Efficient Decision Making under Uncertainty in High-Dimensional State Spaces

Khen Elimelech

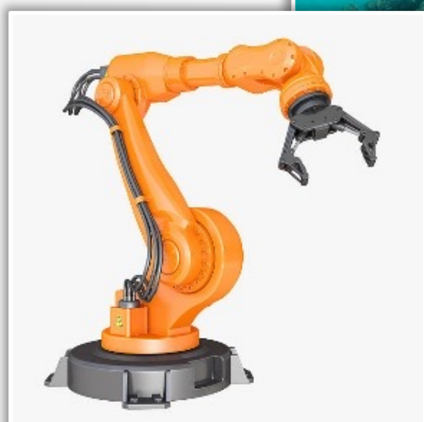
under the supervision of Assoc. Prof. Vadim Indelman

Ph.D. Seminar, May 2021

ANPL | Autonomous Navigation and Perception Lab

TECHNION Israel Institute of Technology

Autonomous Systems



Robust Autonomous Systems

- Need to answer questions, such as:

- Where am I?
- What's around me?

➡ State estimation, inference, mapping…

- Where to go?
- How to get there?

➡ Planning, control, decision making…

- Accounting for uncertainty is essential for reliability

In this talk…

- Online decision making under uncertainty

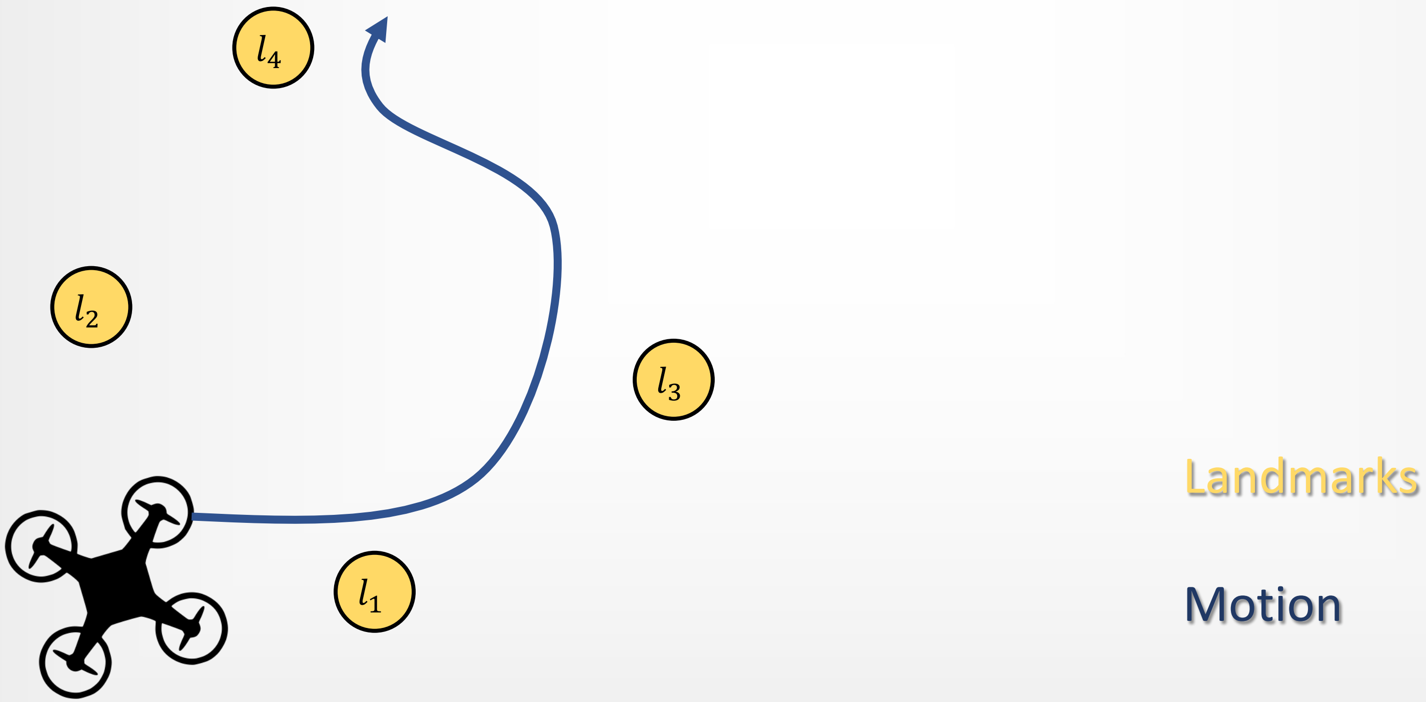
- In the context of (but not limited to) SLAM: Simultaneous localization and mapping

1. State estimation and SLAM
2. Decision making under uncertainty
3. Contributions:
   I. Efficient DM via belief sparsification (+ results)
   II. Efficient DM via predictive reordering (+ results)

Efficient Decision Making ***under Uncertainty*** in High-Dimensional ***State*** Spaces

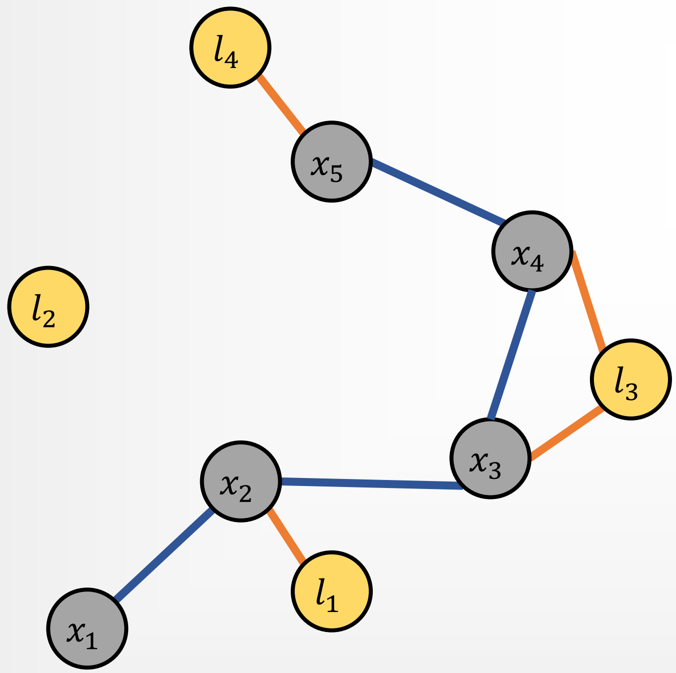
Example Scenario

- A robot navigating in an unknown environment
- Observes features/landmarks around it
- Wants to infer its location in the environment



Example Scenario: Discretized

- Factor graph: a graph of constraints
- Constraints are defined according to the controller and sensor models



Poses
Landmarks
Observation constraint
Motion constraint

Stochastic Constraints => Beliefs

- The motion model is stochastic (Markov assumption):

$$x_k = g(x_{k-1}, u_k) + \text{noise}$$

- The observation model is also stochastic:

$$z_{k,l} = h(x_k, l) + \text{noise}$$

- At each time-step, these constraints induce a belief:

$$b(\boldsymbol{X}_k) \doteq \mathbb{P}(\boldsymbol{X}_k | u_{1:k}, z_{1:k})$$

- The posterior distribution over the state, given past controls and observations.

A Big Optimization Problem

- From the belief b_k (distribution), we wish to find the MAP estimate of the state vector $\boldsymbol{X}_k$
- E.g., robot poses, and position of landmarks

- Offline global optimization:

  Structure from Motion (SfM) / Bundle Adjustment (BA)

- Online iterative optimization, as time progresses:

  Simultaneous localization and mapping (SLAM)

Efficient Decision Making under Uncertainty in ***High-Dimensional*** State Spaces

Traditional Approach: Bayesian Filtering

- E.g. Kalman filter, information filter, particle filter
- Tells us how to update the belief given new actions/observations

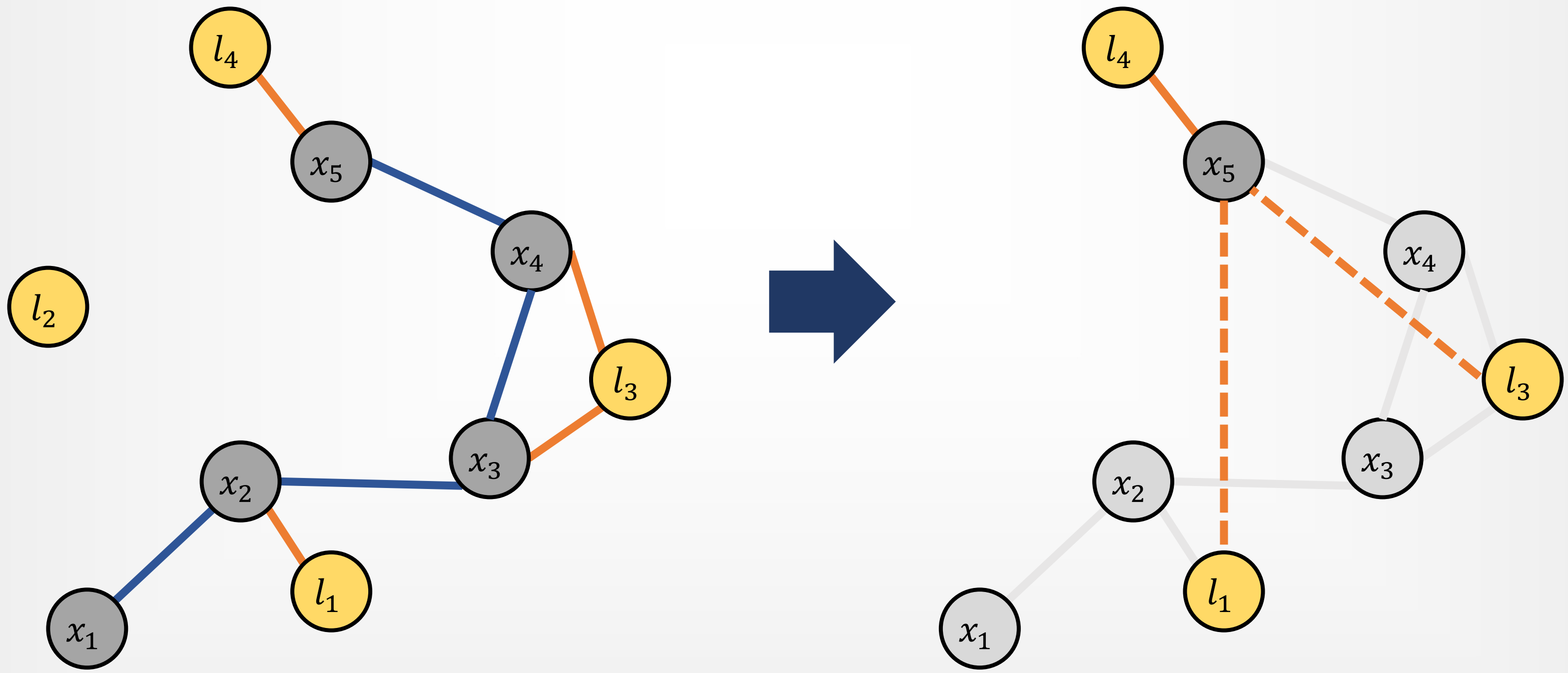
$$b(X_{k+1}) \propto \mathbb{P}(z_{k+1}|X_{k+1}) \int b(X_k) \cdot \mathbb{P}(x_{k+1}|x_k, a_k)\, dx_k$$

- Marginalization of (“forgetting”) previous poses
- We only maintain the most recent pose in the belief:

$$X_k \doteq [x_k, L]^T$$

Traditional Approach: Bayesian Filtering

- Advantages: smaller state size
- Disadvantage: dense(r) system,
  cannot update estimate of past poses



Smoothing and Mapping

$$b(X_{k+1}) \propto b_k \cdot \mathbb{P}(z_{k+1}|X_{k+1}) \cdot \mathbb{P}(x_{k+1}|x_k, a_k)$$

- State vector contains the entire trajectory

$$X_k \doteq [x_{1:k}, L]^T$$

- No marginalization of past poses
- Pose-SLAM vs. full-SLAM

- More accurate estimation (updatable past poses)
- <u>High-dimensional states</u>, estimation cost grows quickly

Common (Non-Essential) Assumptions

- Gaussian noise and linear(ized*) models

- Leads to Gaussian beliefs:

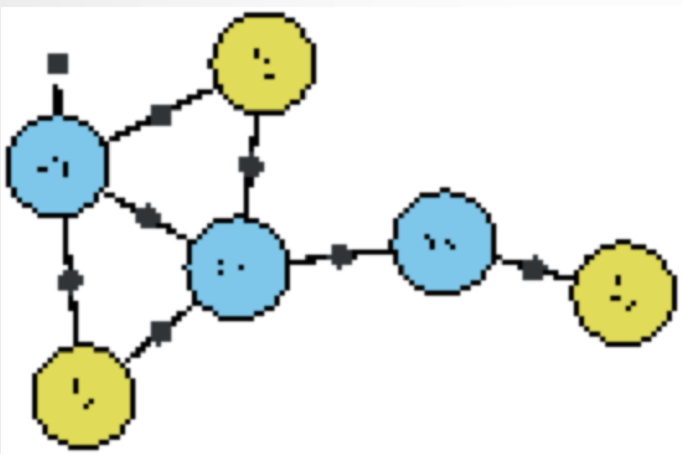
$$b(\boldsymbol{X}_k) \doteq \mathbb{P}(\boldsymbol{X}_k | u_{1:k}, z_{1:k}) \approx \mathcal{N}(\boldsymbol{X}_k^*, \Sigma_k)$$

- Can be described with two components:
  - Mean vector (the MAP estimate)
  - Information matrix Λ_k (the constraints)

* For brevity, not discussed here.

