

Planning Under Uncertainty for Mapping

Team 7

I. THE ACTIVE SLAM PROBLEM

A central problem in deliberative robotics is mapping an unknown world. Search and rescue robots enter an unknown disaster area and must generate a map as they explore. Without an accurate map of the world, a robot cannot effectively reason about future actions. Simultaneous Localization and Mapping (SLAM) is a successful technique for mapping an unknown environment with relatively dense, unique features. Usually, acquiring sensor data over the whole map requires human intervention to cover unexplored area. This process does not scale as intelligent robots become ubiquitous. Beyond the fact that offline algorithms cannot determine a strategy to completely cover the map, the computed path is clearly not optimal. Recently, some online algorithms attempted to actively choose the next observation point depending on the data already recorded to optimize coverage and accuracy.

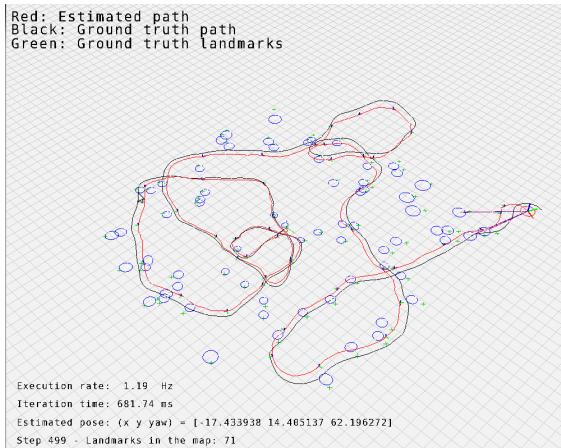


Figure 1. Example robot exploration using SPLAM

Under real-world time constraints, however, a robot must choose between coverage and accuracy of an unknown area in order to maximize information gain while dealing with imperfect and incomplete information. To be effective in online mapping, a robot must be able to evaluate the possible information gained by closing-the-loop or moving to a new observation point, as shown on Fig. 1. We will address the time-constrained issue, the choice of an uncertainty measure, and the exploration-exploitation trade-off in Active SLAM.

II. RELATED WORK

Active SLAM is a particular issue of Active Sensing [6], which address the problem of how to optimally control a sensor for information gathering. Active SLAM computes an exploration policy – a strategy – for a robot equipped with a SLAM algorithm in an

unknown environment. The aim is to obtain a policy which trades off exploration (maximizing map coverage) and exploitation (going back to known locations to increase the accuracy of the map). The first approaches developed to solve this hard task were greedy algorithms [2] that use the current knowledge to maximize the information gained in the next measurement. It should be noted that planning with a longer horizon could improve results.

The most general way to describe the policy is a sequence of freely chosen actions. The problem of finding a sequence of actions to minimize cost for systems that fully observe the system's state is called a Markov Decision Process (MDP). A system where the measurement is noisy or does not fully observe the system state is a Partially Observable Markov Decision Process (POMDP). Clearly, Active SLAM is a continuous-state and continuous-action POMDP. In that formulation, solving a POMDP is intractable computationally. Nevertheless, authors succeed to solve it with approximations or discretization. In [3], authors use Bayesian optimization of a Gaussian process. In [4], model-free reinforcement learning is applied. While these approaches seem to produce high quality paths, their computation is expensive.

III. APPROACH

A. Implementation Overview

Instead of trying to solve the POMDP, we implemented our solution using Model Predictive Control (MPC). In [8, 7, 5], authors use Model Predictive Control optimization with multiple step look-ahead in order to decide the next action to maximize the information gain. To plan the next actions, the robot simulates possible moves and predicts sensing. To simulate its moves, the robot chooses between a set of discretized actions. To predict the sensing, the robot uses its estimated map of the known features and a noisy sensor model. Thus, the robot builds a tree of possible paths from its current position and predicts the sensing along these trajectories. These observations are used to update the state of a Kalman Filter (KF). The first action that produced the trajectory which leads to the maximum information gain is then selected to be executed. After that move, the state of the KF is updated according the current sensing and the MPC search is run again to find the next action.

