PLANNING FOR SIMULTANEOUS ROBOT LOCALIZATION AND TASK

PAULO PINHEIRO*, JAQUES WAINTER*

*Institute of Computing,
University of Campinas - UNICAMP
6176, Campinas - SP, Brazil

Emails: pinheiro@ic.unicamp.br, wainer@ic.unicamp.br

Abstract— This paper will present a robot localization model with planning and task execution support. Localization models are constantly explored, however how the robots act on the environment is generally defined only by localization actions (actions that maximize the pose estimation) or actions to achieve the goal. This work describes a proposal for robot localization with planning of actions, which also considers the final task. The objective is to describe a model where policies define the best action to be executed by robots. The best action is defined by the combination of two vectors. First vector is defined by the localization action (that maximizes the robot’s localization). The second is defined by the goal of the task (arrive at a particular point on the map). The combination of these vectors indicates the best action. Thus, the robot is able to locate while reaches its destination. We compare the proposed model with a model that considers only the goal (without localization planning) and with a model that considers separately the localization and task. The results showed that proposed model is able to estimate the positions of robots and complete the task with lower number of steps, being more efficient that the models compared.

Keywords— Markov Localization, Planning, POMDP, Mobile Robotics.

1 Introduction

The robot localization problem consists in estimating the pose of each robot in a certain environment. The pose of a robot is represented by the position and heading in which the robot is found in relation to the environment (x, y, θ).

It is fundamental to know the pose (or at least to have a good estimate) for tasks such as objects manipulation, exploration and development of navigation plans. Even to make use of a map, it is necessary to know the self-location of the robot. It can be stated, therefore, that the self-localization is a prerequisite for many of the tasks performed by mobile robots and this has a conditional relation to the final results. For these reasons, self-localization becomes one of the most important capabilities of a mobile robot (Cox, 1991).

However, first locate and then execute the mission is not always the best alternative. Often it is necessary that these two tasks are executed in parallel, even if there is a precondition relationship (Fox et al., 2000).

Currently, many works studied cooperation in the localization problem, which is explored by communication between the robots. Many models of detection and transmission of information were devised in an attempt to improve the performance of the localization process. However, they do not consider other tasks.

The proposal of this work is to present a planner for simultaneous localization and execution of robot tasks. In this model, actions are defined by policies, allowing the robot to maximize its pose definition while moving towards the goal. This way, planned actions produce more useful observations, improving the quality of the estimates and allowing the robot to locate faster in unknown environments, while considering the final task. So, in this paper, we attempt to relax this precondition relationship between localization task and final tasks.

2 Related Work and Objective

The problem of the precondition between localization and task was mentioned by Kadous et al. (2006) and Fox et al. (2000). For Murphy (2004), this problem is so relevant that it became one of the reasons why Human-Robot Interaction (HRI) is so important for solving practical localization problems due to relative inefficiency of the robot.
to locate and execute the mission at the same time.

Regarding the matter of robot global localization, previous research have focused on the exploration of communication techniques among the agents or action planning for localization. The final task of robot is usually performed after the localization is found. In Fox et al. (2000) the global localization problem was solved with a single robot, so no communication, using the Markov localization and without planning actions. The selection of actions was random and a mission could only be performed after the localization. In Odakura (2006), multiple robots using communication were used, which made the localization process more efficient, even keeping the random actions.

To further improve the estimates in Markov localization, Hoffman et al. (2005) described the model of Negative Information, in which the absence of observation of landmarks was used to assist in the definition of robot pose. Thus, if the expected landmark was not found then there is the certainty that the robot is not in that location (Hoffman et al., 2006).

All this previous research focused on exploring techniques of sharing information but very little has been done to improve the quality of information that is shared. Pinheiro and Wainer (2010) described a planner for robot localization, where problems were modeled in POMDP, creating policies to define what action would produce an observation, that when used in the calculation of estimates, it would be more useful to define the robot posture, enabling fewer steps need be taken. The main issue about this work is that it assumed that localization is a goal on itself. Zhou and Sakane (2008) presented a similar work, but with partial planning.

In this context, this paper proposes a localization and task model that is able to select the best action, not only, for localization, but combined with the best action for accomplishing the task. This is called Model of Combined Planning or MCP. The objective of MCP is solve the problem of global robot localization using planning of actions, without disregarding the task to be performed.

For each scenario, a policy is generated for the agent select the best action in each decision period for localization. This action is combined with the best action for accomplishing the task. The combination of two vectors results in the final action. This action is one that maximizes the chances of obtaining a relevant observation able to improve the distribution of probabilities of the agent’s pose and accomplish the final task.

### 3 Localization x Final Task

In problems where the location is pre-condition of another task, these two processes are usually executed in sequence. Therefore, the robot performs the localization process and when your pose is defined, be initiated to the main task.