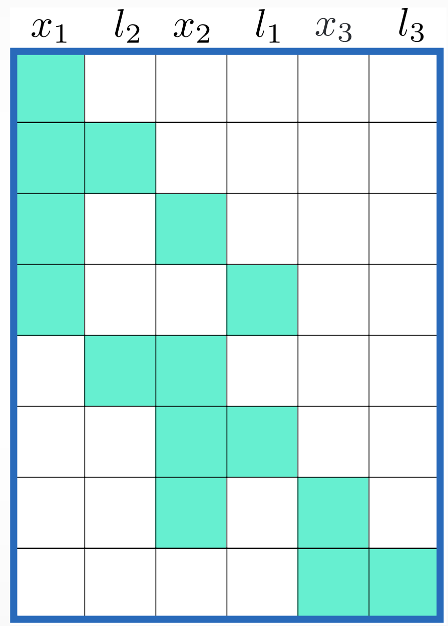
Constraints ⇔ Graphs ⇔ Matrices

Factor Graph



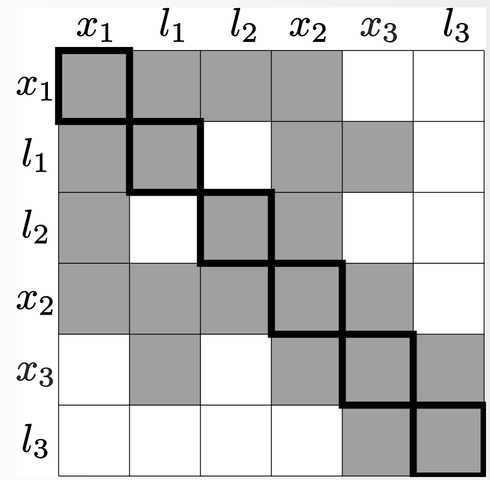
"Jacobians Matrix"

A_k



Information Matrix

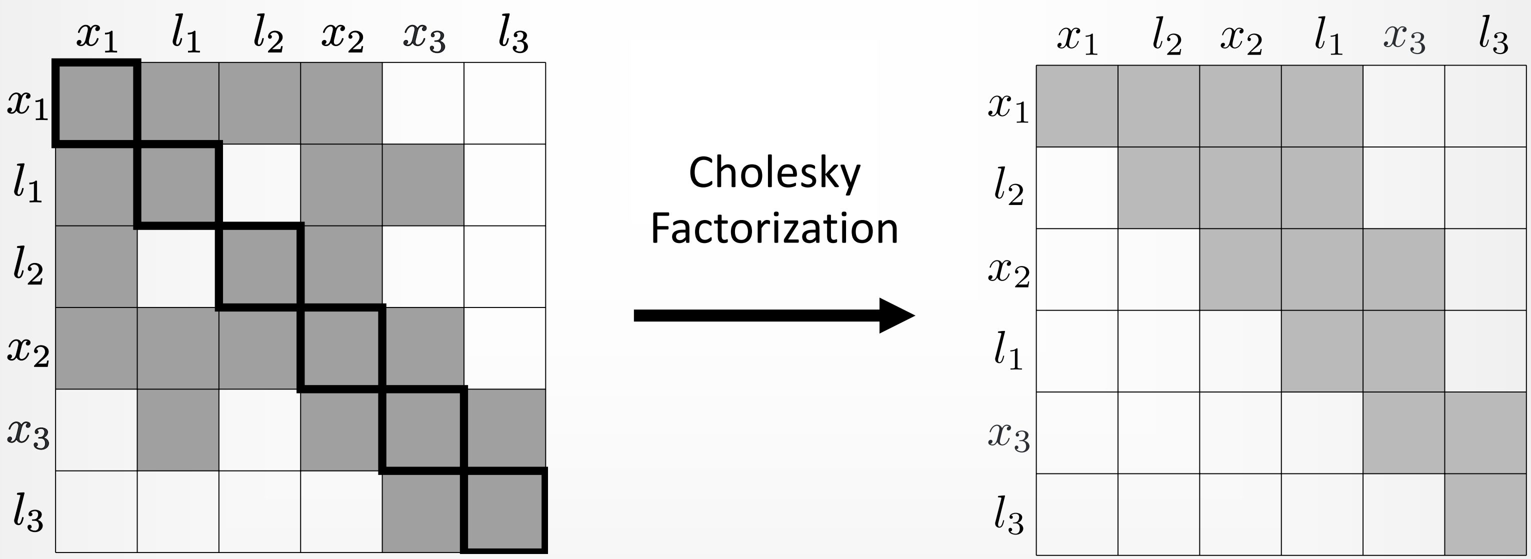
$\Lambda_k \doteq A_k^{\mathrm{T}} A_k$



- Given the constraints, we wish to find the MAP estimate

Belief Factorization

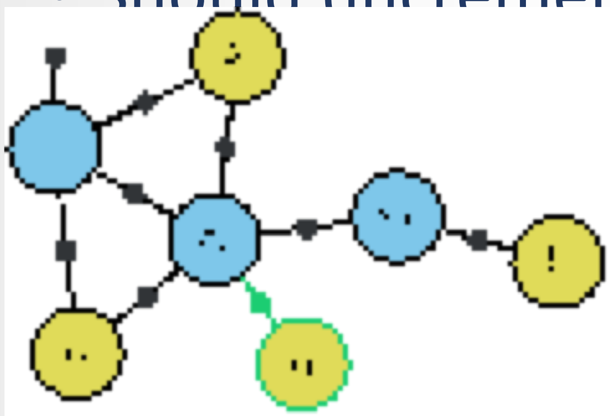
- Skipping some equations…
- In practice, looking for the upper triangular “square root” of Λ_k, such that $\Lambda_k \doteq R_k^T R_k$

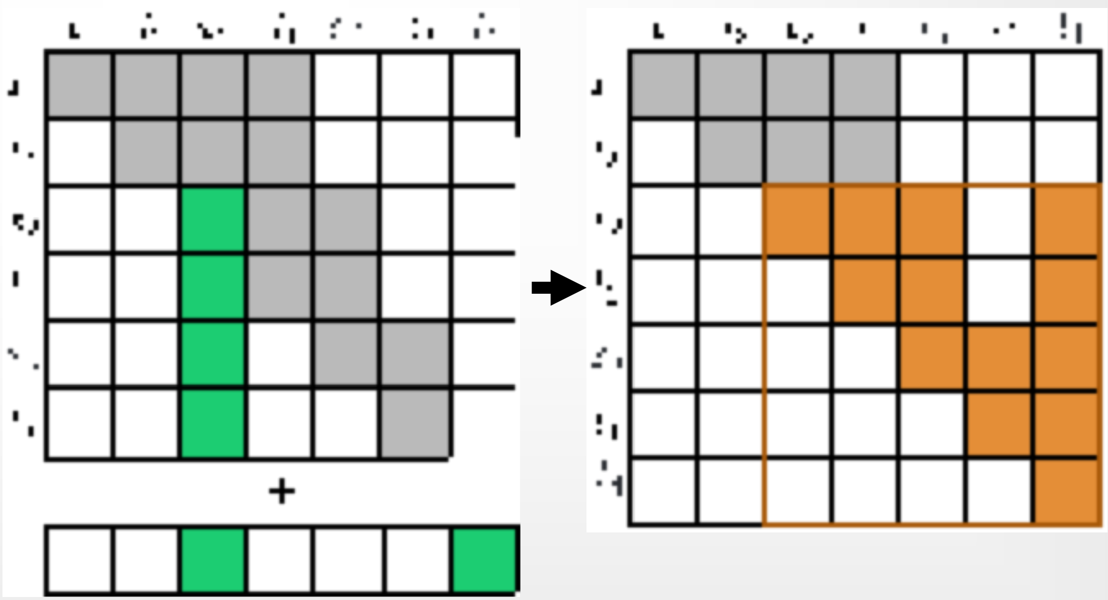


- At worst, factorization holds a quadratic cost

Belief Factorization Update

- New constraints are represented with “Jacobian” rows
- Should (incrementally) update the square root





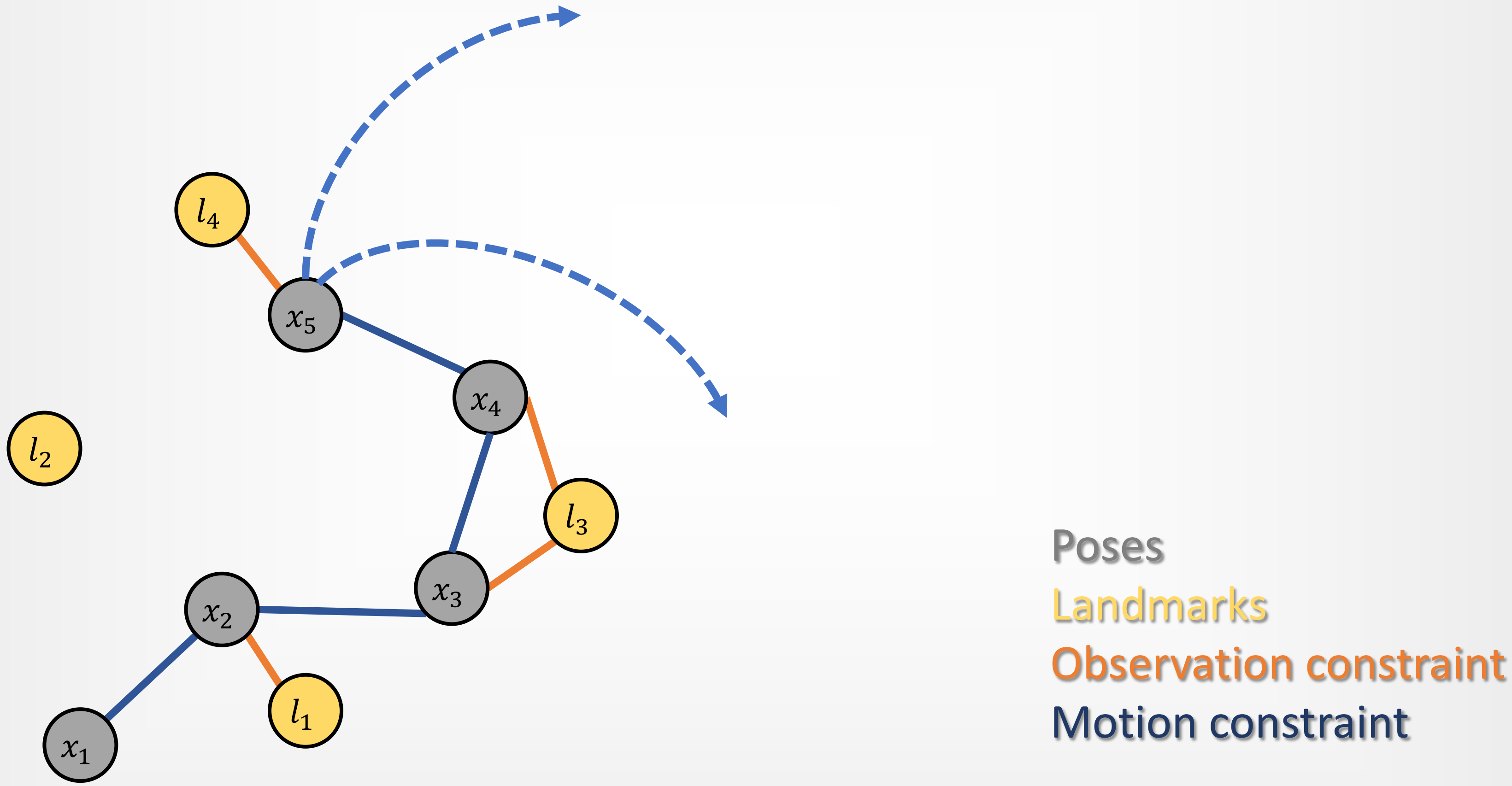
Recap: State Estimation (Inference)

