Motion Planning Under Uncertainty for Planetary Navigation

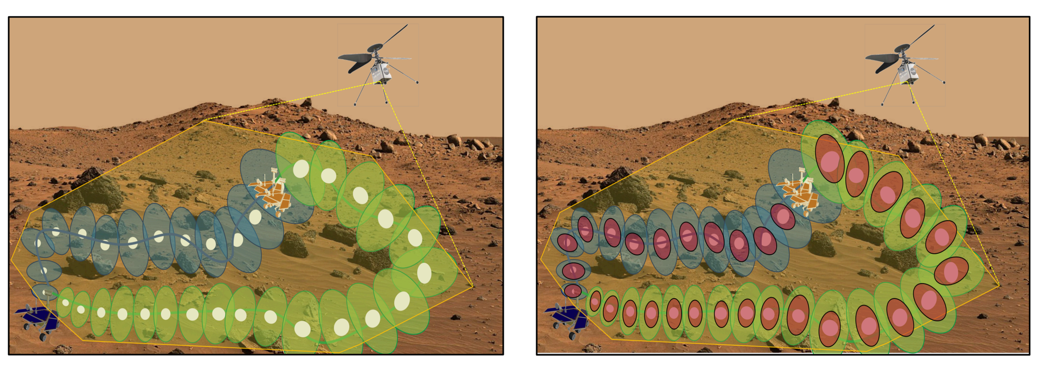
Emilie Naples
University of California, Riverside
JPL Maximizing Student Potential (MSP) in STEM Internship Program

Abstract:

The concept of motion planning under uncertainty for planetary navigation is very important as it pertains to a machine's goal to be able to operate autonomously from a start state to a goal state using internal operation and 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:

Use new variations on FIRM algorithm that allow for more exact motion planning by now taking into account both motion *and* sensing uncertainties in the environment. These methods take into account noise, previous action, and current state, and are the first to consider all three when motion planning



Paths defined by previous methods of motion planning vs. those as a result of new motion methods that incorporate both motion and sensing uncertainty. The larger ellipses at each node represent a larger amount of uncertainty.

Methods:

POMDP Definition

A 6-tuple (S, A, O, T, Z, R):

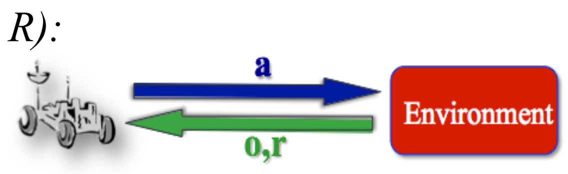
S: State space

A: Action space

O: Observation space

T: Transition function

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



Z: Observation function

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

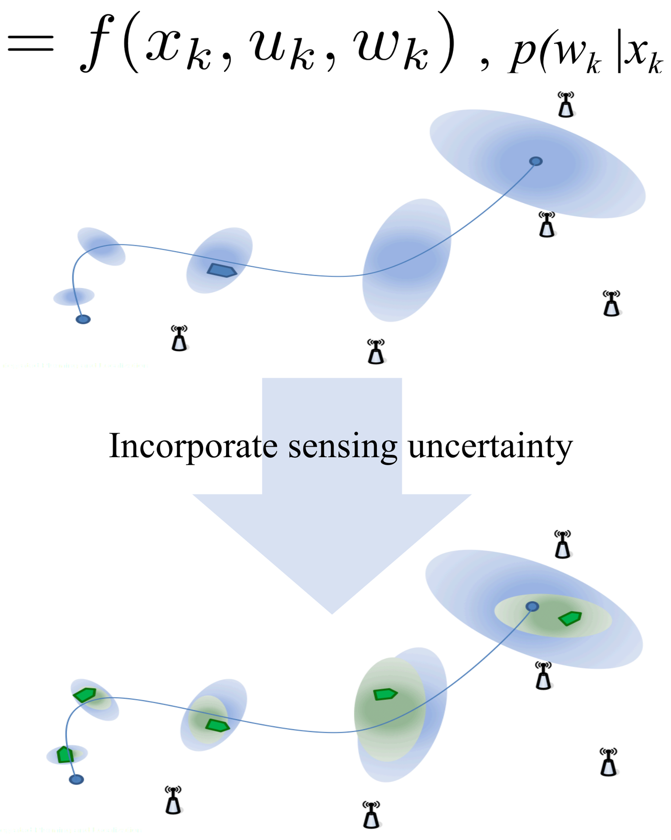
R: Reward function

$R(s,a)$

POMDP Formulation: The Belief MDP Problem

Motion Model

$$x_{k+1} = f(x_k, u_k, w_k)\ ,\ p(w_k \mid x_k, u_k)$$



Sensing Model

$$z_k = h(x_k, v_k),\ p(v_k \mid x_k)$$

Belief Evolution Model

(belief state and its update)

$$b_k = p(x_k \mid z_{0:k}; u_{0:k-1})$$

$$b_{k+1} = \tau(b_k, u_k, z_{k+1})$$

Cost Function

(to choose an optimal policy)

