

Predictive Autonomous Robot Navigation

Amalia F. Foka and Panos E. Trahanias

*Institute of Computer Science, Foundation for Research and Technology-Hellas (FORTH), Heraklion, Greece
and Department of Computer Science, University of Crete, Heraklion, Greece, {foka,trahania}@ics.forth.gr*

Abstract

This paper considers the problem of a robot navigating in a crowded or congested environment. A robot operating in such an environment can get easily blocked by moving humans and other objects. To deal with this problem it is proposed to attempt to predict the motion trajectory of humans and obstacles. Two kinds of prediction are considered: short-term and long-term. The short-term prediction refers to the one-step ahead prediction and the long-term to the prediction of the final destination point of the obstacle's movement. The robot movement is controlled by a Partially Observable Markov Decision Process (POMDP). POMDPs are utilized because of their ability to model information about the robot's location and sensory information in a probabilistic manner. The solution of a POMDP is computationally expensive and thus a hierarchical representation of POMDPs is used.

1 Introduction

For mobile robots, navigation in dynamic, real-world environments is a complex and challenging task. Such environments are characterized by their complex structure and the movement of humans and objects in them. In such cases, a navigating robot can easily get blocked by moving humans and obstacles and may become immobilized and not able to continue its movement towards its goal position until the moving objects free its way. To avoid ever getting in such a situation, many researchers have tried to predict the motion of humans and obstacles. Future motion prediction 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.

Acknowledgments

This work was partially supported by EC Contract No IST-2000-29456 (WebFAIR <http://www.ics.forth.gr/webfair>) under the IST Programme.

Future motion prediction is an intrinsic behavior of humans. Consider the example of a man trying to cross a street. The man tries to estimate how long it will take for a vehicle to reach the point that he stands and then decides if he should cross the street.

Generally speaking, a robot has to make decisions about the actions it should perform at each time step considering the information its sensors provide about the environment state. 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). 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 paper 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 been only used in local collision avoidance modules [7, 5, 2, 9, 6]. In this paper, we make use of the prediction information in the global navigation model. Moreover, the POMDP model is used to formulate the navigation task because it provides a probabilistic way of representing information and also solves the localization problem. POMDPs have been previously used in robotics but only in either simple environments [1, 3] and as a high-level mission planner with a separate module for collision avoidance [8]. This approach proposes the use of a POMDP model that integrates the localization and collision avoidance with the use of future motion prediction modules for autonomous robot navigation.

The rest of this paper is organized as follows. In section 2, the modelling of the predictive robot navigation problem is given. In sections 3 and 4, the short-term and long-term prediction is described and in section 5 the integration of prediction into the global model is illustrated. Experimental results are given in section 6 and the paper concludes in section 7.

2 Partially Observable Markov Decision Processes (POMDPs)

POMDPs 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.

A POMDP is a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{Z}, \mathcal{O} \rangle$, where [4]:

- \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{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.
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R}$ is the *reward function*, giving the expected immediate reward gained by the agent for taking each action in each state, $\mathcal{R}(s, a)$.

2.1 Problem Formulation and Hierarchical POMDPs

The tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{Z}, \mathcal{O} \rangle$, of the POMDP is instantiated in our formulation as:

- \mathcal{S} , each state in \mathcal{S} corresponds to a discrete square area of the environment’s occupancy grid map (OGM).
- \mathcal{A} , consists of the basic motion actions: *North, South, North-West, South-West, West, North-East, South-East* and *East*.
- \mathcal{T} is defined by setting the probability $\mathcal{T}(s, a, s')$ to 0.9, when s' is the correct resulting state given action a and previous state s . The probability

of 0.04 is assigned when the resulting state is an immediate neighbor of s' , and the probability of 0.01 is assigned to any other neighboring state of s' . Zero probability is assigned to all other states. Figure 1, illustrates the transition probabilities assigned for a state s , given that the action is *North*. The correct resulting state is s' . The assignment of transition probabilities for other actions is performed in a similar manner.

0.04	0.9 s'	0.04
0.01	s	0.01
0	0	0

Figure 1: Assignment of transition probabilities.