- In sequential estimation, we gradually collect a set of constraints over a state vector of variables of interest
- We wish to maintain an up-to-date state estimate throughout the process
- The smoothing paradigm suggests no pose marginalization, i.e., examining high-dimensional states
- At each time-step, the set of constraints induces a belief: the posterior distribution over the state
- To find the estimate, we shall find (and maintain) the belief's upper triangular square root matrix

Efficient *Decision Making* under Uncertainty in High-Dimensional State Spaces

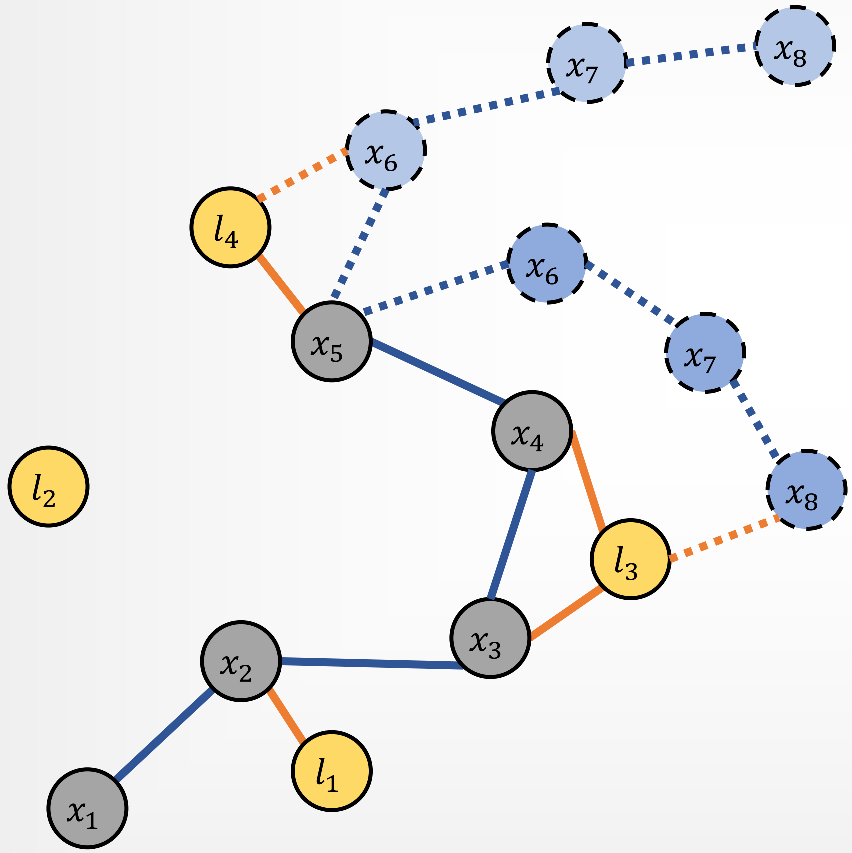
Planning in the Belief Space

- We wish to plan the next action (sequence)



Planning in the Belief Space

- Predict belief development (new poses/constraints) for multiple candidate actions/policies



Poses
Landmarks
Observation constraint
Motion constraint
Dashed: predicted

Decision Problems

- Planning is comprised of many sub-problems, e.g.:
  - action generation (motion planning)
  - motion prediction
  - reward engineering
  - <u>candidate comparison</u>

- A decision problem $\mathcal{P} \doteq (b, \mathcal{A}, V)$:
- Given a set $\mathcal{A}$ of candidate actions, we wish to find the optimal one, according to the objective function V:

$$a^* = \underset{a \in \mathcal{A}}{\operatorname{argmax}} V(b, a)$$

Measuring Uncertainty

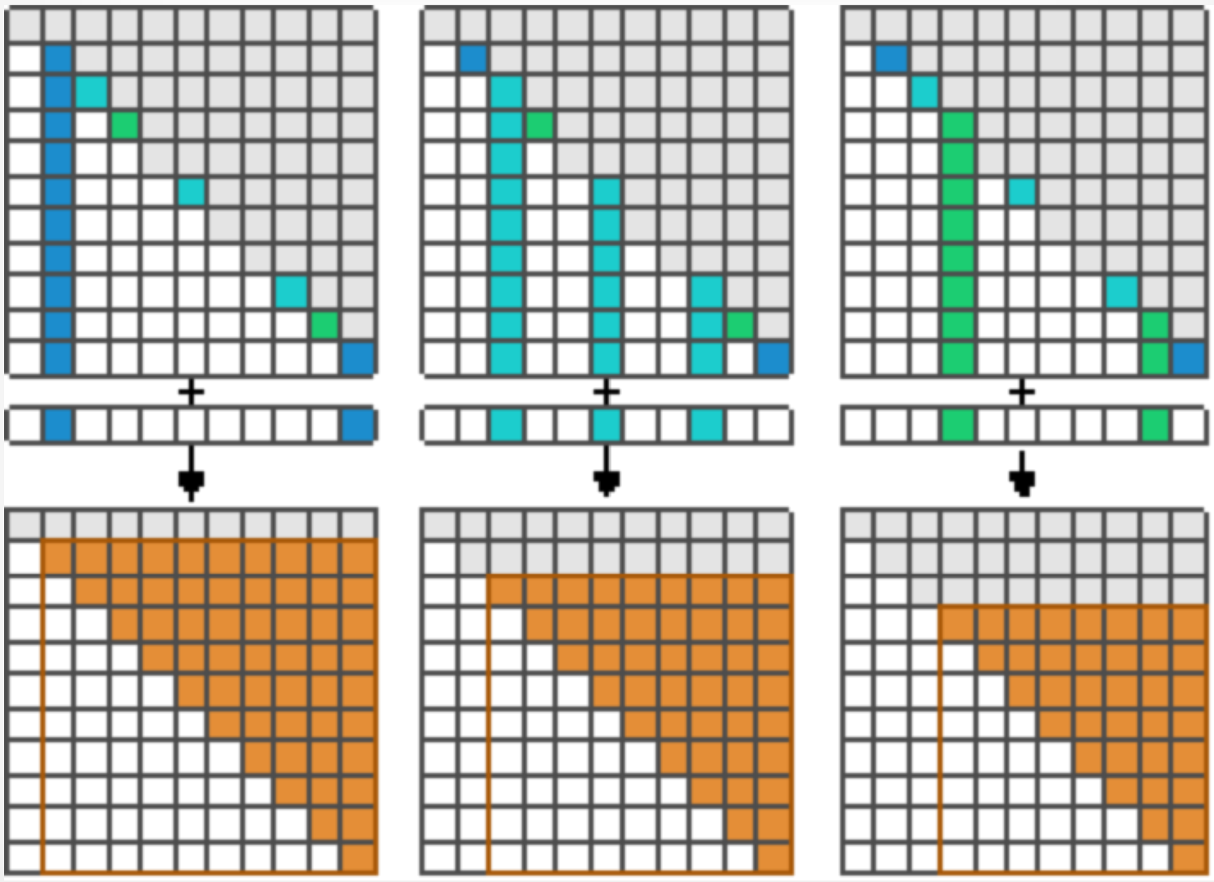
- We often wish to evaluate the impact of actions on the information/uncertainty of the posterior belief
- (Differential) entropy: $\mathrm{H}(X) = -\int \mathbb{P}(x) \ln \mathbb{P}(x)\, dx$
- For a Gaussian belief, yields the objective function:

$$V(b, a) \doteq \ln|\boldsymbol{R}^{+}| - \frac{N}{2} \cdot \ln(2\pi e)$$

- $\boldsymbol{R}^{+}$ is the square root matrix of the posterior belief

Computationally Challenging

Belief update for every candidate, over (possibly) long horizons



Efficient Decision Making under Uncertainty in High-Dimensional State Spaces

Decision Making: Simplified

- Consider a decision problem $\mathcal{P} \doteq (b, \mathcal{A}, V)$
- The concept: Identify and solve an equivalent, yet "easier" decision problem $\mathcal{P}_s \doteq (b_s, \mathcal{A}_s, V_s)$
- In this talk, focus on simplifying the initial belief
- Goal: improving efficiency, maintaining quality
- How should we measure the simplification quality?

Action Consistency

Definition:

The problems are action consistent, if the following applies $\forall a_i, a_j \in \mathcal{A}$:

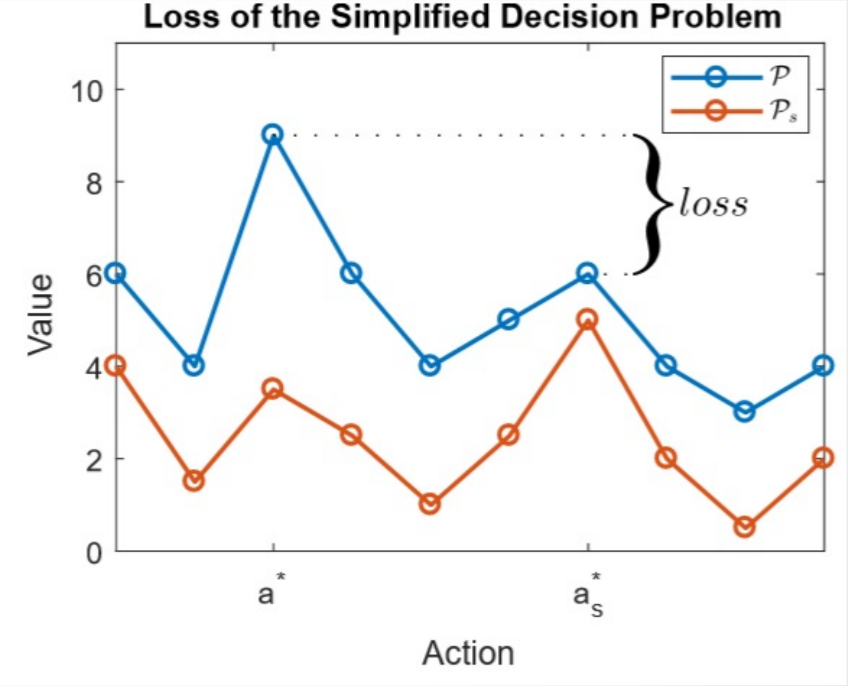
$$V(b, a_i) < V(b, a_j) \Leftrightarrow V(b_s, a_i) < V(b_s, a_j)$$

- In decision making, we only care to rank the actions
- Action selection is not affected by the actual objective values

