Technion – Israel Institute of Technology

Planning in the Continuous Domain: A Generalized Belief Space Approach for Autonomous Navigation in Unknown Environments

Vadim Indelman, Luca Carlone, and Frank Dellaert

International Journal of Robotics Research - 2015

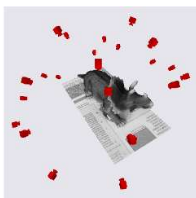
Submitted by:

Alon Spinner	305184335	alonspinner@gmail.com
Dan Hazan	308553601	Danhazzan4@gmail.com

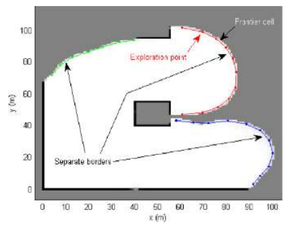
June 29, 2022

Introduction and Motivation

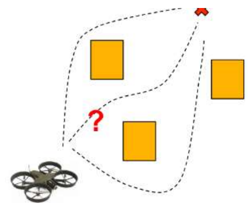
This work investigates the problem of planning under uncertainty with application to mobile robotics. It proposes a general framework where the robot bases its decisions on the probabilistic description of its own state and of external variables of interest - on the general belief space (GBS). Common applications for such work include path planning, active exploration, and "next best view".



Next best view



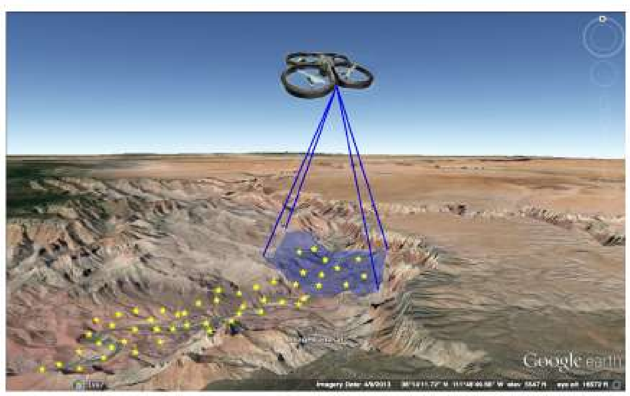
Active Exploration



Path Planning

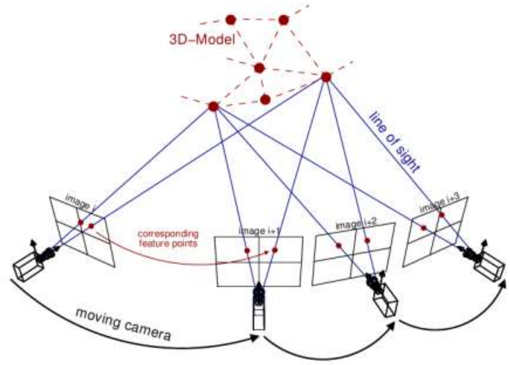
There are three limitations that are common in the related work that the authors do not partake in: discretization, maximum likelihood assumption and the availability of prior knowledge. The proposed method relays on a dual-layer architecture: an inner layer that infers a cost of some future GBS for a given set of actions, and an outer layer which optimizes those set of actions.

The authors research their proposed method using a simulation of a UAV that hovers above an unknown environment and is required to reach goals without passing some localization uncertainty limit.



General Belief Space Inference (SLAM)

To understand how one performs inference on GBS in the future, we first focus on GBS inference in the present.
We focus on the researched example, of a UAV hovering over unknown terrain, the GBS is over a state of UAV poses and landmarks.



$$X_k = \{x_0, \dots x_k, l_0 \dots l_n\}$$

robot poses landmarks

Given a collection of measurements Z and actions U (data history) and motion and observation models, one can factor the inference problem as follows:

$$p(X_k|\mathcal{Z}_k, \mathcal{U}_{k-1}) \sim priors \cdot \prod_{i=1}^{k}\left[p(x_i|x_{i-1}, u_{i-1}) \prod_{j=1}^{n_i} p\left(z_{i,j}, X_{i,j}^{o}\right)\right]$$