We assume that the main task of the robot is to reach at a particular point on the map, as show in Figure 1a and that the robot does not know its location.

![Figure 1: (a) Main task, (b) Optimal localization path, (c) Matching localization process and task](image)

The aim of this work is to find a location path, that does not necessarily be as good as optimal localization path, but that is compatible with the goal. Figure 1c describes this example. Note that the resultant path is longer than the optimal location path, but is more compatible with the goal. Therefore, the robot would be located and accomplish the task in less steps.

### 4 Localization and Task Problem Modeling

Before producing a plan for the robot, it is necessary to model the problem. In this phase, several forms of modeling were tried to make it possible to produce efficient policies. For this, we applied the formalism of POMDP (Partially Observable Markov Decision Process).

POMDP model is a tuple $\langle S, A, T, R, W, O \rangle$, where $S$ is a finite set of states, $A$ is a finite set of actions, $T$ is a Markovian transition probability, $R$ is a reward function, $W$ is a finite set of observations and $O$ is a table of observations probabilities (Theocharous et al., 2004).
4.1 Localization problem

As a localization problem, we developed a mechanism that could represent such a problem. The mechanism allows for the robot to perform a special action, which we name shout, when it notices that it already knows its position in the environment. This action will reward it positively when the estimate of pose presented is correct and negatively when it is wrong (Pinheiro and Wainer, 2010). The weights for these rewards were tested exhaustively.

Thus, the modeling allows you to choose the best action that maximizes the chances that the robot be located in less steps. In this work, we used this technique to the localization process.

4.2 Planning for the localization and tasks

If we apply only the localization technique described, the policy would consist of an optimal sequence of steps for the localization. However, in this work we need to consider the task of the robot, its mission, not only to find its pose.

For this, we propose a technique of combining vectors. The first vector to be considered is the vector of location. This vector is found using the location technique described in Pinheiro and Wainer (2010). The second vector considers the robot’s final task.

To continue with the explanation of the technique, we need to define what will be a task. In this work, a task is to reach a particular point \((x, y, \theta)\), also called the goal point.

To demonstrate the modeling with the combination technique, we will use an example of a localization and task problem.

Example 1 (map 3x3) A robot is in an environment of 3x3 size, made up of cells of 1x1 size; the environment presents an obstacle and 8 free cells, of which one of these cells is where the robot is placed (start) and the other is the goal (X), as shown in Figure 2a;

![Figure 2a](image)

Figure 2: Start and Goal positions and localization vectors

The robot does not know its starting position, needing to find its position and achieve the goal. In this case, the technique of combined planning performs the following procedures:

Step 1: Calculate \(\overrightarrow{L}\): First of all, the best action for the localization process is calculated using the localization planner presented in the work cited (Pinheiro and Wainer, 2010).

Therefore, each cell of the probability distribution contains a vector \(\overrightarrow{L}\) indicating the direction that maximizes the robot’s pose, as shown in Figure 2b.

Step 2: Calculate \(\overrightarrow{G}\): We assume that the robot’s position is the cell that has the highest probability. For example, suppose that cell is \((3,3)\). For this cell is calculated the vector that is pointing towards the goal \((\overrightarrow{G})\) as shown in Figure 3a.

![Figure 3b](image)

Figure 3: Definition and scaling of the vectors

Step 3: Resize \(\overrightarrow{L}\): The next step is to resize the localization vector. The size of the vector \(\overrightarrow{L}\) is inversely proportional to the localization probability, \(P(l)\), in that state. In other words, \(|\overrightarrow{L}| = (1 - P(l))\). So, if the robot is uncertain about its pose, for example, \(P(l) = 0.1\), then \(|\overrightarrow{L}| = 0.9\). If the robot is quite sure about its pose, for example, \(P(l) = 0.9\), then the vector \(|\overrightarrow{L}| = 0.1\).

This way, much less certainty about the position, the greater the effect of the localization vector \(L^{-}\). And therefore, the greater certainty about the pose, the lower the influence of the localization vector.

Step 4: Resize \(\overrightarrow{G}\): The next step is to resize the goal vector \((\overrightarrow{G})\), calculated as \(|\overrightarrow{G}| = P(l)\). Thus, when the robot’s pose is uncertain, the vector of goal will have less influence. If \(P(l)\) is high, the robot has a good estimate of where to be and therefore can proceed to the goal with more certainty.

In summary, if \(P(l)\) is very low, \(|\overrightarrow{L}| = 1 - P(l)\) will be high and \(|\overrightarrow{G}| = P(l)\) will be low. Otherwise, if \(P(l)\) is high, \(|\overrightarrow{L}|\) will be low and \(|\overrightarrow{G}|\) will be high, as shown in Figure 3b.