Simplification Loss

<u>Definition:</u>

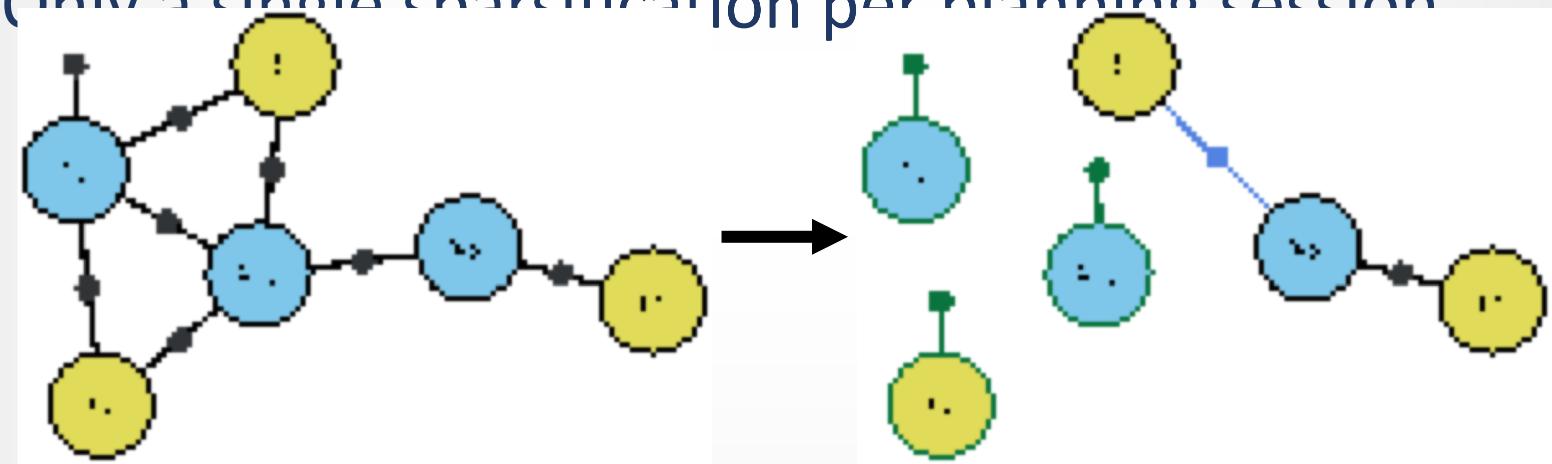
$$loss(\mathcal{P}, \mathcal{P}_s) \doteq V(b, a^*) - V(b, a_s^*)$$



- Simplification may lead to a sub-optimal action
- Measure for the quality-of-simplified-solution
- Further work on how to derive loss bounds for simplification methods is not discussed here…

Belief Sparsification

- Plan with a sparse approximation of the initial belief
- Reduce the number of factors and disconnect variables
- Only a single sparsification per planning session



[1] Consistent Sparsification for Efficient Decision Making Under Uncertainty in High Dimensional State Spaces, K. Elimelech and V. Indelman, ICRA ‘17

[2] Scalable Sparsification for Efficient Decision Making Under Uncertainty in High Dimensional State Spaces, K. Elimelech and V. Indelman, IROS ’17

[3] Fast Action Elimination for Efficient Decision Making and Belief Space Planning Using Bounded Approximations, K. Elimelech and V. Indelman, ISRR ‘17

[4] Simplified Decision Making in the Belief Space using Belief Sparsification, K. Elimelech and V. Indelman, IJRR ‘18 (conditionally accepted)

Belief Sparsification

- In practice – (square root) matrix sparsification

Algorithm 1: Scalable Belief Sparsification

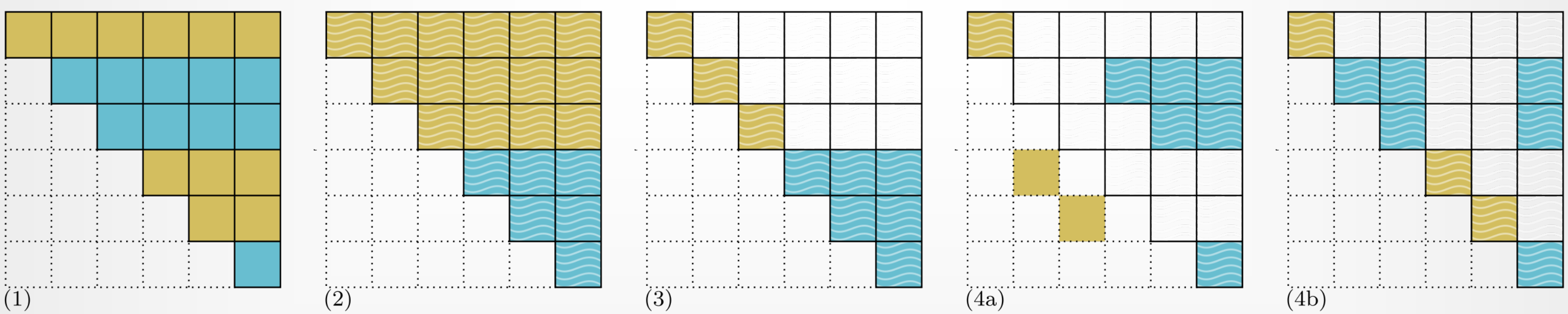
Inputs:

A belief $b = \mathcal{N}(\boldsymbol{X}, \boldsymbol{\Lambda}^{-1})$, s.t. $\boldsymbol{\Lambda} = \boldsymbol{R}^T \boldsymbol{R}$

A subset $\mathcal{S}$ of state variables to sparsify

Output:

A sparsified belief b_s

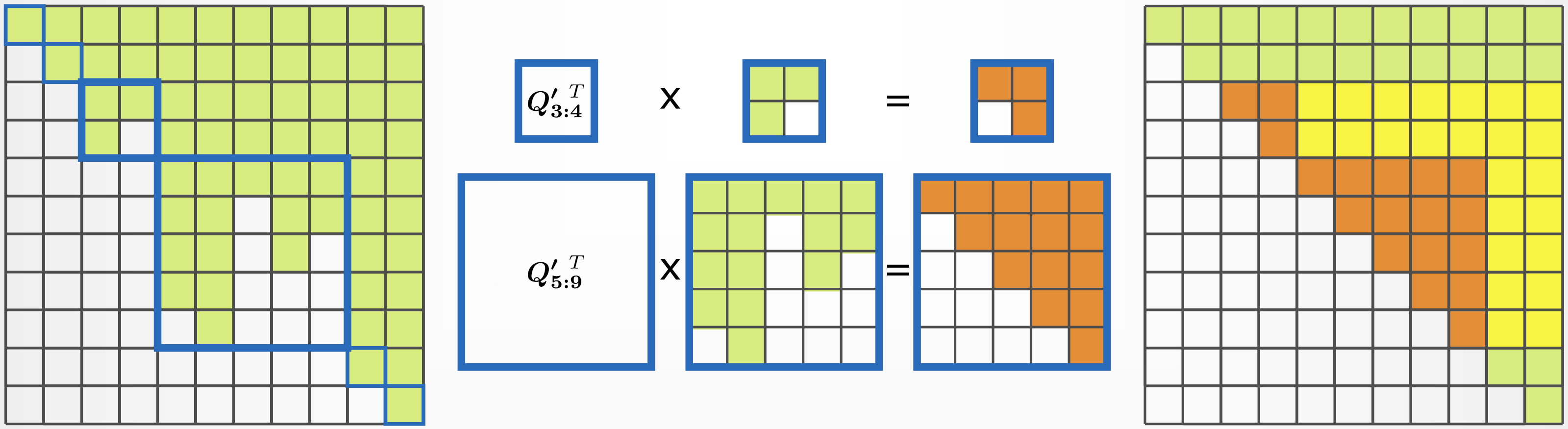


1. Separate variables
2. Remove entries
3. Reorder back

10 **return** $b_s \doteq \mathcal{N}(\boldsymbol{X}, \boldsymbol{\Lambda}_s^{-1})$, s.t. $\boldsymbol{\Lambda}_s \doteq \boldsymbol{R}_s^T \boldsymbol{R}_s$

Efficient Reoredering

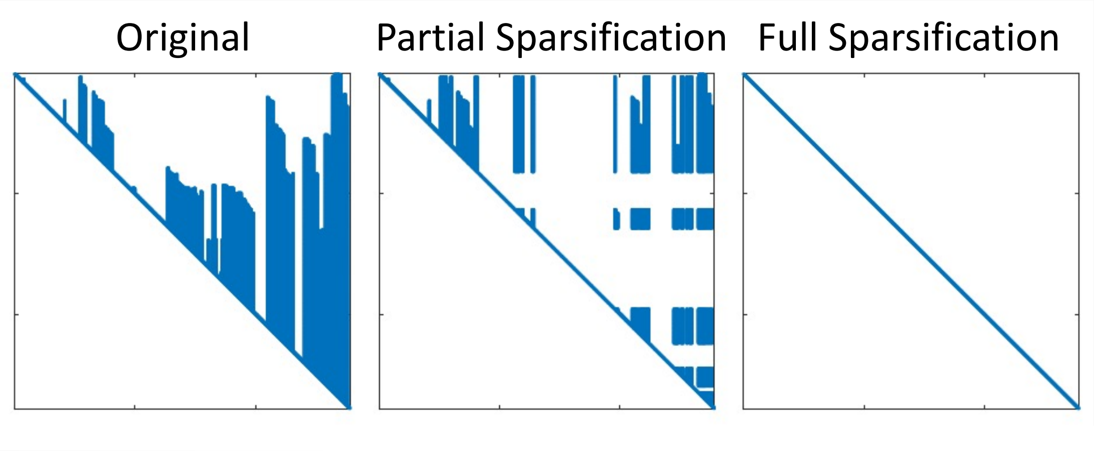
- Reordering the variables (=columns) would break R's shape
- Thus, variable reordering typically requires re-factorization
- We showed that we can simply apply (in parallel!) "local" modifications to the matrix, with minimal to no re-factorization



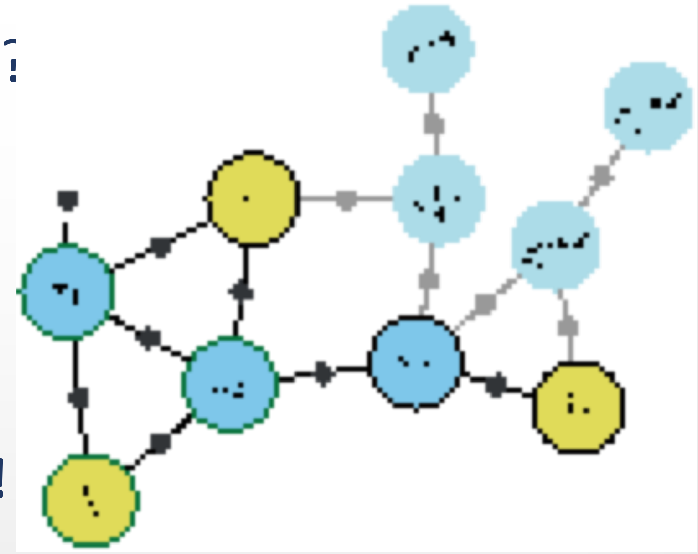
[1] Efficient Modification of the Upper Triangular Square Root Matrix on Variable Reordering, K. Elimelech and V. Indelman, RA-L '21

Belief Sparsification

Sparsification is scalable:

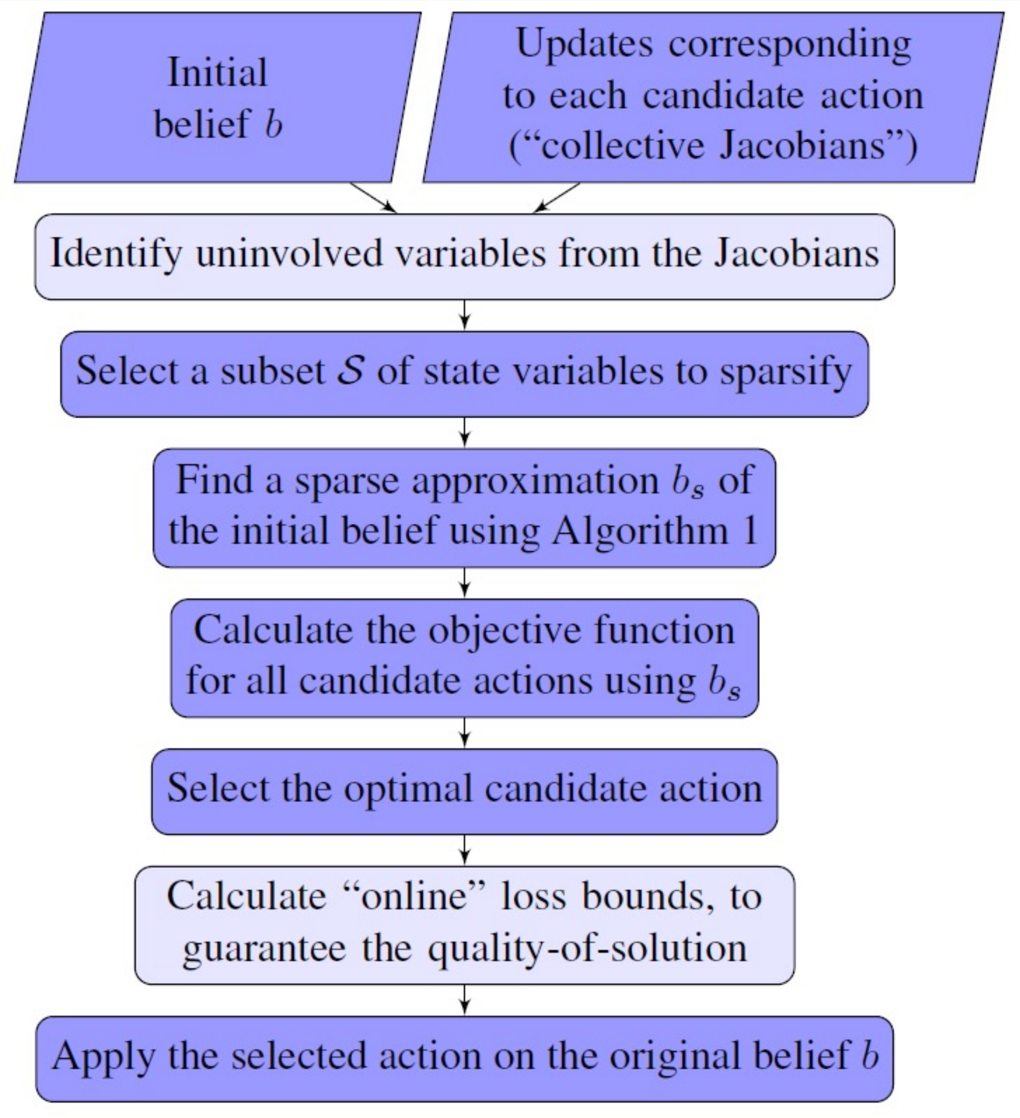


But which variables should we sparsify?



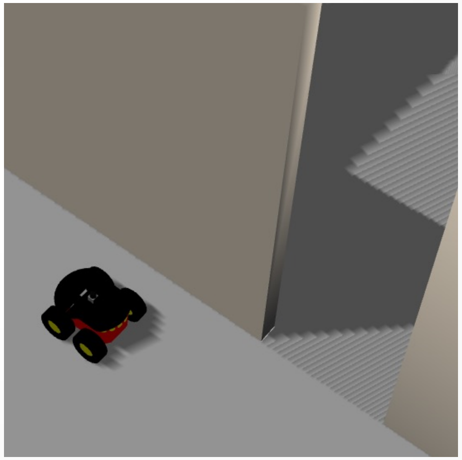
Sparsification of uninvolved variables does not compromise action consistency!

Approach Summary



Experimental Results

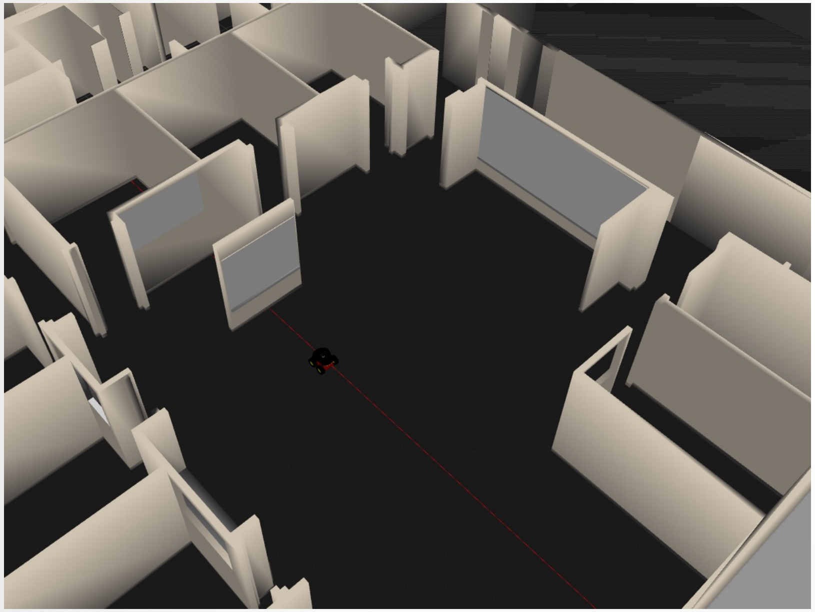
- A highly realistic active-SLAM problem
- The robot should navigate through a list of goals in an unknown indoor environment.

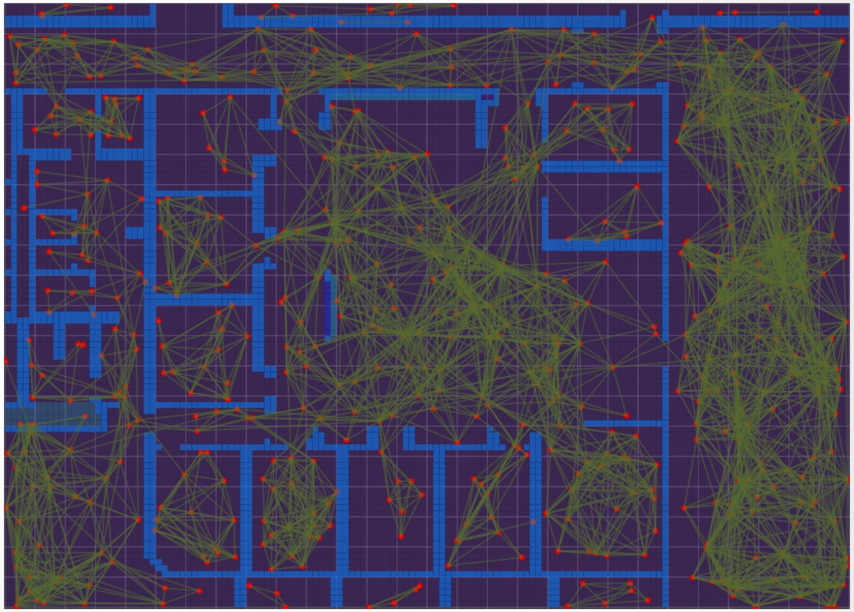


- Pioneer 3-AT robot, with a lidar sensor, Hokuyo UST-10LX – real and in simulation.

Experimental Results

- The environment and the PRM graph, used for generation of candidate trajectories.





- Every square stands for 1m x 1m in reality.

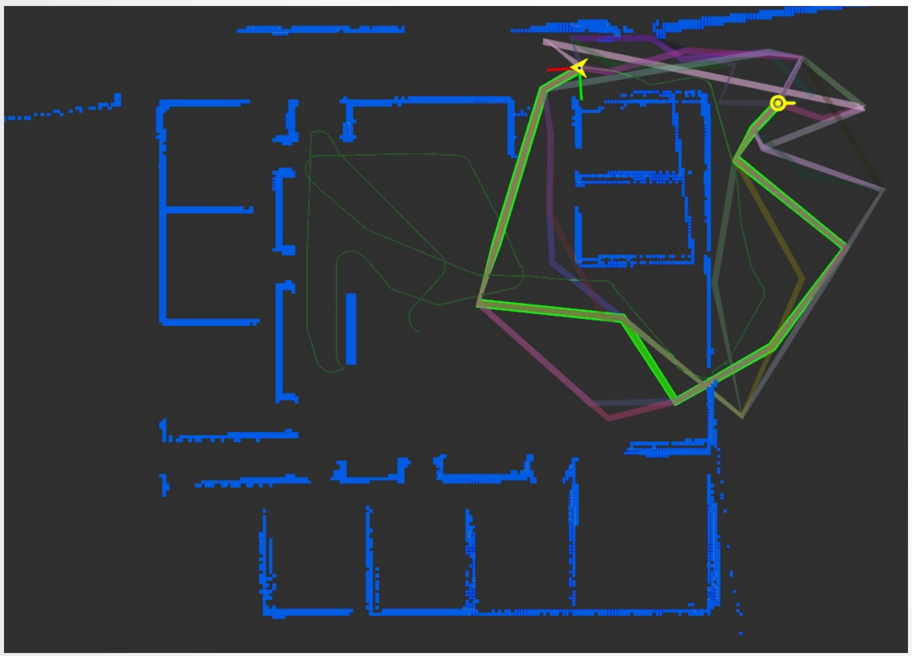
Experimental Results

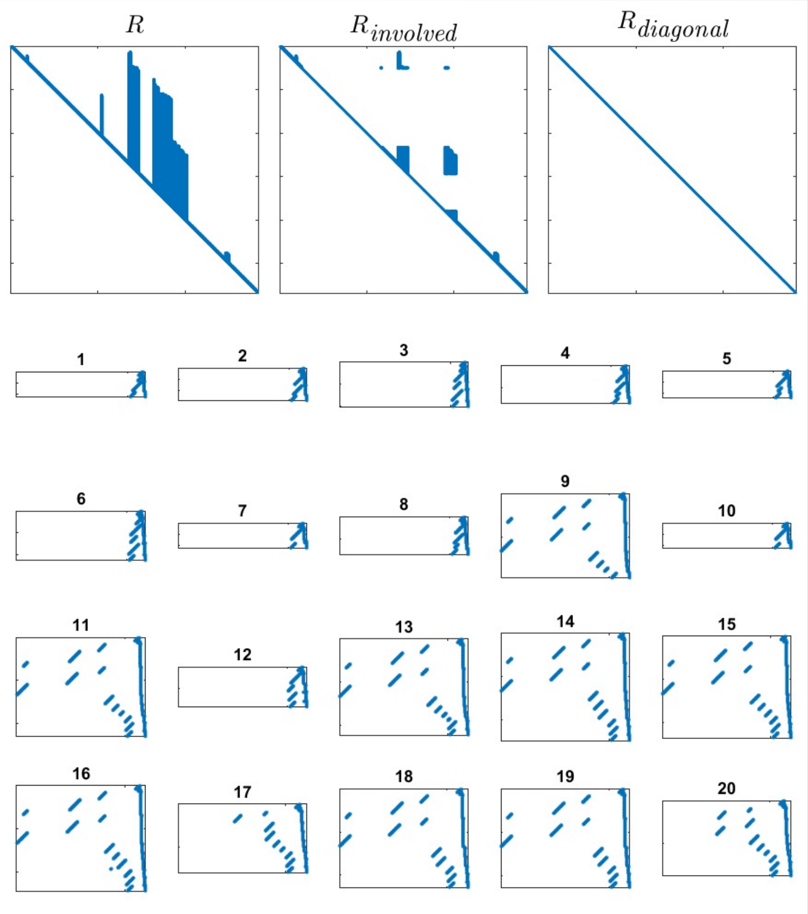
- Pose SLAM
- Examining 20 different trajectories to the next goal
- Objective: minimal final uncertainty (entropy)
- Loop closures via point cloud matching (ICP)

- For comparison, for each goal we solved three versions of the decision problem:
  - $\mathcal{P}$ – using the original belief
  - $\mathcal{P}_{involved}$ – with sparsification of the uninvolved variables
  - $\mathcal{P}_{diagonal}$ – full sparsification

