Emilie Naples, Department of Mathematics
University of California, Riverside
NASA MIRO FIELDS Program
JPL Robotics Research Intern, Summer 2017
Motion Planning Under Uncertainty for Planetary Navigation

ABSRACT
This paper will discuss the concept of motion planning under uncertainty for planetary navigation as it pertains to a machine's goal to be able to operate autonomously from a start state to a goal state using path planning methods. The greatest challenge of enabling such autonomous operation in a system is incorporating the uncertainty in location and environmental conditions that a robot experiences. When motion planning, a robot must be able to go from a start state to a goal state while taking into account such uncertainties in the environment, avoid any obstacles, and calculate risk to determine the best path for doing so. We approach the problem of enabling autonomous operation in a system as an optimization problem that minimizes time and risk (cost to goal). Minimizing such things allow for accurate and efficient motion and path planning by decreasing uncertainty in the environment. Specifically, we address the problem of autonomous robotic planning under motion and sensing uncertainties and methods by which to compute the shortest, most cost-effective path between start state to goal state.

INTRODUCTION

The main goal when planning under uncertainty for planetary navigation is to minimize cost when going from one goal to another. To be able to do so, an agent must be able to operate under uncertain conditions by finding the most effective path and following it. The most effective path is defined as the path which allows the agent to go from one state to anther while minimizing uncertainty and risk to the autonomous machine. This is done by an optimization problem where optimization is performed over a series of different potential paths.

Existing path planning methods for motion planning under uncertainty provide several results. These paths take the certainty of motion into account. On these paths, the certainty of the robot's location is considered and propagated upon the path options in an open-loop fashion. In other words, data and information from sources around the area are not incorporated into the algorithm for path planning. More recent methods do incorporate such information gathered by sources around the terrain, allowing for more accurate planning. In these newer, more practical methods, updates about the robot's surroundings are performed periodically. They also take motion and sensing uncertainty into account, thus allowing for more conservative path planning.

a) Motivation in Robotics

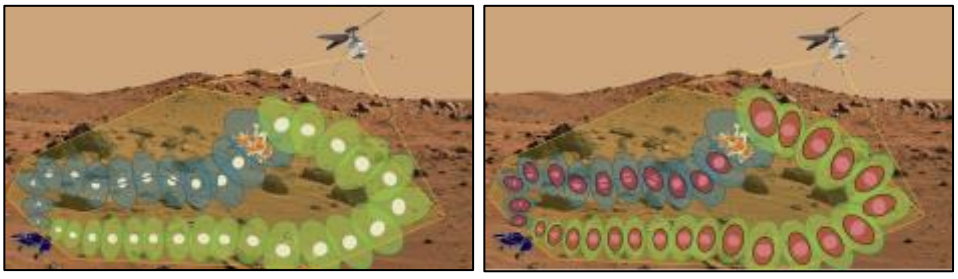


Figure 1

In the figure above, see two path options that the robot can follow, created by a path planning algorithm. Ellipses represent covariances at each node along the path, giving an overview of the uncertainty of the robot's state at that location (node). The path outlined by blue ellipses is that which has the most uncertainty, and hence has the largest ellipses. The path outlined by the green ellipses is more certain, and hence, has smaller covariances along each node because it is formulated by taking information from the surrounding environment into account. When planning under uncertainty, older methods later discussed would take into account the uncertainty of motion, defined mathematically by x_{k+1}. $f(x_k, u_k, w_k)$ accounting for state, control, and motion noise in the surrounding environment, respectively. x_{k+1} represents the state reached after the previous state x_k, u_k represents control, and w_k represents motion noise. We model this using a Markov decision process, or MDP explained in the next subsection. Newer path planning methods are more efficient in the way that they take into account both motion and sensing uncertainty. They are also defined by $x_{k+1} = f(x_k, u_k, w_k)$ accounting for a robot's state, control, and motion noise. In addition, since these newer methods also take sensing uncertainty into account, they incorporate sensor noise represented by $z_k = h(x_k, v_k)$ where v_k represents sensor noise and x_k still represents the previous state of the robot.

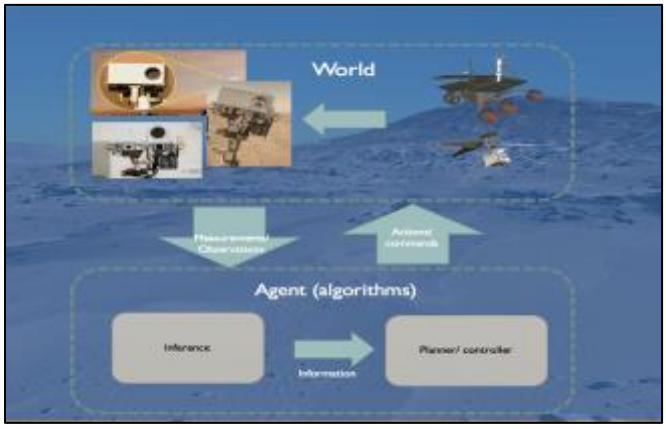


Figure 2: A system is represented by a state, control, and motion noise. Sensors detect information gathered by the system. It is measured and sent to the state estimator where the most efficient action for the robot to take is determined.