Motion model Observation model

$$x_i = f(x_{i-1}, u_{i-1}) + w_i \quad w_i \sim N\left(0, \Omega_w\right)$$
$$z_{i,j} = h\left(x_i, l_j\right) + v_{i,j} \quad v_i \sim N\left(0, \Omega_v\right)$$

Assuming the priors, motion model and observation models are all gaussian in nature, the posterior distribution over the state is also gaussian. To find the maximum of the posterior distribution one can search for the minimum of its negative logarithm (log is a strictly increasing function). This will then lead to a least-squares problem.

$$\underset{\mathrm{X_k}}{argmax}\ \mathrm{p}(\mathrm{X_k}|\mathcal{Z}_\mathrm{k}, \mathcal{U}_{\mathrm{k-1}}) = \underset{\mathrm{X_k}}{argmin}\left\{-\log\left(\mathrm{p}(\mathrm{X_k}|\mathcal{Z}_\mathrm{k}, \mathcal{U}_{\mathrm{k-1}})\right)\right\}$$

$$X_K^* = \underset{X_k}{argmin}\left\{||x - \hat{x}_0||_{\Omega_0}^2 + \sum_i\left[||x_i - f(x_{i-1}, u_{i-1})||_{\Omega_{w_i}}^2 + \sum_{j=\mathcal{M}_i}\left||z_{i,j} - h\left(x_i, l_j\right)\right||_{\Omega_{v_{i,j}}}^2\right]\right\}$$

Prior Motion Models Observation models

the popular approaches for solving the least squares are iterative Gauss–Newton or Levenberg–Marquardt. In those, one is required to linearize the cost function on some point and optimizes around it in the optimization step.

Linearize models around $\tilde{X}_k$

$$\Delta X_k = X_k - \tilde{X}_k$$

$$(\Delta X_k)^* = \underset{\Delta X_k}{argmin} \left\{ \left|\left| \mathrm{A} \Delta X_k - b \right|\right|^2 \right\}$$

$$(\Delta X_k)^* = (\mathrm{A}^T \mathrm{A})^{-1} \mathrm{A}^T b$$

$$\tilde{X}_k \longleftarrow \tilde{X}_k + (\Delta X_k)^*$$

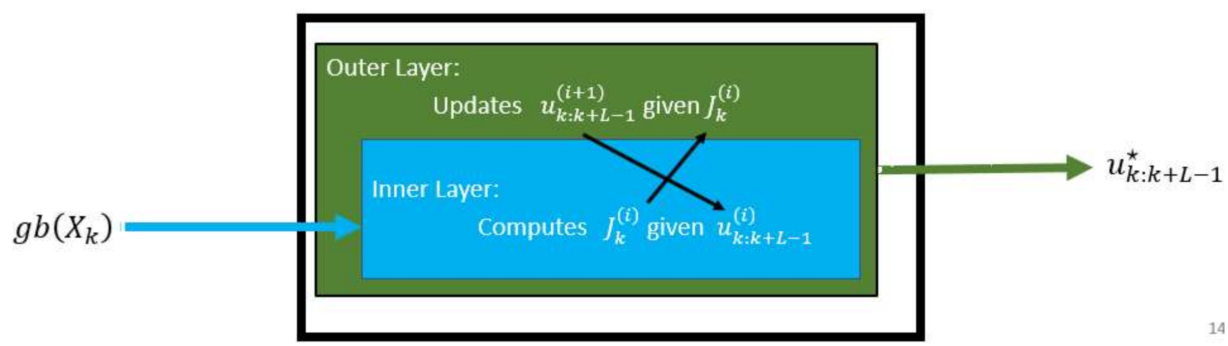
A key observation of which the authors take advantage is that for the first optimization iteration, the matrix A is not a function of the measurements.