Experimental Results

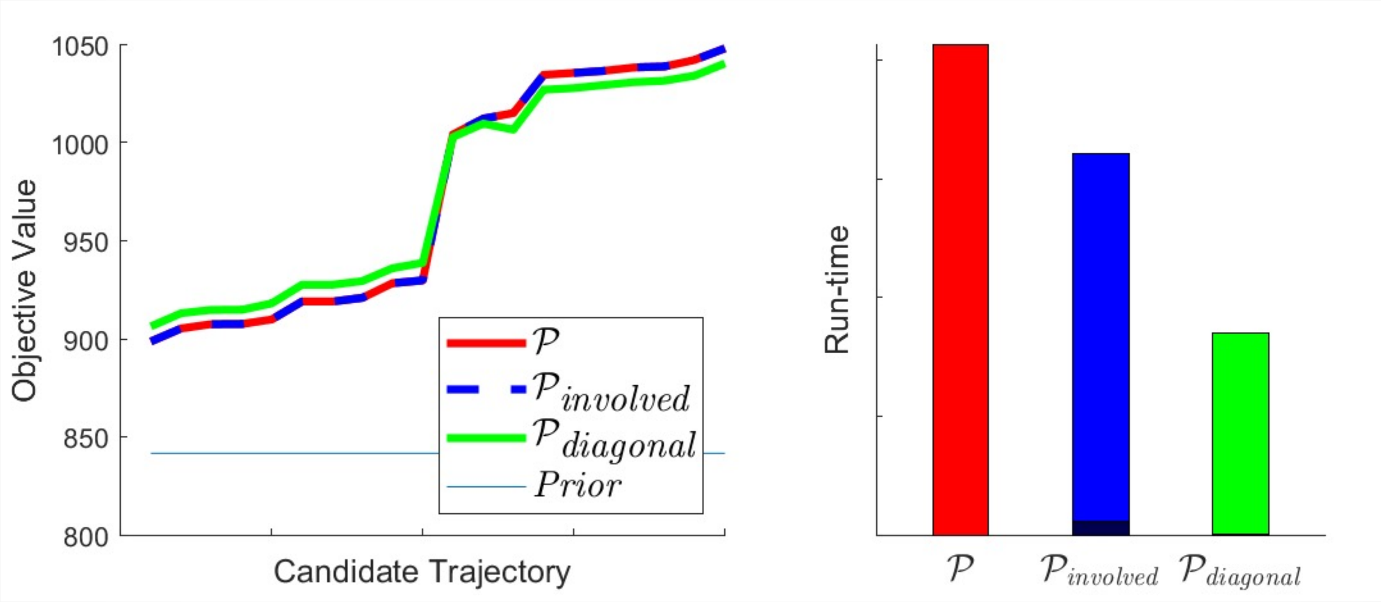
- Scenario, three versions of the initial belief, and the considered updates of the candidate trajectories



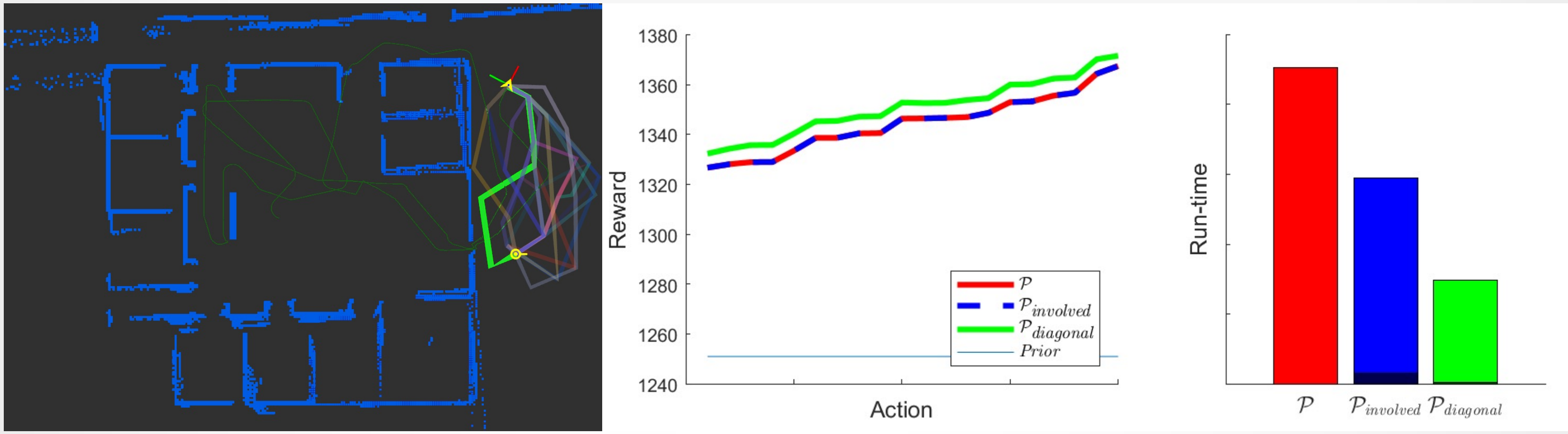


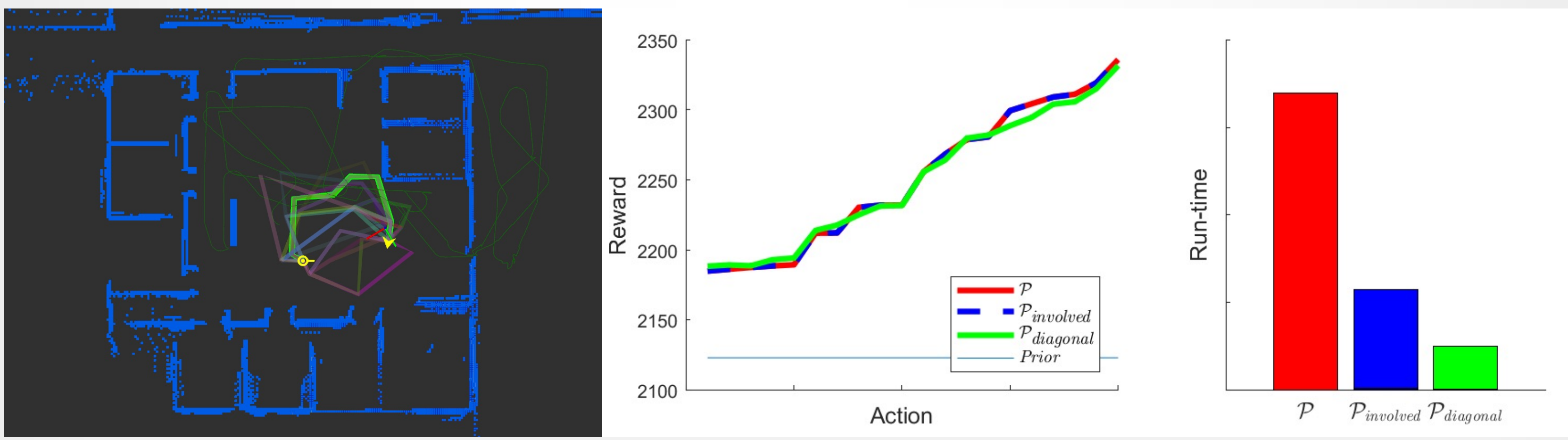
Experimental Results

Comparison: objective trend (quality-of-solution), and run-time.

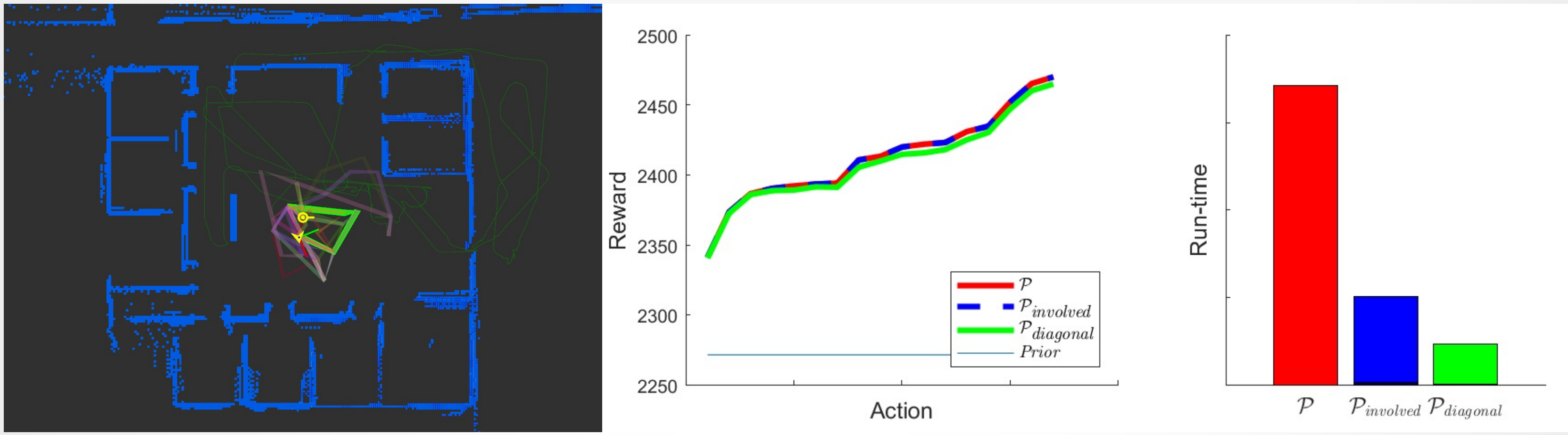


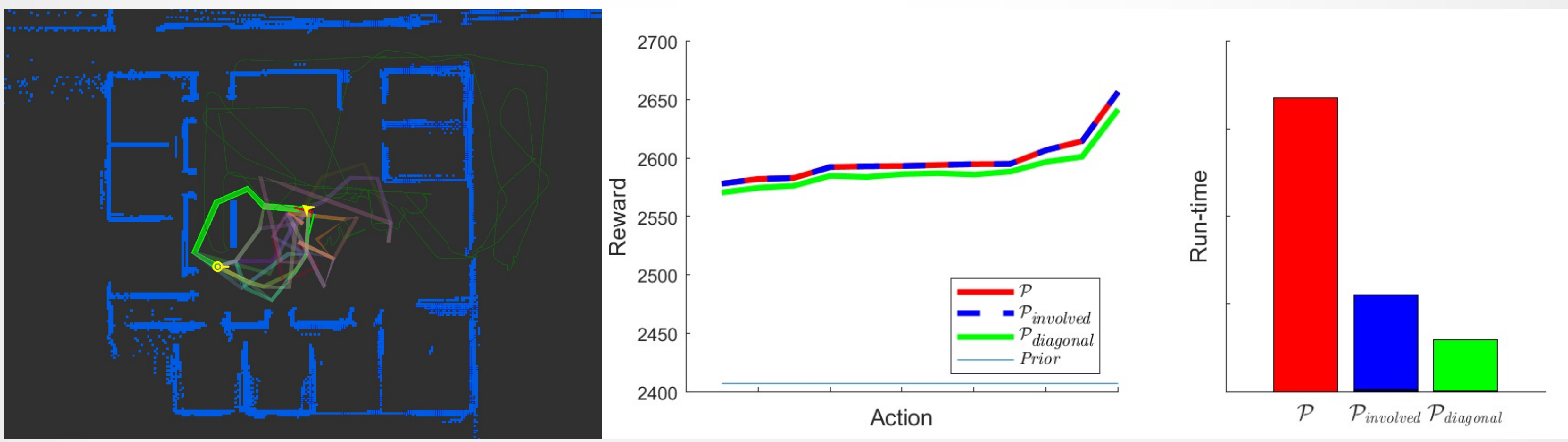
More Results





More Results





Belief Sparsification: What's Next?