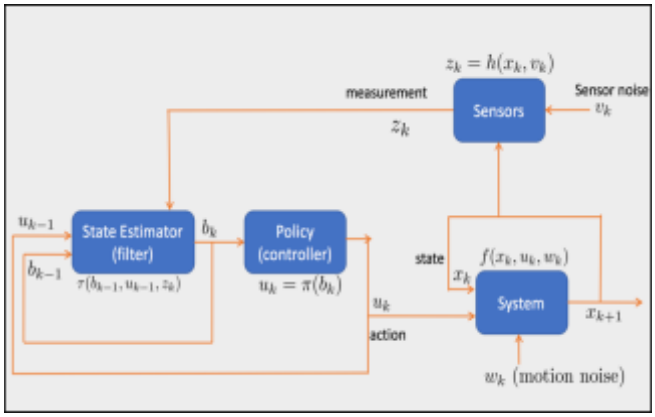


Figure 3: A more detailed and internal view of Figure 2. The process in which an agent determines the best action based on outside information from the environment.

b) Markov Decision Processes

A Markov decision process, or MDP, is a decision-making process used to mathematically model situations where an outcome is partly random and partly due to a set of actions chosen by a decision maker [5]. In an MDP, the state of the agent is known, so uncertainty is not highly accounted for. At each time step in a modeled MDP, the decision-making process allows the decision maker to choose any action available in a present state, *s*, of the agent to then move to another state, *s'*, thus continuing along a path and moving closer to a set goal state. A reward is then given to the decision maker after it brings the process into the new state. MDPs model situations where the state of the agent is known. These are situations in which only the location of the agent has been taken into account, and not that of its surroundings.

However, in cases where the state of the agent is not known, i.e. in the presence of motion *and* sensing uncertainty, a partially observable Markov decision process, or POMDP, is used instead of an MDP to model the decision-making process. Like an MDP, a POMDP allows the agent to complete a set of actions in order to go from a start state to a goal state without collision. But, in situations modeled by POMDPs, a probability distribution is created over all possible states of the agent since its exact location is unknown. In other words, the difference between a POMDP and an MDP is that a POMDP models situations in which we account for the location of the agent and its surroundings.

Specifically, a POMDP is a 6-tuple (S, A, O, T, Z, R) with the following components:

S: state space

A: action space

O: observation space

T: Transition function

$$T(s,a,s') = P(S_{t+1} = s') \mid S_t = s, A_t = a)$$

Observation function:

$$Z(s,a,o) = P(O_{t+1} = o \mid S_{t+1} = s, A_t = a)$$

Reward function:

$$R(s,a)$$

At each time step, the environment is in some state. The agent takes an action, determined previously by a planning algorithm, which causes the environment to transition to the next state, *s'*, with probability $T(s,a,s')$. At the same time, the agent receives an observation *o* which depends on the new state. The new state has probability $O(o,s',a)$. Once the agent has reached the next state, *s'*, it receives the reward previously determined by the planner. The goal is for the agent to choose actions at each time step that maximizes the expected future discounted reward. This action selection strategy is called a policy. A policy, once determined from the information given to the state estimator, maps each state, *s*, to a desirable action, *a*. The policy determined by the estimator is the solution to the POMDP [4].

Figure 4: A partially observable Markov decision process representation with 3 states (s_0, s_1, s_2, s_3), and 3 corresponding observations (o_0, o_1, o_2, o_3), actions (a_0, a_1, a_2, a_3), and rewards (R_0,R_1,R_2,R_3) respectively.

When path planning, random points sampled from the environment are put together to form a search space. Random points in a search space are sampled so that the robot can get a feel for the surrounding environment. The possibilities are grouped together to form a map, and the state estimator is responsible for determining the best path from all the possibilities. As previously mentioned, an MDP is used to model these situations in which the state of the agent is known. However, in a POMDP, where its location is not known as it is in an MDP, each state in the map is associated with a corresponding belief. The belief represents the probability that an agent could be at that state at a given time. A POMDP models situations where there is a considerable amount of motion and sensing uncertainty, i.e. we do not know the state or "location" of the agent. Because of this, a belief *b* is necessary to define. Each belief is said to be a mean *x* and a covariance *P*, each representing a unique "location". The word *location* is in quotes here because in belief space where POMDPs are modeled, the robot's location is not certain. It only makes sense to say that a robot has a certain location in *state space*. It is important to understand that when searching for a path from a start state to a goal state for an agent to follow, the sampled points form a map of states in the state space where motion uncertainty is taken into account. When both motion and sensing uncertainty are accounted for, a corresponding map of the state space is formed by the probability distribution over all states. The space in which this map is defined is called the belief space. A visual representation follows.