Planning in the General Belief Space

The planning method describes in the article provides a way of obtaining an optimal set of actions given the gaussian GBS. The algorithm computes a cost function for each set of actions via a bellman objective function of the following form:

$$J_k\left(u_{k:k+L-1}\right) \doteq \underset{Z_{k+1:k+L}}{\mathbb{E}} \left\{ \sum_{l=0}^{L-1} c_l\left(gb\left(X_{k+l}\right), u_{k+l}\right) + c_L\left(gb\left(X_{k+L}\right)\right) \right\}$$

To optimize the actions with the objective function, a dual-layer architecture was devised. The inner layer is responsible for computing the objective function, and the outer layer is responsible of the suggesting a new set of actions given the objective function just computed.



$$u_{k:k+L-1}^{\star} \doteq \left\{u_k^{\star}, \ldots, u_{k+L-1}^{\star}\right\} = \underset{u_{k:k+L-1}}{\arg\min}\ J_k\left(u_{k:k+L-1}\right)$$

Outer Layer

The outer layer is responsible for updating the set of actions given the objective function J_k such that they will minimize J_k. The method suggested is gradient descent with numerical differentiationn.

$$u_{k:k+L-1}^{(i+1)} = u_{k:k+L-1}^{(i)} - \lambda \ \nabla J_k^{(i)} \qquad \left\|\nabla J_k^{(i)}\right\| < \epsilon \text{ or } \left\|\frac{J_k^{(i+1)} - J_k^{(i)}}{J_k^{(i+1)}}\right\| < \epsilon \text{ or } i \geq i_{max}$$

$$\frac{\partial J_k}{\partial \boldsymbol{u}[m]} \approx \frac{J_k(\boldsymbol{u} + \epsilon \cdot \boldsymbol{e}_m) - J_k(\boldsymbol{u})}{\epsilon} \qquad \boldsymbol{u} = \begin{bmatrix} u_k \\ \vdots \\ \vdots \\ u_{k+L-1} \end{bmatrix} \qquad \boldsymbol{e}_m = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix} \longleftarrow \boldsymbol{e}_m[\boldsymbol{m}] = \mathbf{1}$$

$$\nabla J_k = \left[\frac{\partial J_k}{\partial \boldsymbol{u}[0]} \cdots \frac{\partial J_k}{\partial \boldsymbol{u}[m]}, \ldots, \frac{\partial J_k}{\partial \boldsymbol{u}[L]}\right]^T$$

Inner Layer

The inner computes the objective function via the gaussian approximations of future GBSs' $gb(X_{k+l}) \sim N(X_{k+l}^{\star}, I_{k+l})$.

Number of complications arise when one tries to solve for the posterior of the general belief $gb(X_{k+l})$. The first is that we don't know if a future observation $z_{k+l,j}$ will be obtained, and the second is that wihout assuming ML measurements, we can't assume its value. These complications lead to definition of two new random variabels $z_{i,j}$ and $\gamma_{i,j}$. The first is observation at time i of the jth landmark and the second is $binary$ random variable for each observation (acquired or not). These two random variabels lead to new joint probability density:

$$p\left(X_{k+l}, \Gamma_{k+1:k+l}, Z_{k+1:k+l} \middle| \mathcal{Z}_k, \mathcal{U}_{k-1}, u_{k:k+l-1}\right)$$

- Noted that $\Gamma_i \doteq \{\gamma_{i,j}\}_{j=1}^{n_i}$ being the number of possible observations at time t_i.