- \mathcal{Z} and \mathcal{O} , are the elements of the POMDP that assist in the localization of the robot, that is the determination of the belief state after an action has been taken, i.e. the probability of all states that the robot occupies them. In this work we employ a simulated environment, and hence the state which the robot occupies is always known with certainty. Therefore, these two elements are not crucial in our current implementation. \mathcal{Z} and \mathcal{O} are defined though to be able to solve the POMDP. Thus, \mathcal{Z} consists of two observations: goal position and “nothing”. $\mathcal{O}(s', a, z)$, is defined by assigning the value of 1 if after taking action a the resulting state is the goal state. Otherwise, it is assigned the zero value.
- The initial belief state, $p_0(s : s \in \mathcal{S})$, is defined by assigning the probability 0.9 to the state that corresponds to the robot’s start position. Neighboring states are assigned probabilities that sum up to 0.1. Zero probability is assigned to all other states.
- \mathcal{R} , is the element of the POMDP that this work gives most emphasis. The reward function 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 as a high-level mission planner as in previous approaches [1, 3, 8]. It will ensure that the robot avoids moving obstacles, humans and other robots and also moves predictively according to the independent motion of other objects. The determination of the reward function is discussed in detail in section 5.

Hierarchical POMDPs. Much of the criticism for the use of POMDPs is that they are computationally inefficient when there is a large state space involved. This is the main reason that POMDPs have been utilized until now only as high-level mission planners. Implementations so far for robot navigation with POMDPs used coarse discretization of the OGM to maintain the state space small [1]. Such implementations used typically one-square-meter discretization, and it is desirable to use a much finer discretization without large computational cost. For this reason, a hierarchical representation of the POMDP model is used in this work.

Our approach to hierarchy is to decompose a POMDP with large state space into multiple POMDPs with significantly smaller state space. The hierarchical POMDP has a tree structure, where going from the top level to the bottom, the resolution changes from coarser to finer and the number of POMDPs is increased. At the top level there is a single POMDP with coarse resolution. At the next level each state of the POMDP at the top level is divided into four states and a new POMDP is constructed having as state space these four states. Thus, each state in the top level corresponds to a POMDP in the next level. In the same manner, lower levels are constructed until we have reached the level that its states are the states of the original flat POMDP. Figure 2 illustrates the basic concept of hierarchical POMDPs. In this figure, it is shown how a state at the top level is decomposed to multiple states in subsequent levels.

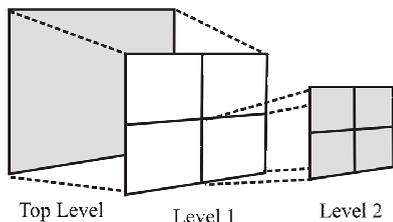


Figure 2: The decomposition of states in the hierarchical POMDP.

Planning with Hierarchical POMDPs. The solution of a POMDP is a policy, i.e. a mapping from each “belief state” (probability distribution over states) to actions that maximize the long-term sum of reward the robot receives. Initially, the POMDP is solved at the top level. The solution of the top level POMDP gives intuitively the general route the robot should follow to reach the goal. Following, the POMDP at the lower level that contains the robot’s current state is solved with its goal being to follow the direction that the solution of the

upper-level POMDP provided. In the same manner, POMDPs at all levels are solved until the bottom level is reached. The action obtained from the solution of the bottom level POMDP is the one that the robot will actually perform.