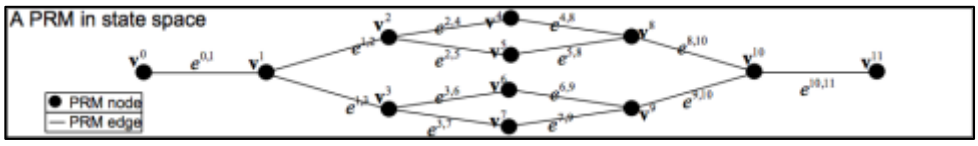


Figure 5: A probabilistic roadmap from sampled points in an environment.

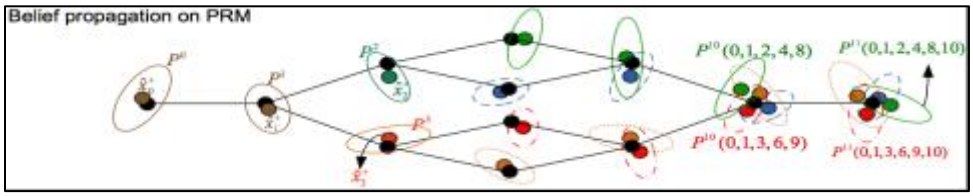


Figure 6: Belief nodes with mean and covariance corresponding to each node in the state space.

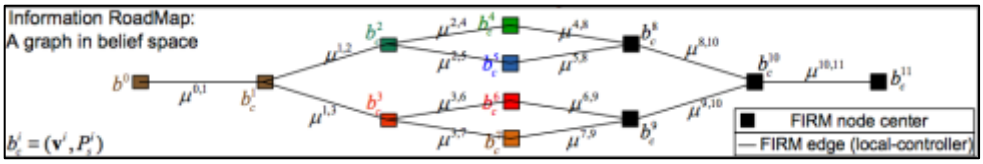


Figure 7: The resulting graph in belief space with belief nodes b_c, each defined by a mean v and a covariance P.

Some benefits of using POMDPs for planning applications are that they allow more realistic path planning situations to be modeled for study and direct action. This is because the POMDP model allows the sensing and motion uncertainty experienced in actual environmental settings to be more easily incorporated into problems. They enable the optimization of any user-defined function. While this is very beneficial, POMDPs are still difficult to solve for large problems because of implications known as the curse of dimensionality and the curse of history. The curse of dimensionality refers to the increasingly large search space that must be considered when searching for a path from a start state to a goal state. For each state, there is a new series of paths that can be followed from it. The number of outcomes for each node, as a result, grows exponentially as search space is enlarged due to time. The curse of history is another result of the growing number of possible paths from each node that robot can take. As seen in Figure 8, the covariance at one node is defined by all of the previous nodes encountered to get there. In other words, previous paths must be incorporated. As mentioned, this a problem for larger search spaces. A solution to this problem is described in the *Results* section.

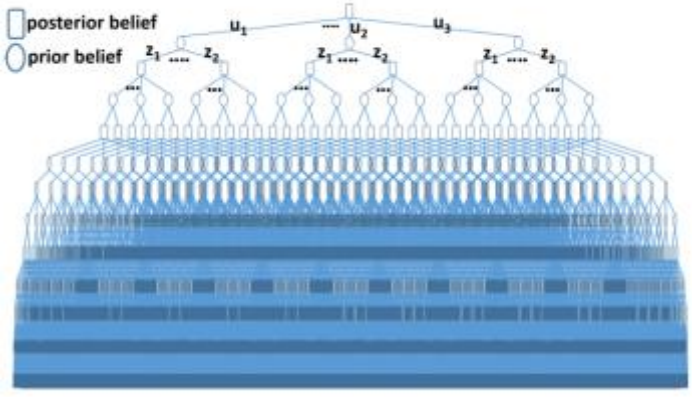


Figure 8: Curse of history. Exponential growth of outcomes for a single belief.

We now discuss at length some types of state-of-the-art POMDP solvers which are effective at computing a cost efficient path from a start state to a goal state under conditions of uncertainty.

METHODS

a) Point-based POMDP solvers

Point based POMDP solvers compute a path from a start state to a goal state by computing a *value function* over a finite subset of the belief space. This action approximates the belief space and therefore makes it easier to develop a policy. For any point (belief) in the set, the optimal action to reach the goal is determined by software that is responsible for creating an action observation sequence. This tree of different paths provides the cost to goal from a belief in the belief space. Each node can be described as a tuple containing all nodes used to get to that location on the map. Point-based algorithms explore the belief space, focusing on the reachable belief states while maintaining a value function. In other words, a variant of value iteration is performed where backup is done only at selected beliefs. This avoids exponential growth of the value function caused by the growing dimension which is proportional in growth to the number of states [4]. For better understanding and explanation, the following information on policies and value functions from MDPs and POMDPs is taken from the *Policies and value functions* section from reference source [4] in the *Reference* section of this paper.

"The main goal of MDP planning (and POMDP planning in the case of both motion and sensing uncertainty) is to discover an action-selection strategy, called a policy, defining how the agent should behave (i.e. what actions it should take) to maximize the rewards it obtains. … In this discounted case, we assume that earlier rewards are preferred, and use a predefined discount factor, $y \in [0,1)$. This reduces the importance of later rewards.