Using chain rule, we can write the joint probability as:

$$p\left(X_{k+l}, \Gamma_{k+1:k+l}, Z_{k+1:k+l} \middle| \mathcal{Z}_k, \mathcal{U}_{k-1}, u_{k:k+l-1}\right) =$$
$$p\left(X_{k+l}, \Gamma_{k+1:k+l} \middle| \mathcal{Z}_k, \mathcal{U}_{k-1}, Z_{k+1:k+l}, u_{k:k+l-1}\right) p\left(Z_{k+1:k+l} \middle| \mathcal{Z}_k, \mathcal{U}_{k-1}, u_{k:k+l-1}\right)$$

As the term $p(Z_{k+1:k+l} \,|Z_k, U_{k-1}, u_{k:k+l-1})$ is uninformative, we can use the following better:

$$p\left(X_{k+l}, \Gamma_{k+1:k+l} | \mathcal{Z}_k, \mathcal{U}_{k-1}, Z_{k+1:k+l}, u_{k:k+l-1}\right) \propto p\left(X_{k+l}, \Gamma_{k+1:k+l}, Z_{k+1:k+l} | \mathcal{Z}_k, \mathcal{U}_{k-1}, u_{k:k+l-1}\right)$$

To introduce these terms into the posterior belief $gb(X_{k+l})$ one would marginalize the latent variable $\Gamma_{k+1:k+l}$ and get:

$$gb\left(X_{k+l}\right) = p\left(X_{k+l} | \mathcal{Z}_k, \mathcal{U}_{k-1}, Z_{k+1:k+l}, u_{k:k+l-1}\right) = \sum_{\Gamma_{k+1:k+l}} p\left(X_{k+l}, \Gamma_{k+1:k+l} | \mathcal{Z}_k, \mathcal{U}_{k-1}, Z_{k+1:k+l}, u_{k:k+l-1}\right)$$

The next step is to compute the MAP using the expression above. But this would prove intractable due to the marginalization integral. Hence, a different approach using expectation maximization is taken, where, Minka-1998 guarantees it will converge to a local minimum.

$$X^\star_{k+l} = \underset{X_{k+l}}{\arg\min} \underset{\Gamma_{k+1:k+l}|X_{k+l}}{\mathbb{E}} \left[-\log p\left(X_{k+l}, \Gamma_{k+1:k+l} | \mathcal{Z}_k, \mathcal{U}_{k-1}, Z_{k+1:k+l}, u_{k:k+l-1}\right)\right]$$

Plugging in the probabilistic models to compute the explicit expression for the expectation in the $argmin$ problem above, the authors arrive at the following form:

$$X^\star_{k+l} = \underset{X_{k+l}}{\arg\min} \|X_k - X^\star_k\|^2_{I_k} + \sum_{i=1}^{l} \|x_{k+i} - f\left(x_{k+i-1}, u_{k+i-1}\right)\|^2_{\Omega_w} + \sum_{i=1}^{l}\sum_{j=1}^{n_i} p\left(\gamma_{k+i,j} = 1 | \bar{X}_{k+l}\right) \left\|z_{k+i,j} - h\left(X^o_{k+i,j}\right)\right\|^2_{\Omega^{ij}_v}$$

The first term is the prior information - Gaussian approximation of the generalized belief at the current time. The second and third terms describe the influence of future controls and measurements on the belief.

The authors define an adjusted information matrix and rewrite the *argmin* problem

$$\bar{\Omega}^{ij}_v = p(\gamma_{k+i,j} = 1 | \bar{X}_{k+l}) \Omega^{ij}_v$$

$$X^\star_{k+l} = \underset{X_{k+l}}{\arg\min} \|X_k - X^\star_k\|^2_{I_k} + \sum_{i=1}^{l} \|x_{k+i} - f\left(x_{k+i-1}, u_{k+i-1}\right)\|^2_{\Omega_w} + \sum_{i=1}^{l}\sum_{j=1}^{n_i} \left\|z_{k+i,j} - h\left(X^o_{k+i,j}\right)\right\|^2_{\bar{\Omega}^{ij}_v}$$