The cost of taking action u at belief b is $c(b,u)$: $\mathbb{B}\times\mathbb{U} \to \mathbb{R}_{\geq 0}$

The cost-to-go function is $J^{\pi}(\cdot)$: $\mathbb{B}\times\mathbb{U} \to \mathbb{R}_{\geq 0}$ from b_0 under policy π as

$$J^{\pi}(b_0) := \sum_{k=0}^{\infty} \mathbb{E}[c(b_k, \pi(b_k))]$$

$$s.t. \quad b_{k+1} = \tau(b_k, \pi(b_k), z_{k+1}), \quad z_k \sim p(z_k \mid x_k)$$

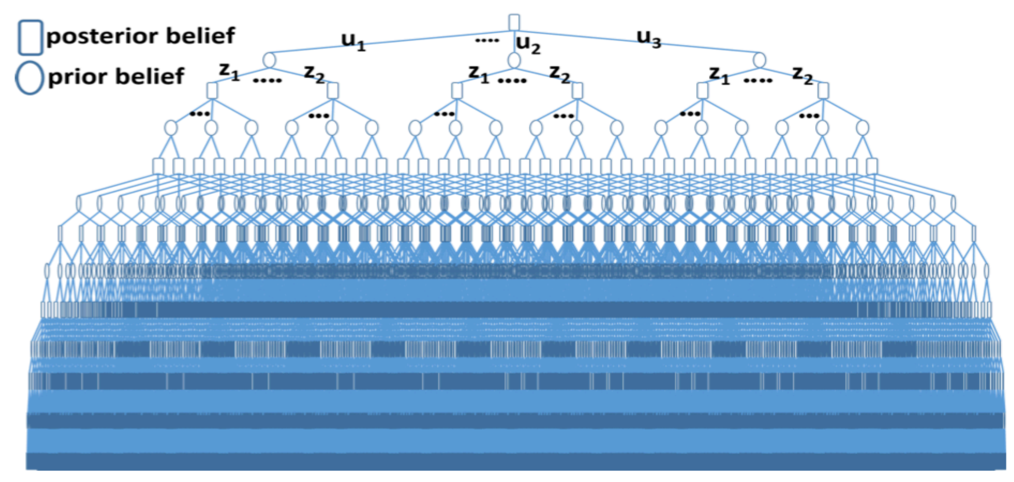
Policy

$$\pi(\cdot): \ \mathbb{B} \to \mathbb{U}:$$

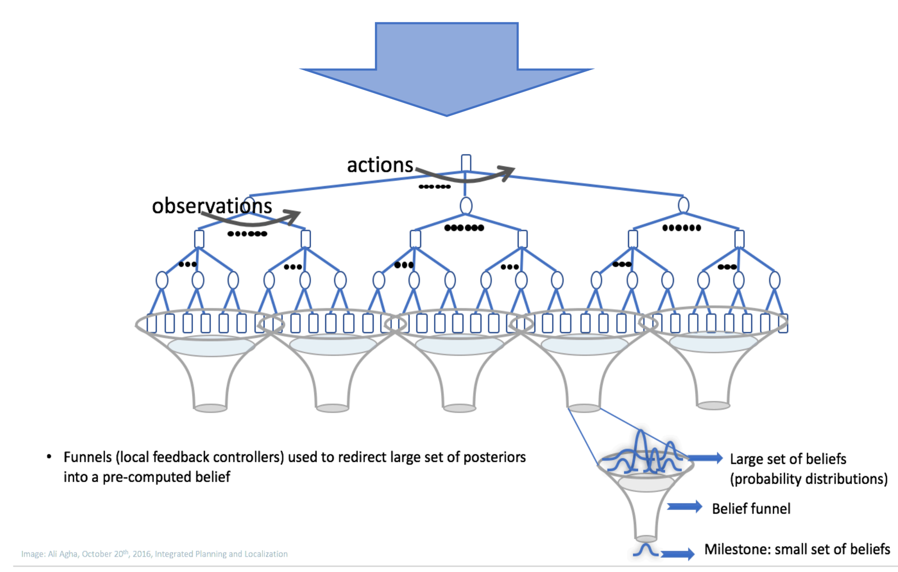
$$U_k = \pi(b_k), \ \forall b_k \in \mathbb{B}$$

Methods:

Problem: Curse of History



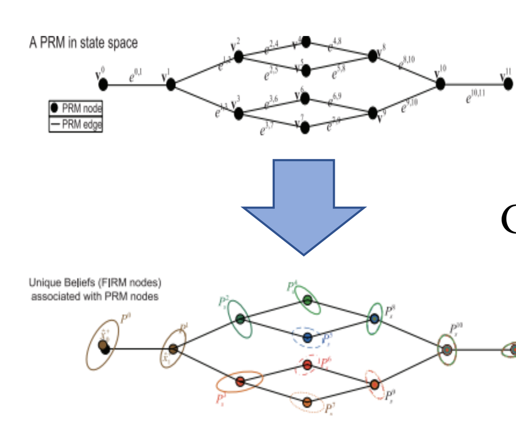
The "curse of history" is a problem presented by modeling the belief space by POMDPs. The exponential growth of outcomes for a single belief is shown here.