A solution to an MDP, and that of a POMDP as well, is a policy, π, that maps each state to a desirable action with expected stream of discounted rewards. The agent seeks a policy that optimizes the expected infinite stream of discounted rewards.

Point-based POMDP solvers are generally effective for maps that contain a large number of nodes in the belief space, avoiding the problem of dimensionality that occurs when the belief space grows exponentially with the number of states. Point-based POMDP solvers take a sample of points around a belief node in the belief space using an MDP. The sample is within only a certain reachable space of each belief node. From the sample around each node, an approximation is created rather than a sample of the whole space. An approximately optimal value function is computed from the samples and is used to find the best path from node to node. However, MDP-guided sampling is usually only effective when uncertainty is not high and there is not a long series of information gathering. When there is *uncertainty and a long history of information gathering, the curse of history problem arises. Each edge, or "piece" of the path, is dependent on the last and how that state was reached,* (as mentioned in the previous subsection of this paper).

For each node in the belief space, a map of paths is created from the start node to all others in the belief space. For this reason, point-based POMDP solvers require that each visited node be remembered by the planner in order to compute the entire path from the start state to the goal state. For each start node, a path is created by visiting other nodes in the belief space until eventually reaching the goal. Each path, J, consists of the nodes "visited" between the start and goal states and is thus defined mathematically by adding each of the belief nodes together as

$$J = ca_1 + ca_2 + ca_3 + ...$$

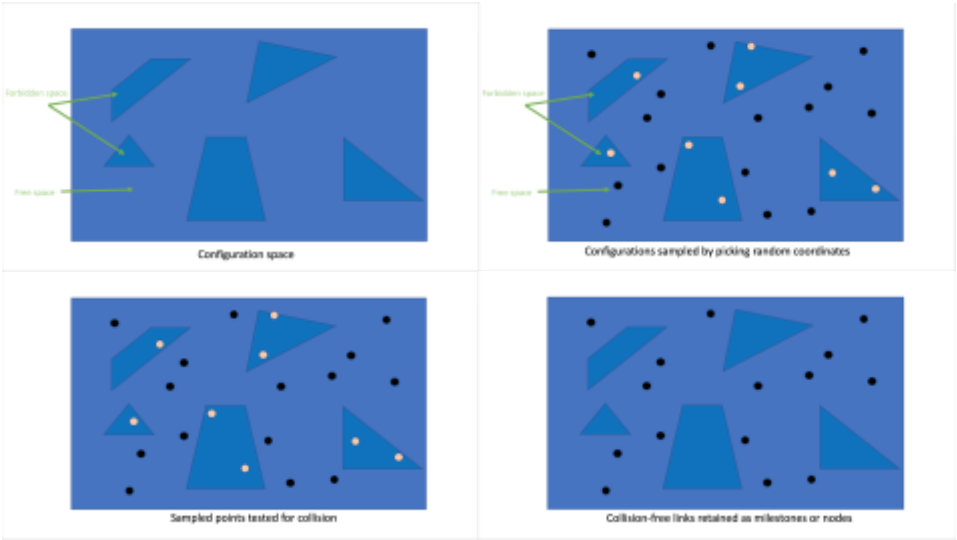
where a represents each action and c represents the corresponding cost of each action. In short, the problem can be fixed by scaling the cost of each action as

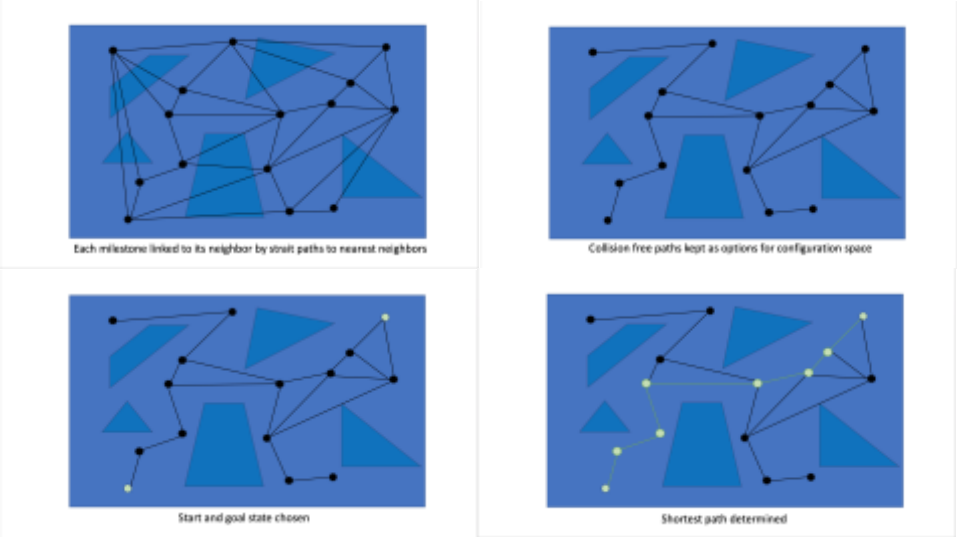
$$J = ca_1 + c^2a_2 + c^3a_3 + ...$$

To conclude, value function computations in point-based solvers are restricted to a finite subset of the belief space, allowing only location value updates for a subset [of the belief space]" [4]. This is known as point-based value iteration. It is what is used to avoid the exponential growth of the value function mentioned above and is thus applicable for domains with relatively large state spaces.