The linearization point $\bar{X}_{k+l}(u_{k:k+l-1})$ is computed as follow: the subset of past states (until time t_k) in $\hat{X}_{k+l}(u_{k:k+l-1})$ is set to $X^\star_k$, while the future states are predicted via the motion model using the current values of the controls $u_{k:k+l-1}$:

$$\bar{X}_{k+l}\left(u_{k:k+l-1}\right) \equiv \begin{pmatrix} \bar{X}_k \\ \bar{x}_{k+1} \\ \bar{x}_{k+2} \\ \vdots \\ \bar{x}_{k+l} \end{pmatrix} \doteq \begin{pmatrix} X_k^\star \\ f\left(x_k^\star, u_k\right) \\ f\left(\bar{x}_{k+1}, u_{k+1}\right) \\ \vdots \\ f\left(\bar{x}_{k+l-1}, u_{k+l-1}\right) \end{pmatrix}$$

Formulating the Objective Function

Given the scenario at hand, the following cost functions were written to follow three principles: Minimizing control effort to create smooth trajectories, minimizing the robot's pose uncertainty, and getting as close as possible to the goal.

$$c_l\left(gb\left(X_{k+l}\right), u_{k+l}\right) \doteq tr\left(M_\Sigma I_{k+l}^{-1} M_\Sigma^T\right) + \left\|\zeta\left(u_{k+l}\right)\right\|_{M_u}^2$$

$$c_L\left(gb\left(X_{k+L}\right)\right) \doteq \left\|E_{k+L}^G X_{k+L}^\star - X^G\right\|_{M_x}^2 + tr\left(M_\Sigma I_{k+L}^{-1} M_\Sigma^T\right)$$

The terms M_Σ, M_x, M_u are uncertainty matrices, and E^G is a selection matrix. ζ is a function on the control, calculating the control effort.

Plugging the cost functions into the objective function, we obtain the following:

$$J_k\left(u_{k:k+L-1}\right) \doteq \overbrace{\sum_{l=0}^{L-1} \left\|\zeta\left(u_{k+l}\right)\right\|_{M_u}^2}^{(a)} + \overbrace{\sum_{l=0}^{L} tr\left(M_\Sigma I_{k+l}^{-1} M_\Sigma^T\right)}^{(b)} +$$
$$\underbrace{\left[\left\|E_{k+L}^G \bar{X}_{k+L} - X^G\right\|_{M_x}^2 + tr\left(Q_{k+L}\left(H_{k+L} \bar{I}_{k+L}^{-1} H_{k+L}^T + \Omega_v^{-1}\right)\right)\right]}_{(c)},$$

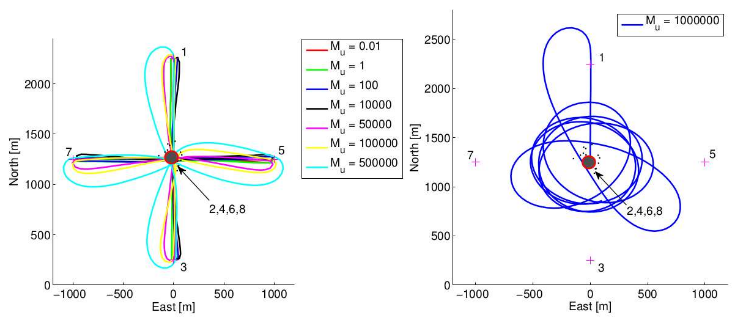
$$Q_{k+L} = \left(E_{k+L}^G I_{k+L}^{-1} \mathcal{H}_{k+L}^T \breve{\Omega}_v^{\frac{1}{2}}\right)^T M_x \left(E_{k+L}^G I_{k+L}^{-1} \mathcal{H}_{k+L}^T \breve{\Omega}_v^{\frac{1}{2}}\right)$$

Term (a) is not a function of the measurements, and hence is just a sum over the control effort.
Under the assumption that one iteration of the gauss-newton suffices for future estimation of the GBS, I_{k+l}^{-1} is also not a function of the measurements, hence the sum in term (b).