The trade-off between exploration (visiting unexplored area) and exploitation (reducing map uncertainty) is made by switching the current goal of a state machine guiding the move generation of the robot. A goal is represented by a virtual feature with a big uncertainty. That causes the robot to visit that feature. During exploration, the current goal is set to a point in an unexplored area of the map. In exploitation, the current goal is set to an already visited location to enhance the quality (reduce uncertainty) of a landmark or the robot state. Below we describe the implementation of our main contributions.

B. Framework

We used the Mobile Robot Programming Toolkit (MRPT) as our simulation framework [1]. It is an open-source toolkit designed to provide simulation tools for rapid robot algorithm development in C++. Modules include mapping, sensing, robot dynamics modeling, and a SLAM framework. Also included is a 3D visualization tool to display the robot in an environment based on OpenGL. Using MRPT, we implemented a basic robot model that is controlled with rotational and linear velocities. The simulation uses a PID controller to reach the commanded velocities. Each millisecond the pose of the robot is updated. We also inject Gaussian noise into odometry readings to simulate imprecise ego-motion sensing. The robot sensor provides distance and bearing measurements for all landmarks in its field of view with injected Gaussian noise to simulate imprecise sensor readings. Finally, SLAM is performed by an Extended Kalman Filter; we used the implementation of the MRPT::SLAM package. Also, we implemented a time-step simulation to update the robot position within the world for discrete time steps during simulation. Moving commands and sensing commands are sent every time-step.

C. Switching Policy

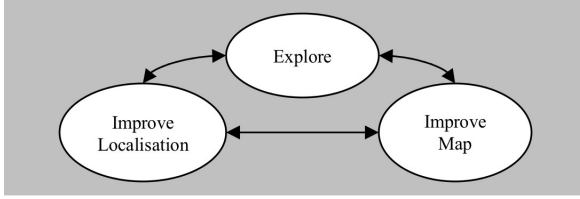


Figure 2. Exploration States

We implemented a 3-state state machine to control the movement policy of the robot, shown in figure 2. The state transitions are defined in terms of the robot state uncertainty, the landmark uncertainties, and the history of information gained from previous actions. The default goal is to *Explore* which will choose the predicted best point outside the known area to explore. The choice of point is described below. If the robot state uncertainty increases beyond a threshold, the state changes to *Improve Localisation* and the robot returns to a well-known point in its map to reduce state uncertainty. If the overall map uncertainty is above a threshold, the state changes to *Improve Map* and the robot revisits known landmarks with high uncertainty. Our thresholds for switching states is predetermined by the user and reflects the desire to focus on exploration or exploitation. Thus, we will avoid infinite loops and map inconsistency.

D. Uncertainty Measure

The uncertainty measure is related to the SLAM algorithm used. The classical approach, which we use, relies on Extended Kalman Filters (EKF) [5]. The uncertainty is contained in the covariance matrix of the EKF. The global goal is to minimize the covariance matrix. Possible norms include the trace, the determinant, and the maximum of Eigen values. The trace has been shown to most effectively measure the uncertainty for SLAM [5, 7]. So, we focus on the trace.

E. Map Expansion

We implemented the frontier-based goal selection [8] using the nearest attractive point [5] to determine the expected best point to move toward during the exploration state. This works by first evaluating the predicted information gain for regions just beyond the boundary of the known map. Using the maximum predicted gain area, we place an artificial attractor in the unknown region to bias the robot's movement toward this goal. These attractors only effect the robot movement in the explore state.

IV. EVALUATION

We compare multiple MPC algorithms running our SPLAM implementation on two maps using the MRPT simulator. The small simulated map contains 30 landmarks randomly distributed throughout $400 m^2$ and the second contains 70 landmarks randomly distributed throughout $900 m^2$. The robot is allowed to wander away from the bounding box around the landmarks, but the attractive forces of visible landmarks always pull the robot back into the landmark region. All the simulation parameters are detailed below. We tested the SPLAM using greedy, 2-step MPC, and 3-step MPC policies.