- Evidently, very effective in reducing the planning cost!
- We calculate the sparsification "from scratch" at every planning session
- Can we make it incremental?
- Can we make it even more efficient?
- These questions led us to another "simplification":

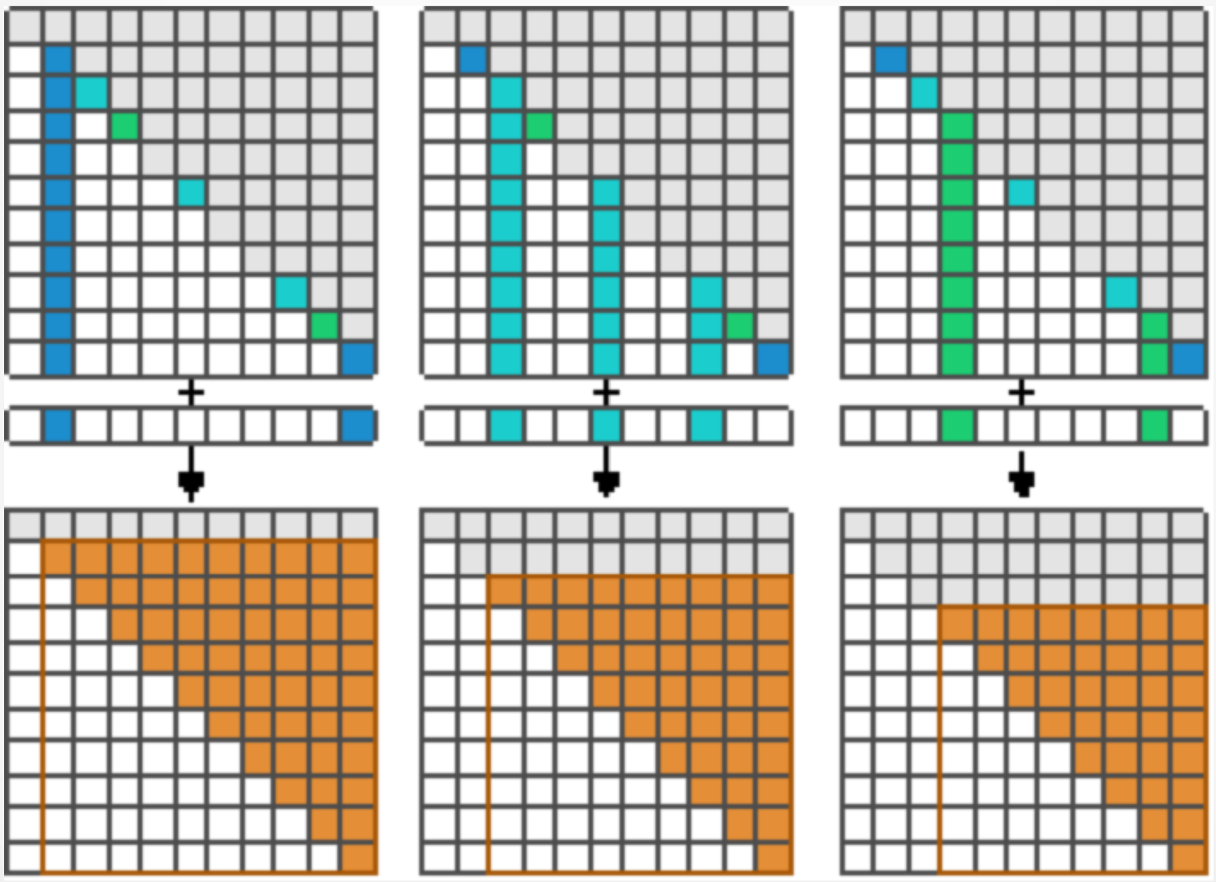
PIVOT: Predictive Incremental Variable Ordering Tactic

[1] Introducing PIVOT: Predictive Incremental Variable Ordering Tactic for Efficient Belief Space Planning, K. Elimelech and V. Indelman, ISRR '19

[2] Efficient Belief Space Planning in High-Dimensional State Spaces using PIVOT: Predictive Incremental Variable Ordering Tactic, K. Elimelech and V. Indelman, IJRR '21 (invited)

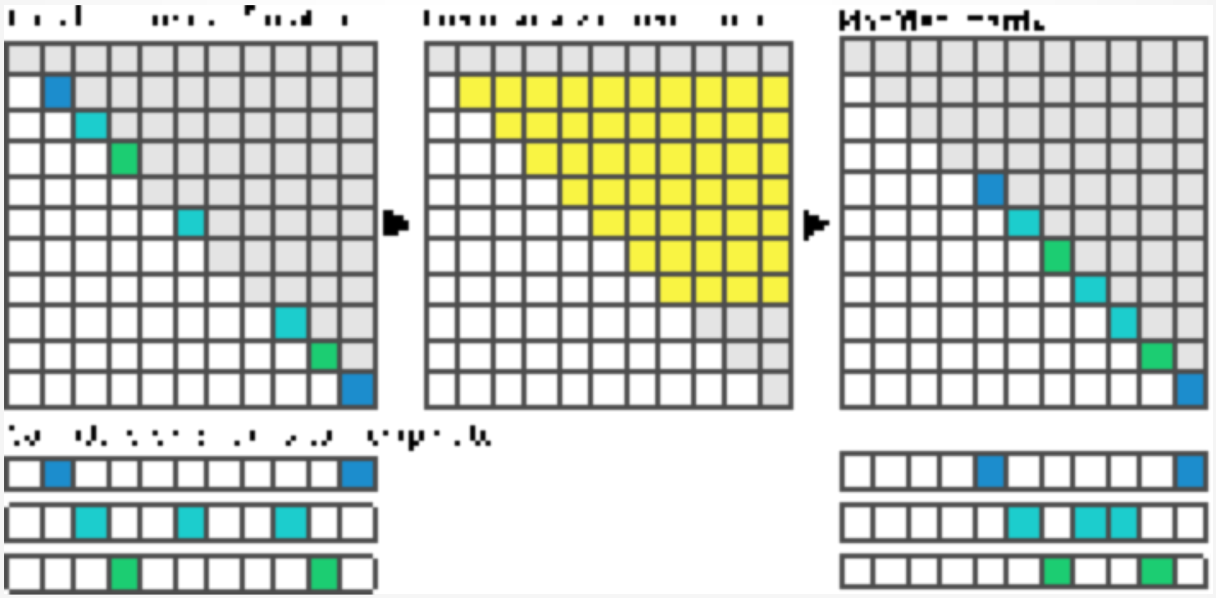
PIVOT: Basic Concept

- Like in sparsification: identify involved variables



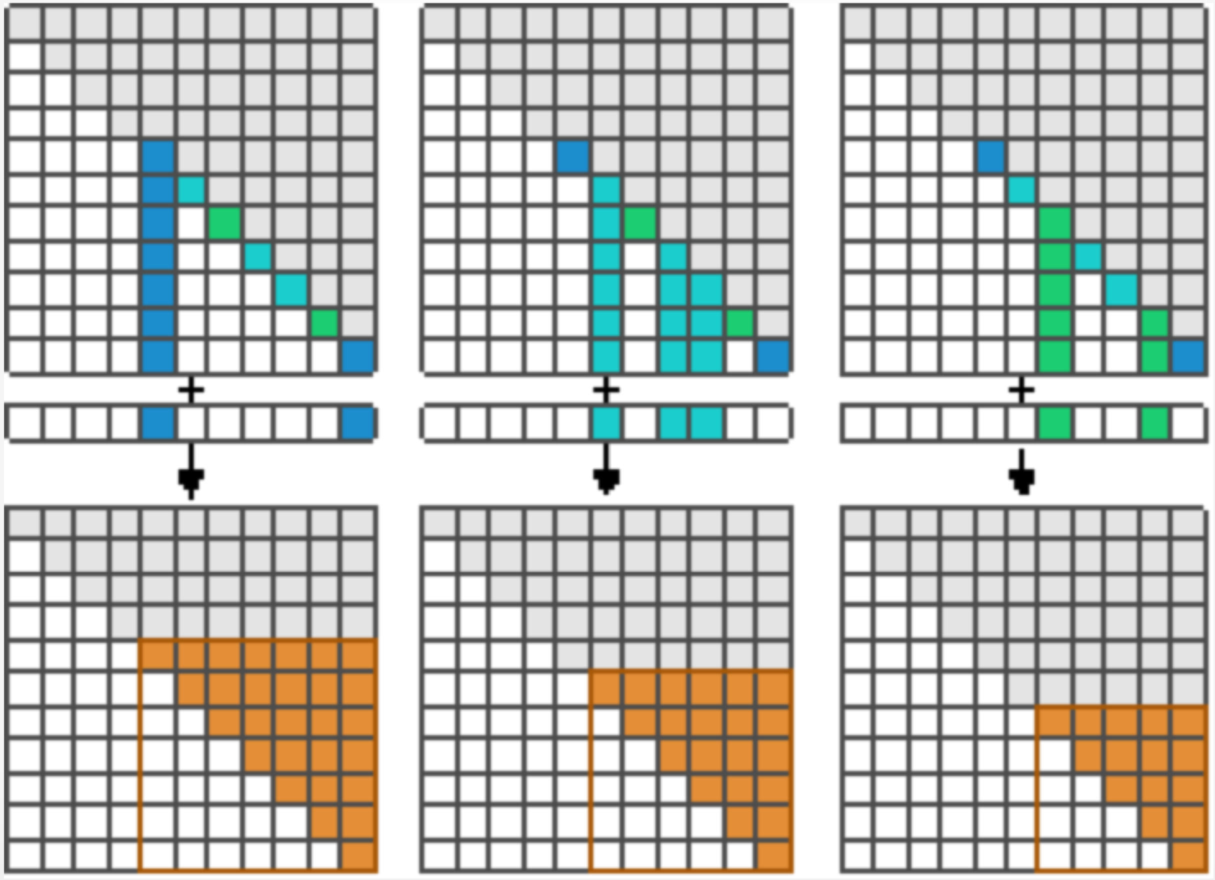
PIVOT: Basic Concept

- Variable reordering: simply push involved variables forwards -- no sparsification, no “reordering back”



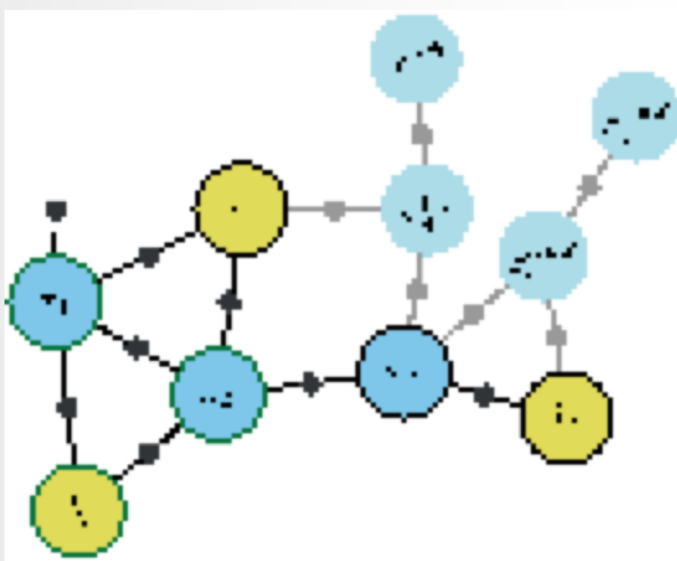
PIVOT: Basic Concept

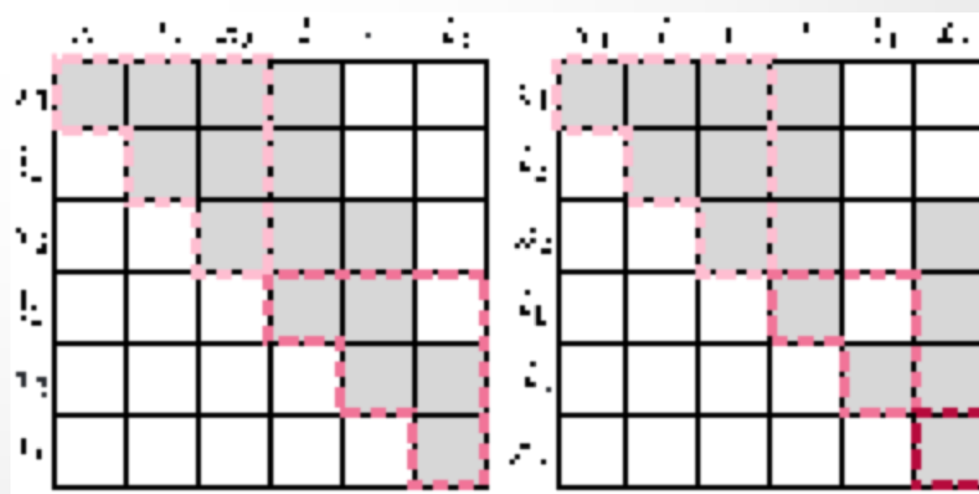
- Efficient updates: reducing the size of the affected blocks!