b) Sampling-based POMDP solvers

In short, sampling-based POMDP solvers sample nodes to form a state space. Those which hit obstacles are rejected and the ones remaining are connected, forming a belief node roadmap of possible paths from the start state to the goal state. Sampled values at each belief node on the roadmap are then determined by software algorithms, and eventually, the whole space is "covered" by a probabilistic roadmap, thus revealing all possible states of the robot and its surroundings at each.





c) Gaussian POMDPs

A POMDP models path planning situations that account for both motion and sensing uncertainty. Both types of uncertainty are incorporated because of the lack of knowledge of the location of the agent. Thus, a probability distribution over all states must be kept in order to plan in this way. In the POMDP path planning model, states in space are represented as beliefs where a probability distribution over all states is created. We will take for granted without explanation here that the distribution over the state space is assumed to be Gaussian. Therefore, the probability that the robot is at a certain state can be represented by a bell-shaped curve for each. It is known that the area in two dimensions under the Gaussian curve if bisected by a plane results in some area represented by an ellipse. Thus, at each node we have a corresponding distribution represented by the ellipse around it. The ellipses at each node indicate how likely it is that the robot is in that state. These ellipses are what defines each belief state in this way, specifically by a mean and covariance.

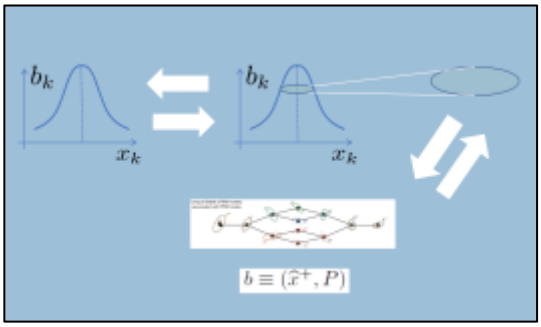


Figure 10: Each belief state *b* is defined by a mean and covariance.

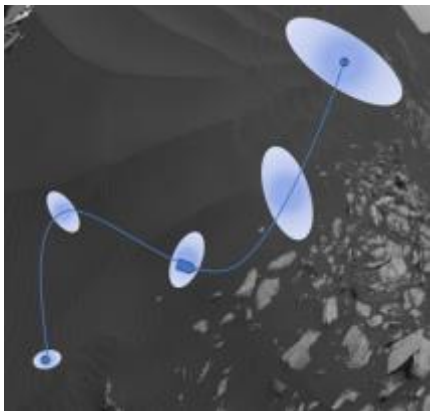


Figure 11: Ellipses at each node along a path in belief space. Due to increasing sensing and motion uncertainty, ellipses get larger as the agent moves farther from its starting point.

In other words, the ellipses at each node represent the level of uncertainty at each state. Note: Situations modeled by MDPs are those which account for the uncertainty of a robot's location. However, MDPs do not account for the uncertainty in the environment. Because of this, we still encounter some uncertainty in terms of surroundings. Because of this, the ellipses that are a result of Gaussian distributions modeled by MDPs are larger than those modeled by POMDPs. Larger ellipses correspond to more uncertainty, or equivalently, less information. So, the ellipses represented by POMDP Gaussian distributions on the nodes of the belief space tend to be smaller since their models account for motion and sensing uncertainties, thus enhancing their capabilities with more information. Ellipses represented by POMDPs are generally smaller than those modeled by MDPs for this reason. Belief states where there is more certainty are those in which there has been more information gathered from an agent's sensors. Figure 12 shows a path consisting of belief nodes formed by a planner. The nodes on the entire right side of the path have smaller ellipses than those on the left side of the path. This is because the robot's sensors were able to detect more information about the terrain on the right than the terrain on the left.

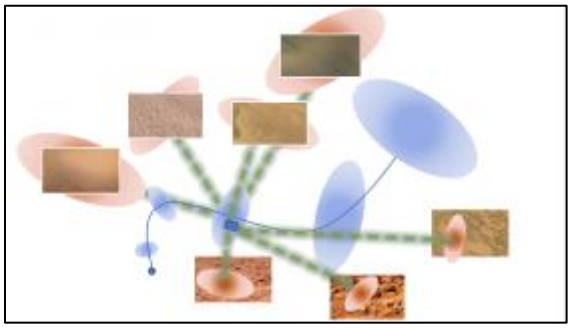


Figure 12: Ellipses are smaller on the right side of the path at where the terrain contains more features, more information. Ellipses are larger where there are less features, corresponding to less certainty.

d) FIRM and Rollout

FIRM (feedback information roadmap) is an enhancement to state-of-the-art path planning methods and POMDP solvers mentioned before because it is a method which eliminates the complications posed by the curse of history. It has been stressed throughout the

introduction of this paper that path planning with motion and sensing uncertainties taken into account is vital to realistic planning. FIRM is the method that meets all the requirements stated in the POMDP introduction section that are necessary for doing so. It is the first path planning method that takes into account these two types of uncertainties while also accounting for noise, previous action, and current state of the agent. In FIRM, path edges are independent of each other, so it is no longer necessary to remember each previously visited node in a sequence to develop a path as a whole. All of a sudden, large POMDP problems are now solvable since there are no longer an exponential number of outcomes stemming from a single belief. FIRM is the belief space variant of other probabilistic roadmaps which puts the belief space components mentioned in *Gaussian POMDPs* into practice.

