Robot Tracking with Relative Distances

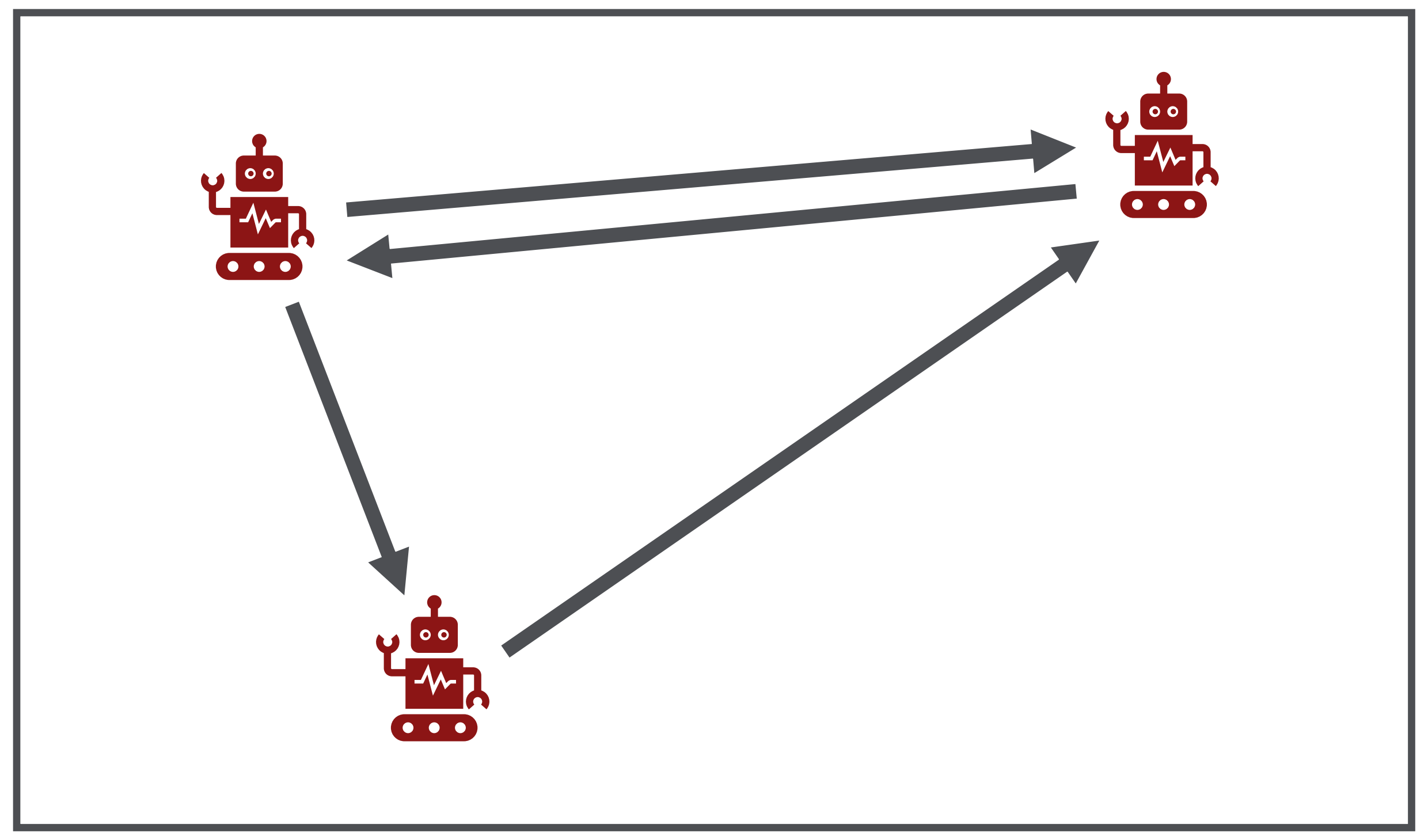
Erik Helmer

EE 364b Project, Stanford University

Stanford
Electrical Engineering

Problem Description

Imagine a group of robots moving in an open environment. Typically, they are tracked using many on-board sensors, measuring the robots themselves, as well as their surroundings. Is all this really necessary? With just how few measurements can the robots be tracked accurately? Is it possible to determine the robot trajectories knowing only pairwise distances?