To break the curse of history, one can simply steer the belief directly using funnels (local feedback controllers) to redirect large sets of posteriors into a pre-computed belief.

FIRM (Feedback-based Information Roadmap)

Offline Planning



Gaussian Belief

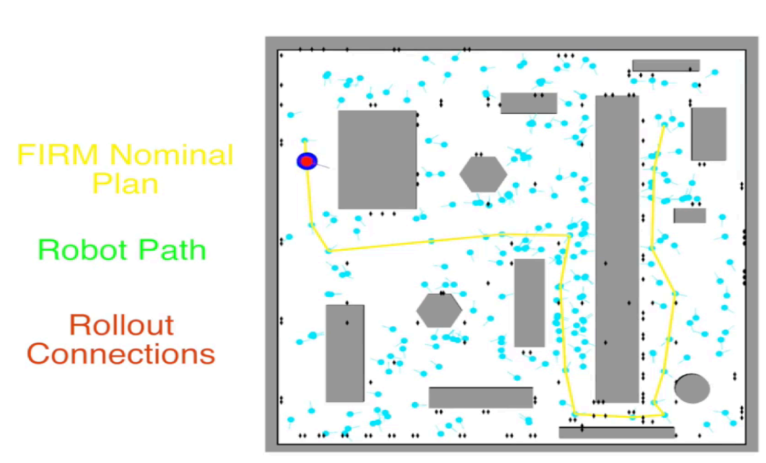
$$b \equiv (\hat{x}^{+}, P)$$

FIRM is the belief space variant of other probabilistic roadmaps.

- Updates roadmap at each belief state
- No longer encounter curse of history
- Preserves optimal substructure policy (no edge dependence), no need for expensive re-planning

Rollout

Online Planning



A robot following FIRM-based rollout plan.
Rollout is an online planner. 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. 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.

At each step in a rollout policy, the following closed-loop optimization is solved:

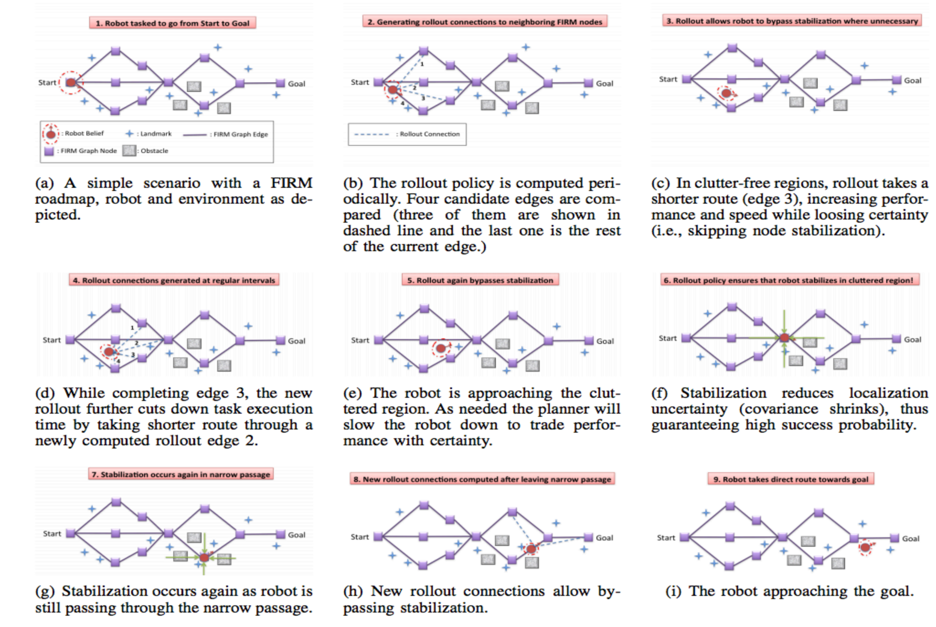
$$\pi_{0:T}(\cdot) = \arg\min_{\Pi_{0:T}} \mathbb{E}\left[\sum_{k=0}^{T} c(b_k, \pi_k(b_k)) + \widetilde{J}(b_{T+1})\right]$$

$$s.t. \ \ b_{k+1} = \tau(b_k, \pi_k(b_k), z_k), \quad z_k \sim p(z_k \mid x_k)$$

