

[56] Sebastian Thrun. Probabilistic algorithms in robotics. *AI Magazine*, 21(4):93–109, 2000.

[57] Carter Wandelken. Learning how to act: A structured connectionist approach. PhD thesis proposal, 2000.

[58] Barry B. Werger. Cooperation without deliberation: A minimal behavior-based approach to multi-robot teams. *Artificial Intelligence*, 110:293–320, 1999.

[59] Xinxin Yang, Kezhong He, Muhe Guo, and Bo Zhang. An intelligent predictive control approach to path tracking problem of autonomous mobile robot. In *IEEE Conference on Systems, Man and Cybernetics*, volume 4, pages 3301–3306, 1998.

[60] N. H. C. Yung and C. Ye. Avoidance of moving obstacles through behavior fusion and motion prediction. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 3424–3429, 1998.

[61] Nelson H. C. Yung and Cang Ye. An intelligent mobile vehicle navigator based on fuzzy logic and reinforcement learning. *IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 29(2):314–321, April 1999.

[62] Nevin L. Zhang and Wenju Liu. A model approximation scheme for planning in partially observable stochastic domains. *Journal of Artificial Intelligence Research*, 7:199–230, 1997.

[63] Nevin L. Zhang and Weihong Zhang. Speeding up the convergence of value iteration in partially observable markov decision processes. *Journal of Artificial Intelligence Research*, 14:29–51, 2001.

[64] Q. Zhu. Hidden markov model for dynamic obstacle avoidance of mobile robot navigation. *IEEE Transactions on Robotics and Automation*, 7(3):390–397, June 1991.

[65] Valentina Bayer Zubeck and Thomas G. Dietterich. A POMDP approximation algorithm that anticipates the need to observe. In *Pacific Rim International Conference on Artificial Intelligence*, pages 521–532, 2000.

[66] Geoffrey Zweig and Stuart Russell. Probabilistic modeling with bayesian networks for automatic speech recognition. *Australian Journal of Intelligent Information Processing Systems*, 5(4):253–60, 1999.

[67] Geoffrey G. Zweig. *Speech Recognition with Dynamic Bayesian Networks*. PhD thesis, University of California, Berkeley, 1998.