Robot Model

The robot states $\vec{x}$ are modelled as classic unicycle robots with controls v_t & ω_t:

$$\begin{aligned} x_{t+1} &= x_t + v_t \cos\theta_t \\ y_{t+1} &= y_t + v_t \sin\theta_t \\ \theta_{t+1} &= \theta_t + \omega_t \end{aligned}$$

Measurements

To simplify, the correspondences are assumed to be known. However, the measurements are assumed to be noisy and incomplete:

$$\vec{y}_t^{(i,j)} = \left\|\vec{x}_t^i - \vec{x}_t^j\right\|_2 + \delta_t$$

As an Optimization Problem

As is often the case, the positions are estimated by minimizing square loss:

$$\min_{\vec{x}^1,\ldots,\vec{x}^N} \sum_{(i,j)\in\mathcal{E}} \left(\left\|\vec{x}^i - \vec{x}^j\right\|_2 - \vec{y}^{(i,j)}\right)^2$$

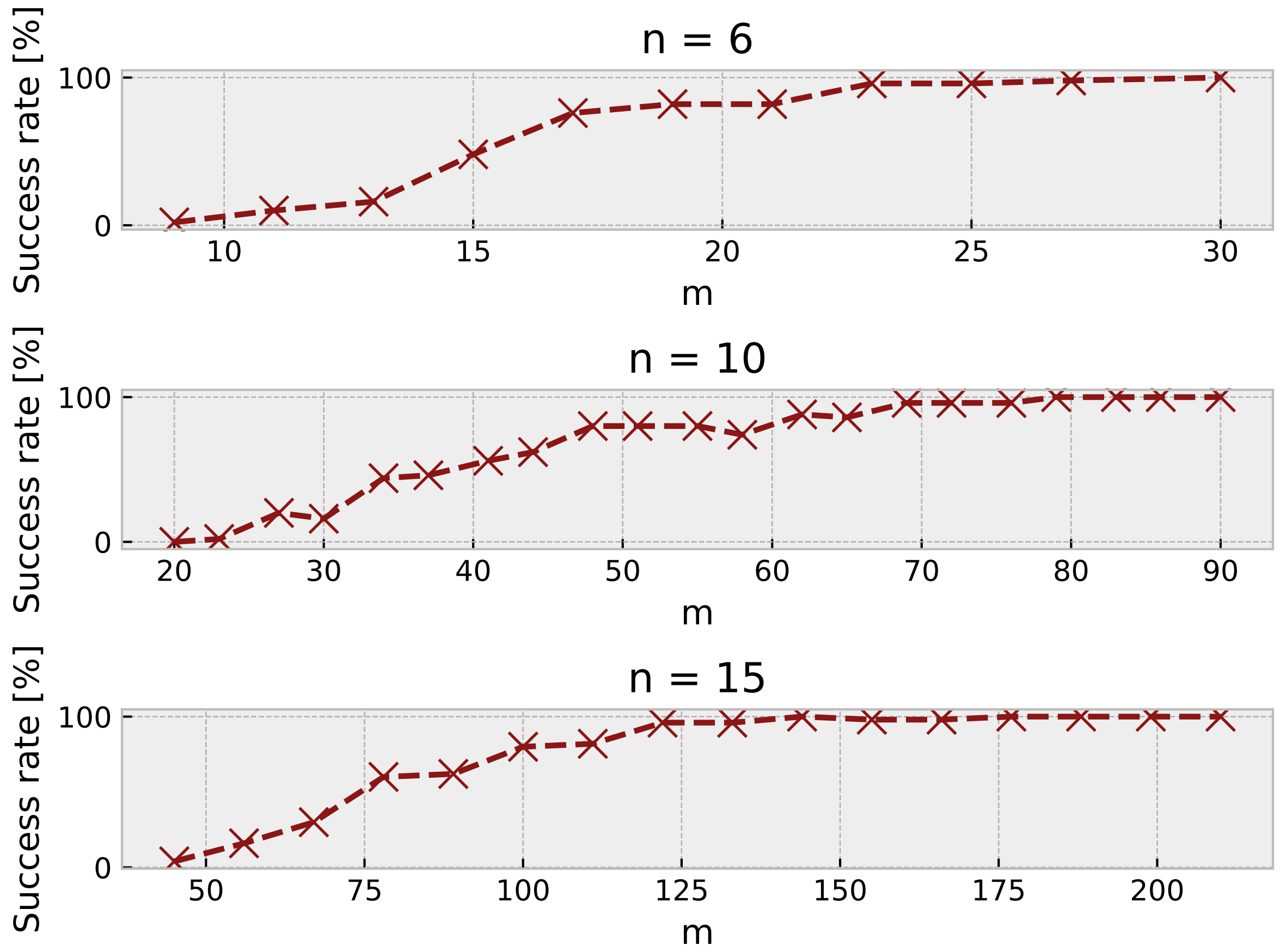
But this is non-convex and non-differentiable! How do we address this?