Step 5: Combine vectors Finally, we combine the vectors, as shown in Figure 3c. The resulting vector \(\overrightarrow{C}\) defines the action to be taken.

We assume that the initial pose probability of robot is the same in all the states of the environment. This modeling using this technique, allows the robot to look for the greatest reward and therefore make a correct localization of the environment without disregarding the mission of the robot. The more lost, more of the location vector will be important. If the robot’s certainty about its location is high, greater the influence of the goal vector.

Once the problem has been modeled for lo-
calization, it is possible to determine an optimum policy of actions by resolving the corresponding POMDP.

5 Experiments

In this section, we present experiments to compare the proposed model of planning with other two models.

5.1 Task-oriented model

The first model to be compared not use planning. In this model, the priority is to go toward the goal (Odakura, 2006). To choose the next action, the robot uses only its current position (state with highest probability) and location of the goal. First, assume that the robot is in position of highest probability and chooses the best action to reach the goal in less steps. While moving in the environment toward the goal, the robot observes the environment, modifies its probability distribution and restarts the process (Fox et al., 2000).

With more observations, the better the quality of the probability distribution and with that, improves the chances of finding its location. This model does not use planning, and by the localization’s view, the sequence of action may appear random.

5.2 Localization-oriented model

The second model prioritizes the localization process. His lemma is “first locate, then accomplishing the task”. This model uses a POMDP planner to determine the best action to take. In contrast with task-oriented model, in this model the best action is not one that would lead to the goal. In this model, the best action is that which maximizes the robot’s pose, i.e, makes the robot find its location more quickly. For this we use the planner for robot localization presented in (Pinheiro and Wainer, 2010).

5.3 Environments

The robot and environments characteristics are similar to the ones found in the research by Fox et al. (2000), Odakura (2006). The robots have a proximity sensor that can measure distances to an obstacle up to 15 cells ahead. The robots have a localization sensor that identifies other robots up to 17 cells of distance.

The environments used in this experiment are shown in Figure 4. The lighter areas represent the free spaces and the dark, obstacles. The environments are divided into grades, and the sizes of data per unit cell.

The first environment (a) is 15x15, representing two rooms and a hall in common. The environment (b) measures 20x20 and has a larger area free of obstacles. The environment (c), also found in the Fox et al. (2000) work, is 30x30.

5.4 Simulation

In the simulator, we select the environment and the goal point. The robot knows the environment map, however its initial pose is not known (global problem). The robot has a probability distribution that identifies a different value of pose belief for each state of the environment.

The initial point of the robot and goal point are chosen randomly. Each comparative model and proposed model are simulated using the same map, the same initial point and the same goal point.

6 Results

In this section we present results comparing the performance between the proposed model (Model of Combined Planning - MCP) with the two comparative models: Task-oriented Model (ToM) and Localization-oriented Model (LoM).

We calculate the number of steps taken and distance to the goal position. The number of steps is the number of times a robot updates its beliefs, so any action, that generates a new update, will be recorded as a step. After each step, the algorithm measure the distance of the shortest path to the goal position.

6.1 An Example

In the environment (a), the robot using Localization-oriented Model (LoM) was the first to be located (step 10) being 16 units away from the goal, completing the task with 26 steps. The robot using the Task-oriented Model (ToM), completed the task in 25 steps as shown in Figure 5.

The robot that used the Model of Combined Planning (MCP), was located more slowly than with LoM, with two more steps. However, when located, it was closer to the goal, completing the task with 17 steps.

In the environment (b), as shown in Figure 6, the robot with LoM was located in only 24 steps from a distance of 27 units from the goal. Completing the task with 51 steps. Using ToM, the location was determined in step 44, completing the task with 63 steps. Used MCP, the location was performed in 30 steps (6 steps more than the
LoM), but when it happened, the robot was only 9 steps from the goal. Completing the task with 39 steps.

The Figure 7 shows the results for the environment (c). In this environment, the comparative results were similar to the results of other maps. The robot using MCP is located only in step 42, but what’s interesting is that from step 20 was already headed for the destination, completing the task with 52 steps. Again, the robot with LoM was located more quickly, with 35 steps, but completing the task in 74 steps, against 124 steps of the robot with ToM.

6.2 Summary for All Cases

To compare the three models we count the number of steps until the task was completed. We also calculated the average gain of the Model of Combined Planning (MCP) in relation to the comparatives models (ToM and LoM).

For each map, 30 simulations were performed (10 using a robot with ToM, 10 with LoM and 10 with MCP). The starting point of the robot and the goal point was the same for each round. We calculated the p-value of t-test for each set of samples. One robot was used in each simulation.