3 Short-Term Prediction

The short-term prediction of the future motion, i.e. the one-step ahead prediction, is obtained by a Polynomial Neural Network (PNN). In PNNs each node’s transfer function is a polynomial. PNN’s can model any function since the Kolmogorov-Gabor theorem states that any function can be represented as a polynomial of the form $f(x) = a_0 + \sum_i a_i x_i + \sum_i \sum_j a_{ij} x_i x_j + \sum_i \sum_j \sum_k a_{ijk} x_i x_j x_k + \dots$, where x_i is the independent variable in the input variable vector \vec{x} and \vec{a} is the coefficient vector. PNNs approximate the Kolmogorov-Gabor polynomial representation of a function by having as a transfer function at each node, a second order polynomial of the form $f(x) = a + bx_1 + cx_2 + dx_1^2 + ex_2^2 + f x_1 x_2$, where x_1 and x_2 are the inputs to the node. The input to the network, x_1 and x_2 , is the moving obstacle’s position at times $t - 1$ and t . The positions of the moving obstacle are given to the network as a state. The output of the network is the predicted position at time $t + 1$. The topology and weights, that is the polynomial coefficients, of the PNN are determined via training with an evolutionary method. The training set used was composed of 4500 samples and is shown in Figure 3. It was obtained by arbitrary movement in the environment that the robot operates. The re-

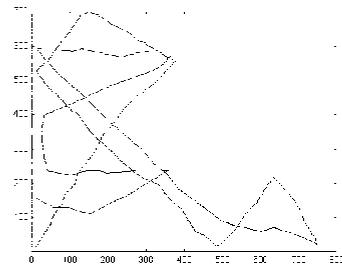


Figure 3: The data set used for NN training.

sults obtained from the trained NN are illustrated in Figure 4. The first 3000 samples from the data set were used for training and the rest 1500 samples for evaluating the network. It can be seen that the network gives a prediction with a small error for the first 3000 samples but as well for the rest of the samples that the network has not seen before. Therefore, the network obtained generalizes well for unforeseen situations.

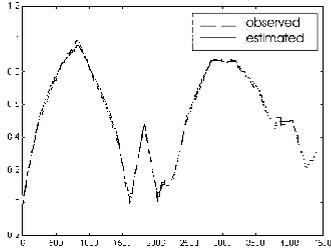


Figure 4: The results obtained from the trained NN.

4 Long-Term Prediction

Prediction methods known 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.

It is unlikely that any available prediction method would 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 employ a long-term prediction mechanism. The long-term prediction refers to the prediction of a human’s or an obstacle’s final destination position. It is plausible to assume that humans mostly do not just move around but with the intention of reaching a specific location. Our approach for performing long-term prediction is based on the definition of the so-called “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 such as the entry points, passages, e.t.c, can be defined as points of interest. Evidently, points of interest convey semantic information about a workspace and hence can only be defined with respect to the particular environment. In this work, such points are defined manually on the internal representation (map) of the environment. Appropriate semi-automated methods for their definition can also be considered, an issue beyond the scope of this paper.

Once the points of interest of an environment are defined, then the long-term prediction refers to the prediction of which “hot” point a moving obstacle is

going to approach. At each time step t , the tangent vector of the obstacle’s positions at times $t-1$, t and the predicted position at time $t+1$ is taken. This tangent vector essentially determines the global direction of the obstacle’s motion trajectory, termed as Global Direction of Obstacle (GDO). This direction is employed to determine which “hot” point a moving obstacle is going to approach. A “hot” point is probable to be the obstacle’s final destination point if it lies roughly in the direction of the evaluated tangent vector. In order to find such “hot” points, we establish a *field-of-view*, that is an angular area centered at the GDO. “Hot” points present in the *field-of-view* are possible points to be reached, with a probability w_i , according to a probability model. The latter is defined as a gaussian probability distribution with center at the GDO and standard deviation in proportion to the angular extent of the *field-of-view*. Thus, points of interest present in the center of the *field-of-view* are assigned a high probability, and points of interest present in the periphery of the *field-of-view* are assigned a lower probability. With this approach, at the beginning of the obstacle’s movement a multiple number of points of interest will be present in its *field-of-view* but as it continues its movement the number of such points is decreased and finally it usually converges to a single point of interest.

5 Integration of Prediction in the Global Model

The short-term and long-term prediction are integrated in the global model by including them in the reward function of the POMDP. Each square area in the OGM 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, i.e. human or other robot,
- whether it will be occupied and how soon by a moving obstacle.

To save computation time, two types of OGMs are used: a static and a dynamic grid map. The static OGM is evaluated once and includes the first two sources of information concerning the goal position and static obstacles.

Information provided from the short-term and long-term prediction modules is included in the dynamic