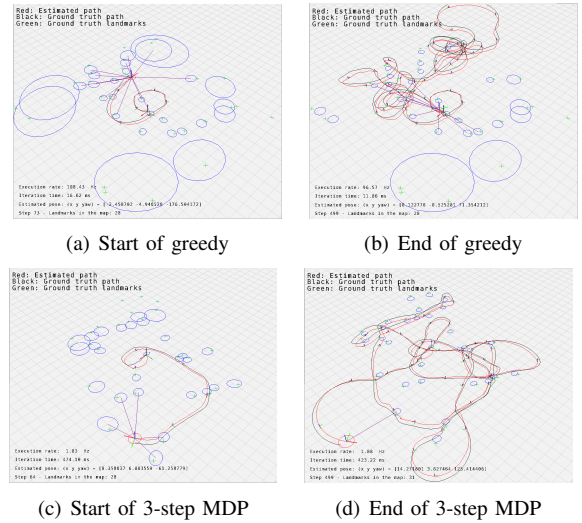
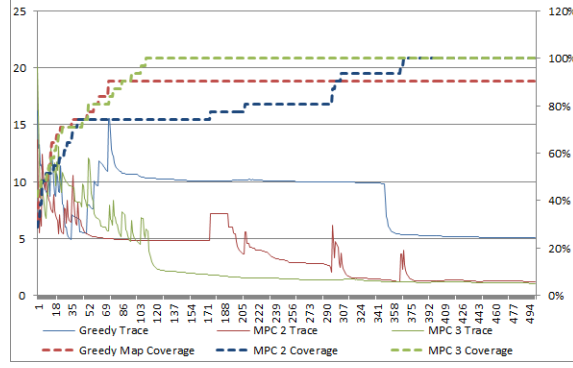
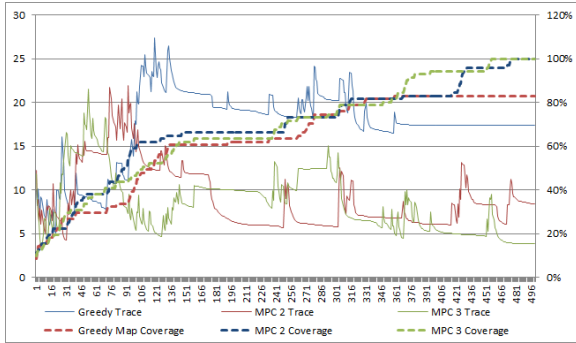


Figure 3. Results of SPLAM on small map

The performance metrics are the map coverage after a maximum of 500 time steps if 100% coverage is not reached prior, and the remaining uncertainty in the map as defined by the covariance matrix of the EKF. In our simulations, one time step is 500 ms. Visualization results are shown in Figure 3 for the small map and Figure 1 for the large map. The estimated path is lighter, while the actual path is darker. Each landmark is marked with a + sign and the diameter of the ring around it indicates the uncertainty of that landmark. Lines emanating from the robot frame indicate landmarks that are within range of the robot's sensors. The graphs in Figure 4 show the map uncertainty along with the coverage percent over the entire 500-step simulation on our two test maps for each implemented algorithm. The evaluation method presented is the trace, which is a common norm choice used to evaluate the uncertainty.



(a) Small Map



(b) Large Map

Figure 4. Map trace (solid lines) and map coverage (dashed) over the course of the simulation

Dynamic constraints	
Number of discretized actions	9
Max speed	2 m/s
Max turn rate	1 rad/s
Min speed	0 m/s
Min turn rate	-1 rad/s
Odometry Error per Step	
Noise std (XY)	0.05 m
Noise std (ϕ)	0.05 degrees
Range and bearing sensor	
Min sensor distance	0.5 m
Max sensor distance	7 m
Field of view	360 degrees
Noise std (range)	1 m
Noise std (Yaw)	10 degrees
Robot state thresholds	
Max trace for Exploration	0.75 m ²
Max trace for Improve Map	0.75 m ²

Table I
SIMULATION PARAMETERS

V. DISCUSSION

We proposed a new way to solve the Active SLAM problem using a Model Predictive Controller to approximate the intractable exact solution of the associated POMDP. In comparison to the other previous approaches that used a Greedy algorithm, our solution is able to efficiently plan with a larger horizon with respect to computation.