Term (c) is the outcome of the expectancy over $\left|\left|E^G_{k+L}X^\star_{k+L} - X^G\right|\right|^2_{M_x}$. It splits to two terms, the first is the distance of the last pose of the linearization point from the goal, and the second is a complex term that would nullify under the assumption of maximum likelihood measurements.

Choice of M_u

To show the effect that each weight has on the planning algorithm, the authors provided an example of a "petal flower" planning problem.
When M_u is higher, the robot is seen to make bigger turns as changing the turning angle is expensive.



(a) (b)

Choice of M_x and M_Σ

In order to balance pose uncertainty against the distance to goal in the decision process, a parameter β was introduced, which serves as soft upper limit of the pose's trace.
From β a variable α_k determining the uncertainty balance can be derived, where high values of α_k correlate to low uncertainty budget.
This variable is than used to balance the weight matrices M_x and M_Σ, and is capped at a maximum of 1.0, to avoid negative weights.

$$tr\left(\bar{M}_\Sigma I^{-1}_{k+L}\bar{M}^T_\Sigma\right) \leq \beta \longrightarrow \alpha_k = \frac{tr\left(\bar{M}_\Sigma \bar{I}^{-1}_{k+L}\bar{M}^T_\Sigma\right)}{\beta}$$

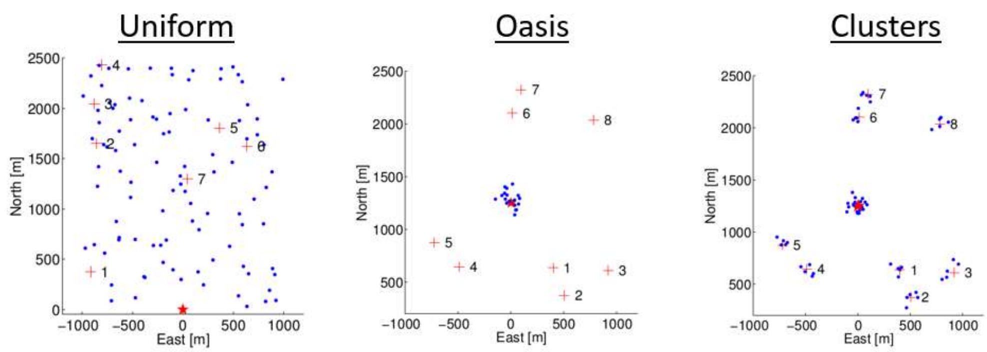
$$M_x = \alpha_k \underbrace{\bar{M}_x}_{\text{Selection Matrix}} \qquad M_\Sigma = \sqrt{1-\alpha_k}\,\underbrace{\bar{M}_\Sigma}_{\text{Selection Matrix}}$$

$$\alpha_k = \frac{tr\left(\bar{M}_\Sigma \bar{I}^{-1}_{k+L}\bar{M}^T_\Sigma\right)}{\beta} \xrightarrow{\alpha_k \text{ can't be bigger than 1}} \alpha_k = \min\left(\frac{tr\left(\bar{M}_\Sigma \bar{I}^{-1}_{k+L}\bar{M}^T_\Sigma\right)}{\beta}, 1\right)$$

Experimental Evaluation

Beyond other evaluations tests of horizon length, and robustness to different noise models, the authors checked their work in three scenarios (presented below), and compared to 3 planning algorithms (4 in total):

- GBS – the presented method
- GBS-ML – the presented method with the assumption of Maximum likelihood (last term nullifies)
- CNU – the presented method with no weight given to uncertainty in planning (measurements not involved in planning).
- Discrete - method by Kim, A. and Eustice, R. (2013)