FIRM is able to accomplish this by updating the roadmap at each belief state. This is what eliminates edge dependence and the need for extensive re-planning. FIRM does its planning offline. In the planning phase, the belief space is searched to determine which path is the least "expensive", (i.e. minimizing cost-to-go). Then in the execution phase, the robot gets the results and executes the best policy found, thus forming a path [2].

The rollout policy in belief space reduces the computational cost a step further because it does not approximate the system with a deterministic one, and it avoids local minima in the optimization problem used in computation by suboptimal policy that approximates the true cost-to-go beyond a certain horizon. FIRM is an offline planner, whereas rollout is an online planner. This means that in the planning phase, the belief space is searched to determine which path gives the least cost to go, just as in FIRM. This is the path chosen to follow. Then, rollout re-plans iteratively along each edge and node, updating the belief state and path plan more frequently. This ensures the maximum amount of certainty possible to obtain the best policy [2].

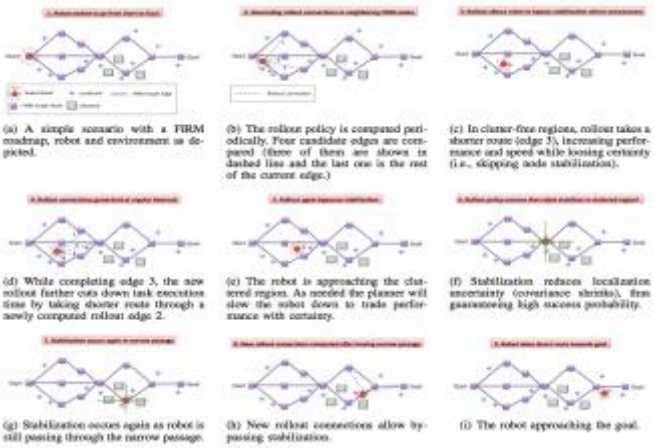


Figure: 13 Rollout-based execution of the FIRM policy.

RESULTS

a) Overview

We have combined the capabilities of previous state-of-the-art path planning methods and models discussed in the above sections to incorporate the motion and sensing uncertainties in an environment. Being able to do so is something new that makes our developed feedback-based information roadmap (FIRM) method different from previous path planning methods like probabilistic roadmaps and rapidly expanding randomized tree searches, which only accounted

for the motion uncertainty in an environment. These previously developed methods for path planning sample points in state space, whereas the POMCP, FIRM, FIRM-rollout, and FIRMCP (FIRM with Monte Carlo planning) methods plan in a belief space while *also* avoiding the curse of history problem encountered when solving large POMDPs. In FIRM, a map is formed based on information about an environment's terrain and a probability distribution over all states in that map is created to incorporate the sensing uncertainty previously left out of older methods of path planning. This sensing (observation) in the path planning algorithm helps reduce the state uncertainty (risk) by filtering. Specifically, it reduces uncertainty by using Bayesian inference to update the overall probability distribution on each of the belief states at each node, thus providing more accurate information about the roadmap and a less costly potential path as time progresses. With good observation, a costly, suboptimal path can be an efficient near-optimal path [A].

Our rollout based execution of the FIRM policy evaluates each path's uncertainty (risk) and optimizes our developed motion plan over it. As a result, however, the search space gets much larger, and the current belief state computed from recursive Bayesian filtering depends on each previously visited state. This is known as the curse of history. We have solved this issue by incorporating feedback controllers, which merge belief states. In other words, we directly control the belief in order to break the curse of history presented by implementing the POMDP model in motion planning under motion *and* sensing uncertainty. If a belief can be steered directly, it can be predefined and break the dependence on the path as whole, thus breaking the curse of history.

To summarize more specifically, we have found a way to incorporate motion and sensing uncertainty into path planning while avoiding edge dependence (i.e. the dependence on each belief node previously visited by a path) and the curse of history. This is what is done by steering each belief directly. Steering each belief is partially achieved by classical controllers that act like funnels. As directly mentioned in [1], *"we construct a stationary linear quadratic Gaussian controller-based (SLQG-based) instantiation of this generic framework, called SLQG-FIRM, where we provide a specific node sampler and connector. In SLQG-FIRM we first focus on the kinematic systems and then extend it to dynamical systems by restricting sampling space to the equilibrium space. SLQG-FIRM is the first method that generalizes the PRM to the belief space such that the incurred costs on different edges of the roadmap are independent of each other, while providing a straightforward approach to sample reachable belief nodes. This property is a direct consequence of utilizing feedback controllers in the construction of FIRM. Based on this property, the FIRM framework breaks the curse of history in POMDPs (Pineau et al., 2003), and provides the optimal feedback policy over the roadmap instead of returning a single nominal path"* [1]. Side note: The programming code allowing this process to happen was also perfected to reach a much faster speed in simulation.