Moreover, the state machine is a simple and effective way to deal with the trade-off between exploration and exploitation. On the Figure 3.b, the greedy algorithm is not able to efficiently cover the entire map because it frequently loops on itself due to the lack of planning horizon. In comparison, the 3-step MDP on Figure 3.d, successfully fulfills the accuracy and coverage requirements. These results are also shown in Figure 4.a. In contrast with the small map, performance differences were much more pronounced on the big map. Indeed, on a big map the greedy algorithm may never explore distant regions because of its inability to plan some steps ahead. In general, the greedy algorithm has a tendency to make the robot loop in local clusters of landmarks. On Figure 4, the MPC with a 3-step look ahead outperforms the other methods for all metrics; it reaches 100% coverage faster and the trace converges to its minimum faster. This shows that our method outperforms previous methods.

For future work, we would like to improve (1) how to predict next steps, (2) which sequences of actions are evaluated and (3) how to influence the robot to explore new areas. First, instead of using the mean positions of estimated landmarks to predicted observations, we could sample landmark position and observation taking into account noises and uncertainty. Secondly, we could replace the breath-first search in the tree of trajectories with an RRT-search to reduce the computation of the complexity. Moreover RRT will allow us to select from a continuous set of actions. Another possibility is to optimize a parametrized trajectories to give the maximum information gain. We could use simulated annealing. Finally, we would like to replace the state machine by adding a gain relative to the chance to discover new feature in unvisited places. The chance to discover new features could be modeled as an uninformative uniform distribution. So the chance to discover new features is proportional to the unvisited area covered by a new observation and to the estimate density of the landmarks. The density of landmarks can be estimated from already explored area and already discovered area, in a Bayesian scheme.

REFERENCES

- [1] J.L. Blanco. *Derivation and implementation of a full 6D EKF-based solution to bearing-range SLAM*. Tech. rep. University of Malaga, 2008.
- [2] H J S Feder, J J Leonard, and C M Smith. "Adaptive Mobile Robot Navigation and Mapping". In: *International Journal Of Robotics Research* 18 (1999), pp. 650–668.
- [3] Thomas Kollar and Nicholas Roy. "Trajectory Optimization using Reinforcement Learning for Map Exploration". In: *Int. J. Rob. Res.* 27 (2 Feb. 2008), pp. 175–196. ISSN: 0278-3649. DOI: 10.1177/0278364907087426.
- [4] Thomas Kollar and Nicholas Roy. "Trajectory Optimization using Reinforcement Learning for Map Exploration". In: *Int. J. Rob. Res.* 27 (2 Feb. 2008), pp. 175–196. ISSN: 0278-3649. DOI: 10.1177/0278364907087426.
- [5] Cindy Leung et al. "Planning under uncertainty using model predictive control for information gathering". In: *Robotics and Autonomous Systems* 54.11 (2006). Planning Under Uncertainty in Robotics, pp. 898 –910. ISSN: 0921-8890. DOI: 10.1016/j.robot.2006.05.008.
- [6] L. Mihaylova et al. "Active Sensing for Robotics - A Survey". In: *in Proc. 5 th Intl Conf. On Numerical Methods and Applications*. 2002, pp. 316–324.
- [7] Robert Sim and Nicholas Roy. "Global A-Optimal Robot Exploration in SLAM." In: *ICRA'05*. 2005, pp. 661–666.
- [8] C. Stachniss, G. Grisetti, and W. Burgard. "Information Gain-based Exploration Using Rao-Blackwellized Particle Filters". In: *Proc. of Robotics: Science and Systems (RSS)*. Cambridge, MA, USA, 2005, pp. 65–72.