PIVOT: Optimized

- Basic PIVOT is comparable to uninvolved sparsification
- However, with PIVOT*, we suggest further order optimizations:
- First, we divide the variables to multiple classes:
  - “More involved” are pushed “more forwards”



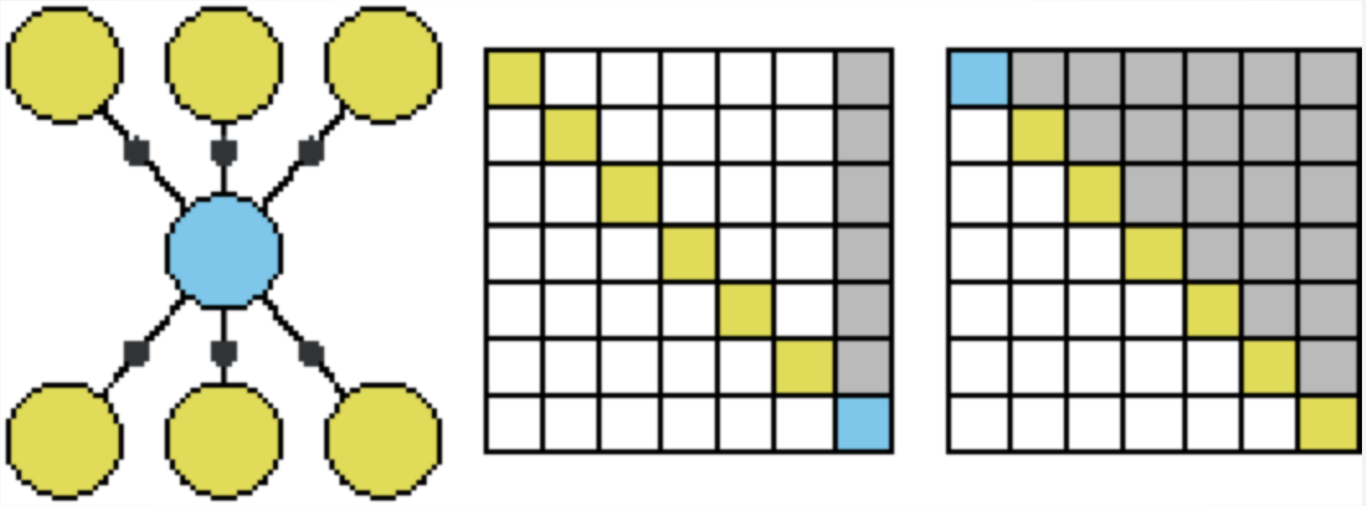


2 variable classes

3 variable classes

PIVOT: Optimized

- Further, PIVOT* is aware of fill-in (density), and takes into consideration variable connectivity (node degree)

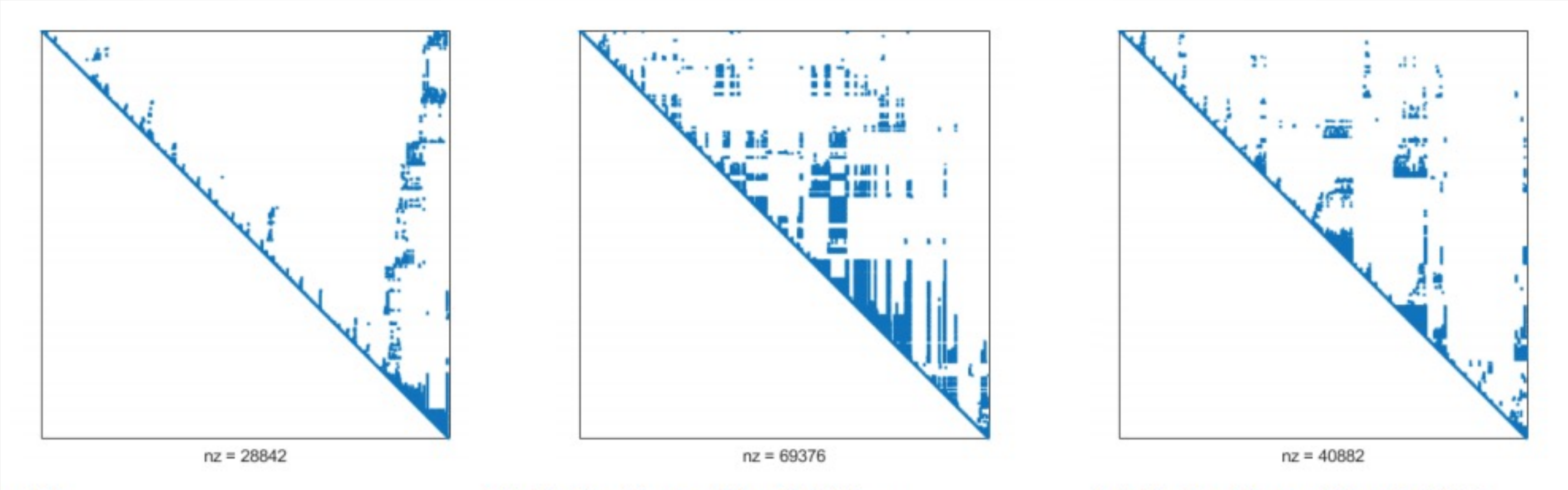


For the full algorithm see:

[1] Efficient Belief Space Planning in High-Dimensional State Spaces using PIVOT: Predictive Incremental Variable Ordering Tactic, K. Elimelech and V. Indelman, IJRR '21 (invited)

PIVOT Applied to the Prev. Scenario

- We solved the previous simulated experiment with variations of PIVOT.
- Reordering the variables before each planning session (instead of spars.)

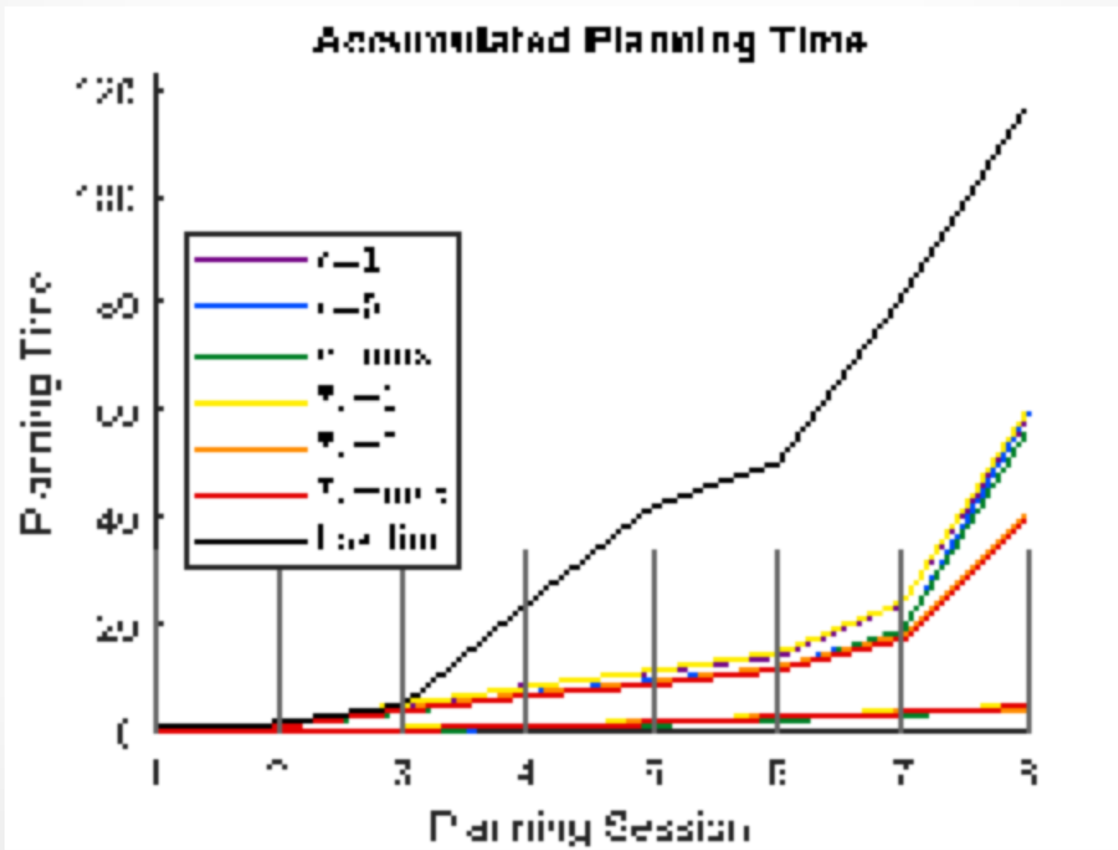


R at the final planning session | after applying PIVOT | after applying PIVOT*

- No approximation! Different representations for the same belief

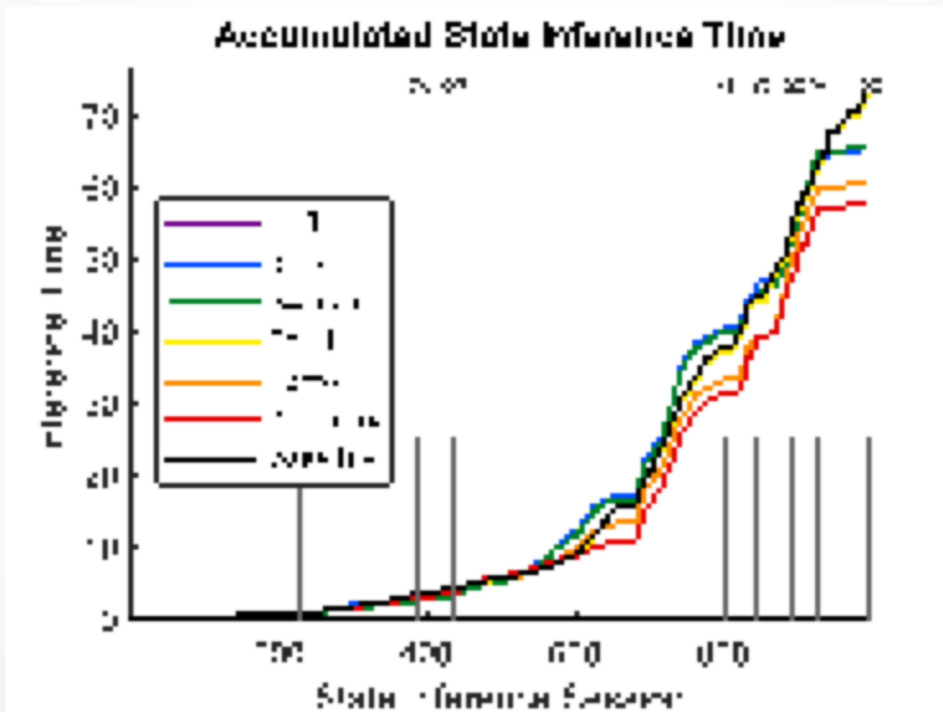
PIVOT: Results

- We solved the previous simulated experiment with variations of PIVOT.
- Reordering the variables before each planning session (instead of spars.)



PIVOT: More Benefits

- If we maintain the modified order after planning:
  - Order can be updated incrementally on re-planning!
  - PIVOT can also improve the efficiency of state inference!



Sparsification vs. PIVOT

- PIVOT = change of representation
- Sparsification = approximation

- Efficiency:

"full" spars. > PIVOT* > PIVOT = "uninvolved" spars. > original

- Quality-of-solution:

original = PIVOT(*) = "uninvolved" spars. > "full" spars.

Summary

We covered:

- SLAM and estimation under uncertainty
- Beliefs over high-dimensional states
- Planning in the belief space
- Our contributions:
    1. Theoretical framework for simplified decision making
    2. Sparsification for efficient planning
    3. Variable reordering for efficient planning (and inference)
    4. Efficient variable reordering algorithm

Thank you!

www.khen.io