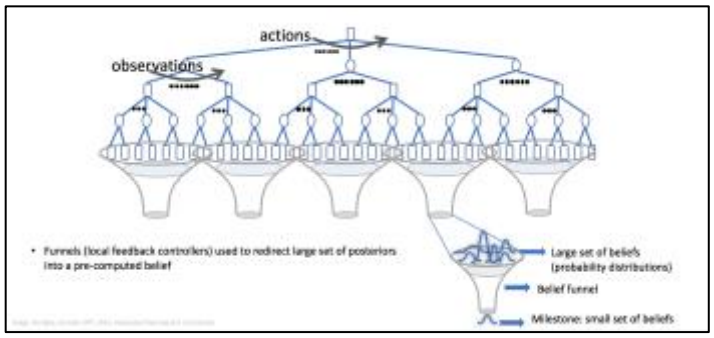
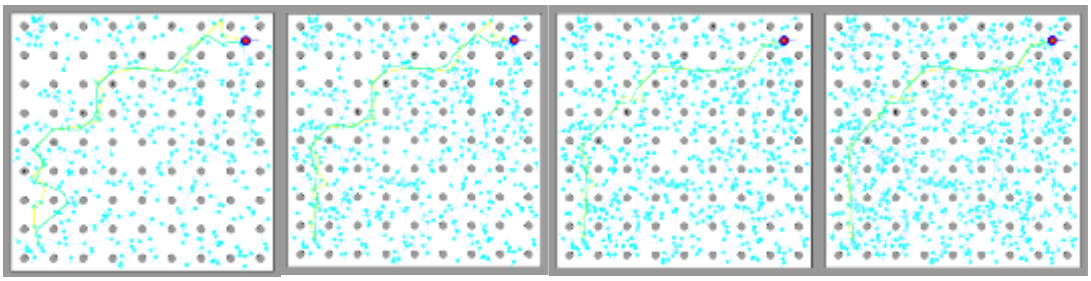


Figure 14: Funnels (local feedback controllers) used to redirect large set of posteriors into a pre-computed belief.

b) Simulation Results

During simulation, increasing the number of nodes sampled in the belief space provided different results for the two paths produced by the FIRM and rollout policies. FIRM is represented by the yellow path and rollout is represented by the green path.



Some variations on FIRM were also explored. The next figure compares the number of FIRM nodes with the number of time steps, the sum of traces of covariance, and the total cost of the FIRM, FIRM-Rollout, and FIRMCP methods.

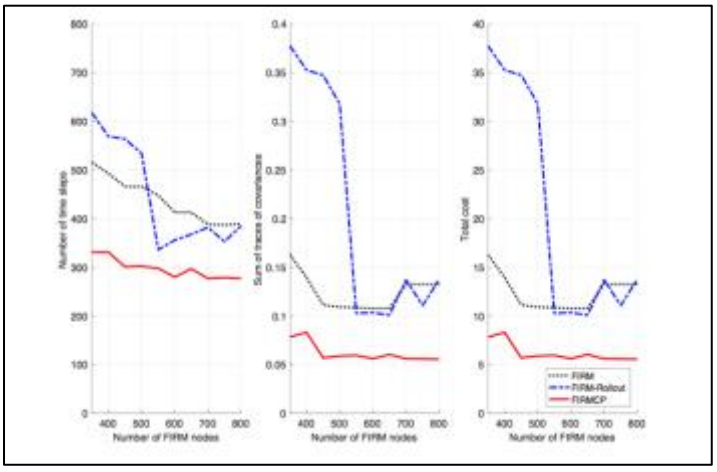


Figure 15

CONCLUSIONS

a) Significance of Results

Older path planning methods for motion planning under uncertainty incorporated the motion uncertainty (noise) present in an environment. The risk-minimizing optimization problem used for finding a path did *not* incorporate sensing into the algorithmic loop. As a result, it was much too conservative and very far from optimal. These full-width offline solvers enumerated every possible case, selected the best action, and found a solution in the reachable tree. We now incorporate the sensing uncertainty experienced in the environment into the loop, and while doing so, have also used an SQLG-FIRM to avoid issues such as the curse of history that we experience while implementing this. Previously written code allowing us to do this was significantly improved and made over 100 times faster by fixing several bugs over a 10-week period. By using perfected full-width online solvers and updating the policy found by the algorithm *during* execution at each belief state (instead of only at the beginning as in previous path planning algorithms), we increase accuracy and decrease the risk posed by high amounts of inevitable motion and sensing uncertainty in the environment.

b) Specific Motivation Applied to Future Space Missions:

Being able to incorporate both the motion and sensing uncertainties from the environment into POMDP solvers and path planning algorithms will significantly decrease the amount of uncertainty encountered when path planning for and during future missions. The FIRMCP algorithm incorporates POMCP's locally near-optimality and online computational power with FIRM's offline computational power and ability to solve large problems (Figure 14). This eliminates the most uncertainty from environmental data that has so far been seen in POMDP solvers and path planning algorithms. Such an advancement in state-of-the-art POMDP solvers will allow for maximum efficiency when exploring bodies in space and detailed operation during future space missions.