Table 1 reports the number of steps for the simulations for environment (a). The pairwise t-test between MCP and ToM shows that the p-value was 0.001, and between MCP and LoM, the p-value was 0.000 that is, with more than 99% of confidence, the differences between the MCP and the comparative models are significant. The average gain for the MCP compared with ToM was 45%, in other words, using MCP, the agents were completed the task with a number of steps, 45% lower than the amount required with ToM. The average gain for the MCP compared with LoM was 49%.

For the environment (b), again the p-value was 0.000 to both comparative models, and thus with more than 99% confidence, the results are significantly different and the average gain for MCP was approximately 51% when compared with ToM and 33% when compared with LoM. Table 2 shows the number of steps until the task is completed.

Table 1: Number of steps to complete the task in the environments (a)

<table>
<thead>
<tr>
<th>Environment (a)</th>
<th>sim</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToM</td>
<td></td>
<td>14</td>
<td>8</td>
<td>14</td>
<td>18</td>
<td>19</td>
<td>13</td>
<td>18</td>
<td>10</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>LoM</td>
<td></td>
<td>15</td>
<td>12</td>
<td>14</td>
<td>21</td>
<td>20</td>
<td>14</td>
<td>25</td>
<td>10</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>MCP</td>
<td></td>
<td>10</td>
<td>5</td>
<td>8</td>
<td>12</td>
<td>15</td>
<td>8</td>
<td>17</td>
<td>6</td>
<td>6</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 2: Number of steps to complete the task in the environments (b)

<table>
<thead>
<tr>
<th>Environment (b)</th>
<th>sim</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToM</td>
<td></td>
<td>41</td>
<td>55</td>
<td>56</td>
<td>45</td>
<td>28</td>
<td>52</td>
<td>57</td>
<td>57</td>
<td>52</td>
<td>42</td>
</tr>
<tr>
<td>LoM</td>
<td></td>
<td>21</td>
<td>32</td>
<td>27</td>
<td>23</td>
<td>24</td>
<td>35</td>
<td>31</td>
<td>37</td>
<td>38</td>
<td>43</td>
</tr>
<tr>
<td>MCP</td>
<td></td>
<td>10</td>
<td>26</td>
<td>27</td>
<td>15</td>
<td>16</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>38</td>
<td>43</td>
</tr>
</tbody>
</table>
In the environment (c) using the MCP, the agents executed a number of steps 67% lower than the number of steps executed by other agents in the ToM and 38% lower than the number of steps executed in the LoM. Table 3 describes the results. Again the differences are statistically significant.

Table 3: Number of steps to complete the task in the environments (c)

<table>
<thead>
<tr>
<th>Environment (c)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>LoM</td>
<td>55</td>
<td>149</td>
<td>159</td>
<td>80</td>
<td>100</td>
<td>141</td>
<td>173</td>
<td>140</td>
<td>113</td>
<td>126</td>
</tr>
<tr>
<td>MCP</td>
<td>24</td>
<td>98</td>
<td>123</td>
<td>84</td>
<td>154</td>
<td>152</td>
<td>158</td>
<td>136</td>
<td>32</td>
<td>22</td>
</tr>
</tbody>
</table>

7 Conclusion

In this work, we compared the proposed model (Model of Combined Planning) that uses a combination of planning for localization and tasks with two other models. The first comparative model prioritizes the task (which is to reach a particular point on the map) and another model prioritizes the localization process, initiating the final task only after finding its location.

Imagining scenarios where the robot does not have a good estimate of its pose, when we use the model that prioritizes the task (goal), the robot believes to be in the cell that has the highest probability, moving to where the robot believes it is the goal. But if the estimated location was not good, its pose was not always correct. Thus, the chances of the robot to choose an incorrect action is high.

Using the model that prioritizes the localization process, we noticed that many times, localization actions were contradictory to goal actions. This meant that, often, the robot needed to head to a region of the environment to locate yourself, but this region was opposite to the goal region.

In all environments, the robot that used the proposed model (Model of Combined Planning) estimated its pose only after the robot that used the localization-oriented model. However, although located steps later, it was closer to the goal, completing the task in less steps. For cases where only the location is necessary, the best method was the LoM.

An open question is about modeling with pure POMDP. The algorithms tested were unable to produce policies for the environments with an amount of states larger than 1400. Aspects of the modeling, such as sensory abilities and the movements of the robots, interfere in the policy generation performance. However, the amount of states is a factor more relevant, since it is associated with functions of transition and individual observations of each state. A proposal to solve this problem would be modeled the problem with factored POMDPs and testing in larger environments.

Acknowledgments

The authors wish to acknowledge the financial support of CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), grants 141061/2010-3.

References