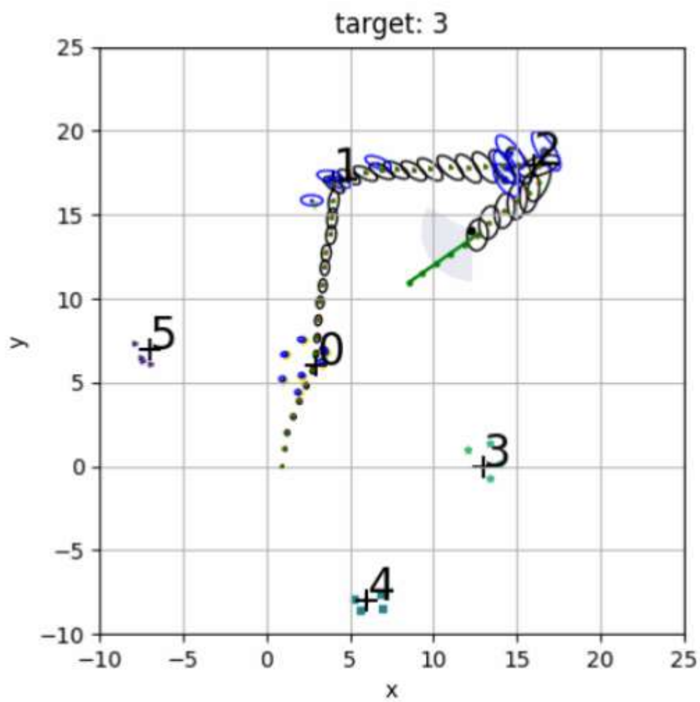


In all scenarios the results were consistent:

- CNU provides the largest estimation error
- GBS and GBS-ML are very similar in performance
- Planning time in GBS and GBS-ML increases when turns are required (more iterations in gradient decent)
- The discrete approach can produce a plan that satisfies soft uncertainty, but the 'zig zag' actions make sure that the control effort is high.

Implementation

We implemented the proposed method via GTSAM in a python environment with some modifications. The robot moves at a constant velocity, and our control is only on the steering, hence, we decided to minimize the distance of any planned pose to goal (rather than the last).
Additionally, we used middle derivative for the gradient decent, making the process more stable.
Lastly, using isam2, we had not found a way to ensure only 1 gauss-newton iteration, forcing us to change the cost function to include a different term that encourages loop closures by punishing the distance from informative landmarks: $J_{LC} \sim r(\bar{x}\,, l)\mathrm{trace}\big(\Omega(l)\big)$



Possible Extensions

The authors provide an extensive analysis of the weakness of their method. The first relates to the problematic nature of Quadratic costs, where the stimuli to achieve a goal is reduced as one gets closer to it.
One potential offered solution is use L_1 norms, but these are not differentiable at 0, and the derivation of the cost function will be very difficult.

The second point is concerned with the soft uncertainty constraint built into the planning algorithm. The experimental results showed that the robot does not continue to chance the goal for long after the uncertainty budget is cleared, but perhaps one can ensure that the change to uncertainty reduction planning is immediate. For this the authors propose two solutions, the first is a logarithmic barrier, ensuring infinite objective function cost to break the uncertainty budget, and the second is solving a constraint optimization.

Furthermore, the authors address the problem of "out of range" goals, where the uncertainty budget would not allow the robot to reach the goal, having it go back and forth from it, creating "loopy solutions".
To avoid this, one can devise an adaptive uncertainty budget, an adaptive β, that increases when the goal is deemed to be too far away, and no loop closures for uncertainty reduction can help.

Lastly, the authors talk about increasing the efficiency of the gradient decent solution with smarter optimization solvers presented in recent years.

We would add to those the following:

- exact differentials with Lie Algebra for faster objective function derivative computation. The inference tools already possess those capabilities.
- Smart initialization of the action set for faster convergence. For example, after loop closure has occurred, a good first guess for the next set of actions would be those that go towards the goal. This will reduce the "turning cost" by a lot.
- We noticed that earlier actions in the action set are more significant than later ones due to nature of actions being the derivatives of states. We would think that optimizing for a shorter horizon and then increasing it can be beneficial.