The Riemannian Elevator

Instead optimize a high-dimensional convex relaxation, then project to lower dimension.

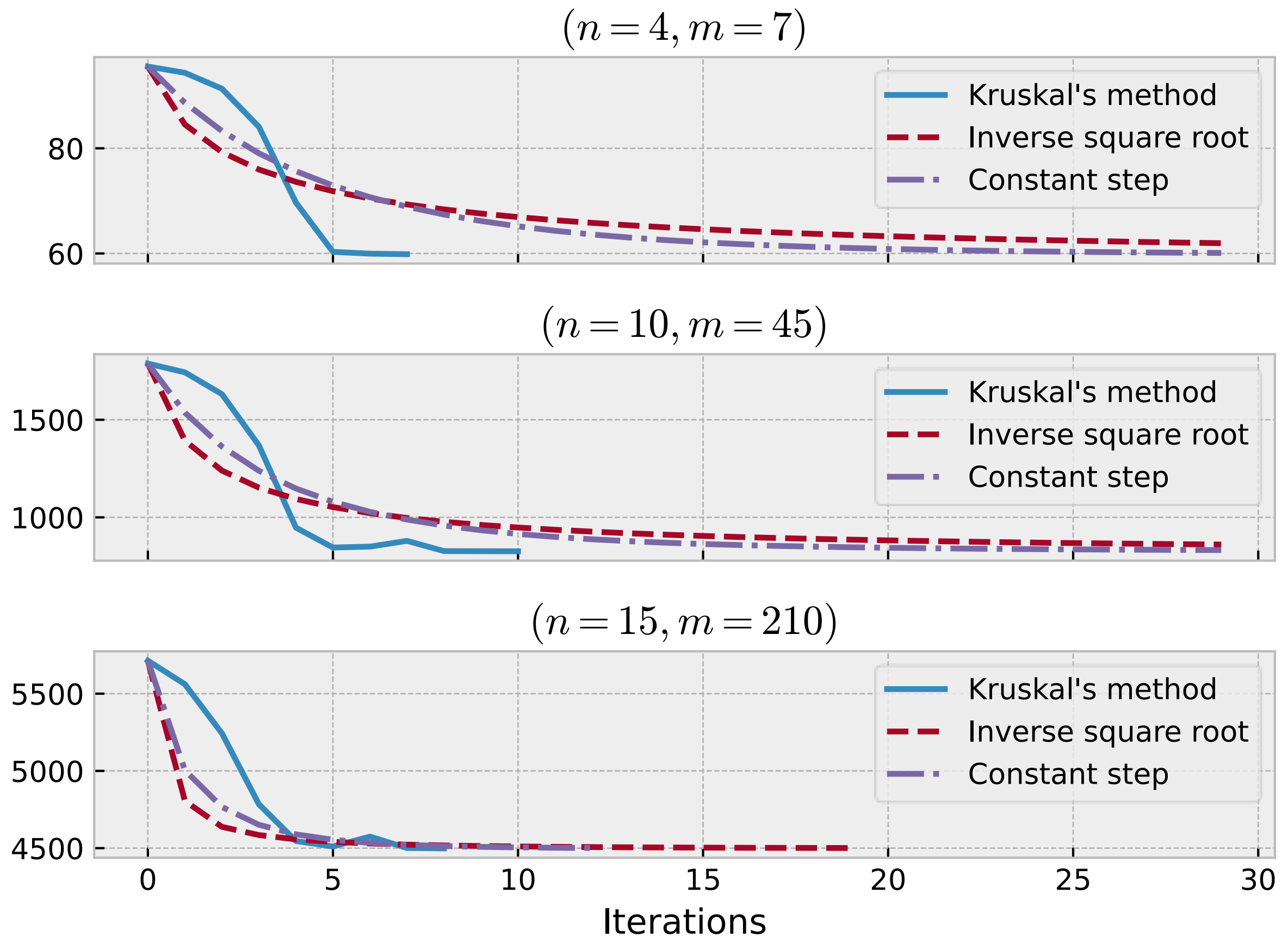
This projection is often accurate enough that that a first-order method then converges to the global minimum.