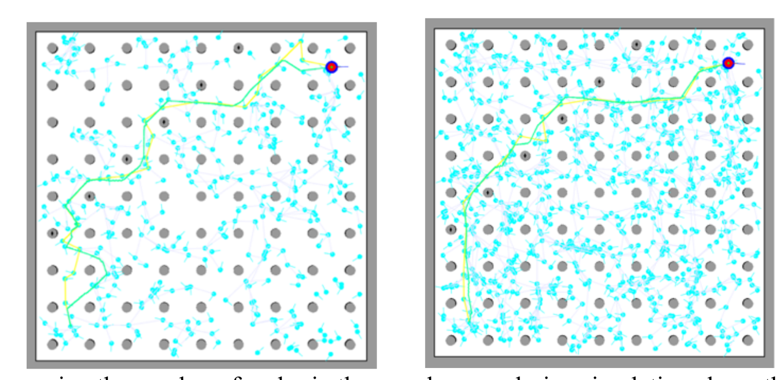
$$x_{k+1} = f(x_k, \pi_k(b_k), w_k), \quad w_k \sim p(w_k \mid x_k, \pi_k(b_k))$$

Results:

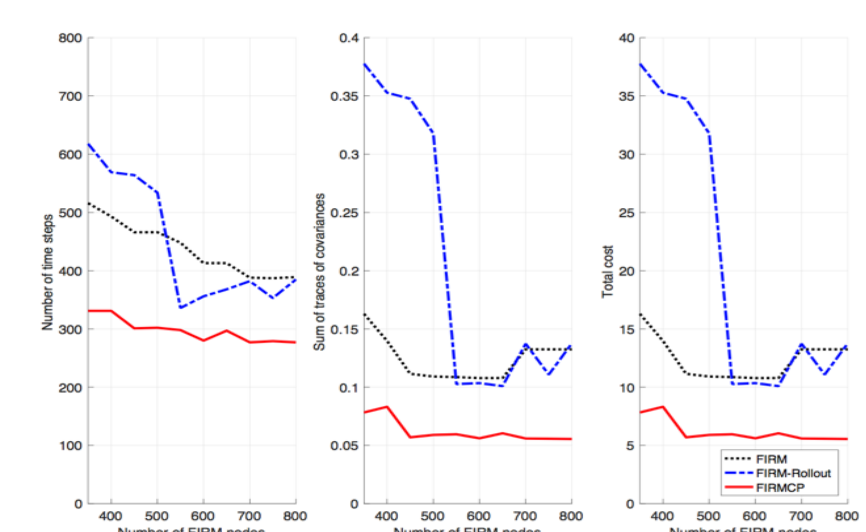
FIRM and Rollout Comparisons and Applications



Rollout-based execution of the FIRM policy.
FIRM does its planning offline while rollout does its planning online.

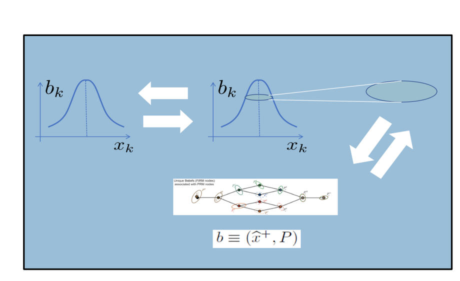


Increasing the number of nodes in the search space during simulation closes the gap between running the FIRM algorithm and the rollout algorithm. Here we see the path produced by FIRM's planner in yellow and that produced by rollout in green.

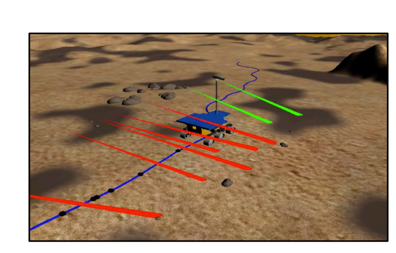


A graphic representation of simulation results compares the number of FIRM nodes with the number of time steps, the sum of traces of covariance, and the total cost of FIRM, FIRM-Rollout, and FIRMCP

Conclusions:



- Previous path planning methods plan in a state space to incorporate motion capabilities and FIRM takes things a step further and plans in the belief space to enable motion and sensing capabilities
- FIRM solves the problem of the curse of history in POMDPs and allows large problems to be solved



- The optimality of POMCPs (POMDPs with Monte Carlo planning) and the scalability of FIRM to produce FIRMCP, a near-optimal long range belief planner

- POMDPs allow us to model real world situations because they account for sensing and motion uncertainties in an environment

Proposed Future Work:

The real world is full of uncertainty. 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 that was 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 towards solving the ultimate minimization problem—that is the one presented by minimizing the uncertainties that come with modeling in space exploration.