PROPOSED FUTURE WORK

The real world is full of uncertainty. Being able to model this uncertainty is very important for future missions and their success. Near-optimal long range solvers are in great need because they ensure that situations with this description can indeed be modeled. By continuing the editing done on previous code and by adding to the design on methods which go beyond the state-of-the-art such as FIRM, we can one day work our way up towards solving the ultimate minimization problem—that is the one presented by completely eliminating the uncertainties that come with path planning for space exploration.

Personal Commentary on Experience at JPL with the NASA MIRO FIELDS Program:

My personal experience at JPL with the NASA MIRO FIELDS program was nothing short of phenomenal. To call it anything less than a life-changer would be an understatement. First, I will share a brief overview of just a couple out of a thousand neat things I got to see at JPL, and then go into the great experience I had with the FIELDS program. As expected, I was introduced to many things that I had an idea existed, but definitely not to the level I was able to witness here at JPL. The ALPS lab tour and the Spaceflight Operations tour were among the some of the greatest sights. I got to witness the room under which all NASA spaceflight operations take place, and have been taking place nonstop since the very first launch of Explorer 1. Since then, the room has never been left unattended. As neat as this sounds, things became even more out-of-this-world when I got to see the Spaceflight Assembly Room where all

spacecraft missions to have left NASA's JPL facility have been launched into space—some returning and some still in route. To try and describe the way these rooms looked and the things inside them on paper would be impossible. In addition to seeing these famous landmarks, I also got to see many other labs and engineering bases where research vital to our understanding of the universe's contents is currently taking place. Things that people hear stories about came to life at JPL because I got to witness them first hand, and also see the amount of effort that goes into all of the little pieces. I directly observed research and ideas that are still in the works to improving space exploration even further, in addition to those which have already made it great. Being on lab was an eye-opener to something new every day.

The FIELDS program was just as great because after being at JPL for only a day, I realized that the program's main focus is a student learning experience. Nothing was stressed more during my internship than the importance of learning during this time. This not only took a lot of the pressure off of me as a student intern, but it also maximized what I am now able to say that I took away from JPL having been here through the FIELDS program. I would advise the staff in charge not to change a single thing. I think about how smoothly and perfect everything was set up and organized through UCR and the education office staff and student mentors at JPL, and it amazes me. The amount of communication, one-on-one student contact, and the learning focus here through the FIELDS program at a facility such as NASA known world-wide for its contributions to space is quite literally indescribable. Without the FIELDS program, it would have been a lot harder to think that I could intern at JPL. Now, because of this opportunity that was made possible for me, and with the knowledge and insight I have gained here, I feel I have been set me up for success no matter where I go. Before coming to JPL, I did not know which direction my life would go after college, but now I can say that I do.

In addition to an interesting and constant learning experience, JPL provided direction for my life in ways that words could never describe, and it was the FIELDS program that has made all of it possible by providing a path for me to JPL that was specifically made for students. This kind of support was exactly what I needed to figure out my next step towards a successful career. If I had to use an analogy to describe the experience at JPL made possible by the NASA MIRO FIELDS program, I would say it was like a picture worth a million words—except instead of looking at a picture, I got to live the picture. Thank you.

Acknowledgements

UCR NASA MIRO FIELDS program staff

JPL Education office staff

JPL Robotic Systems Section 347 staff including Supervisor Issa Nesnas for being a stellar group leader and Robotics Research Technologist, Dr. Aliakbar Aghamohammadi for being the most patient and dedicated to research I have ever seen a person be.

References

[1] H. Kurniawati, D. Hsu, and W. Lee. "SARSOP: Efficient point-based {POMDP} planning by approximating optimally reachable belief spaces," *Proceedings of Robotics Science and Systems*: Zurich, Switzerland, 2008.

[2] Ali-akbar Agha-mohammadi, Suman Chakravorty and Nancy M Amato. "FIRM: Sampling-based feedback motion planning under motion uncertainty and imperfect measurements," *The International Journal of Robotics Research* published online. Sage Publications: 15 November 2013. DOI: 10.1177/0278364913501564

[3] Ali-akbar Agha-mohammadi, Saurav Agarwal, Suman Chakravorty, and Nancy M. Amato. "Simultaneous Localization and Planning for Physical Mobile Robots via Enabling Dynamic Replanning in Belief Space," *CoRR:* 16 Sep 2016.

[4] Guy Shani, Joelle Pineau, Robert Kaplow. "A survey of point-based POMDP solvers,"*Auton Agent Multi-Agent Syst.* Springer Publishing: 8, June 2010. DOI 10.1007/s10458-012-9200-2

[5] http://wiki.ubc.ca/Course:CPSC522/Markov_Decision_Process