In the plots below, we see the at which rate the Riemannian Elevator generates an initial guess that converges to the global optimum with first-order methods.



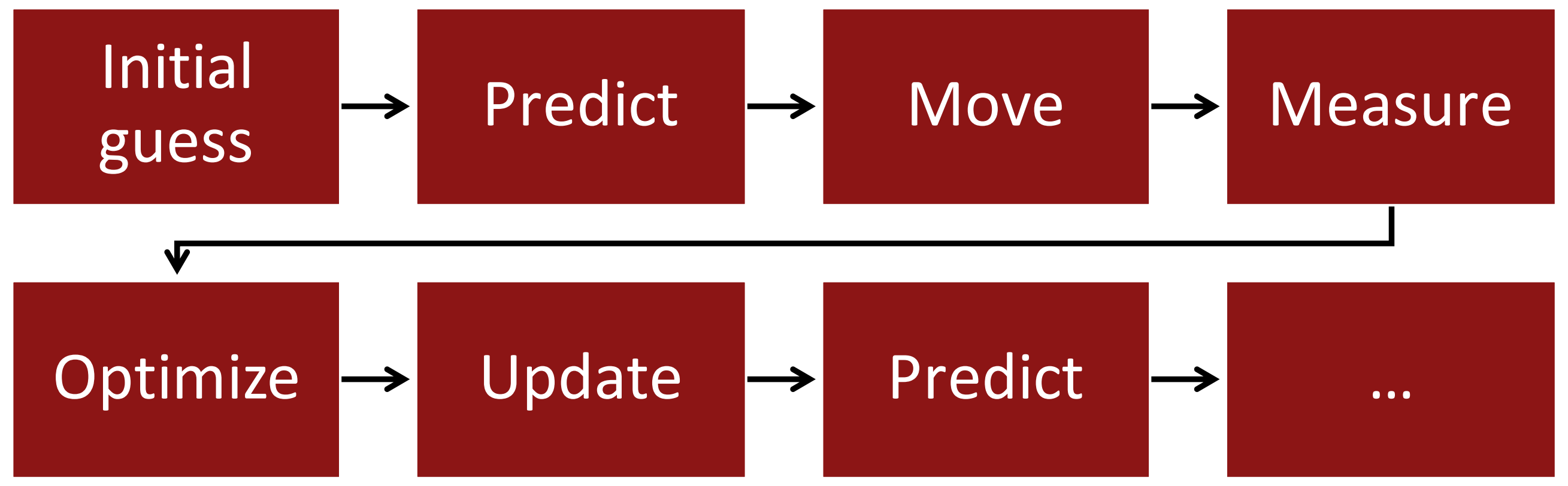
Kruskal's Method

We use gradient descent, with step sizes chosen by Kruskal's method, to refine the estimates. Below is a plot of the convergence rate when tracking of Kruskal's method:



Extended Kalman Filter

To improve tracking performance, we add an EKF. The complete tracking pipeline is:

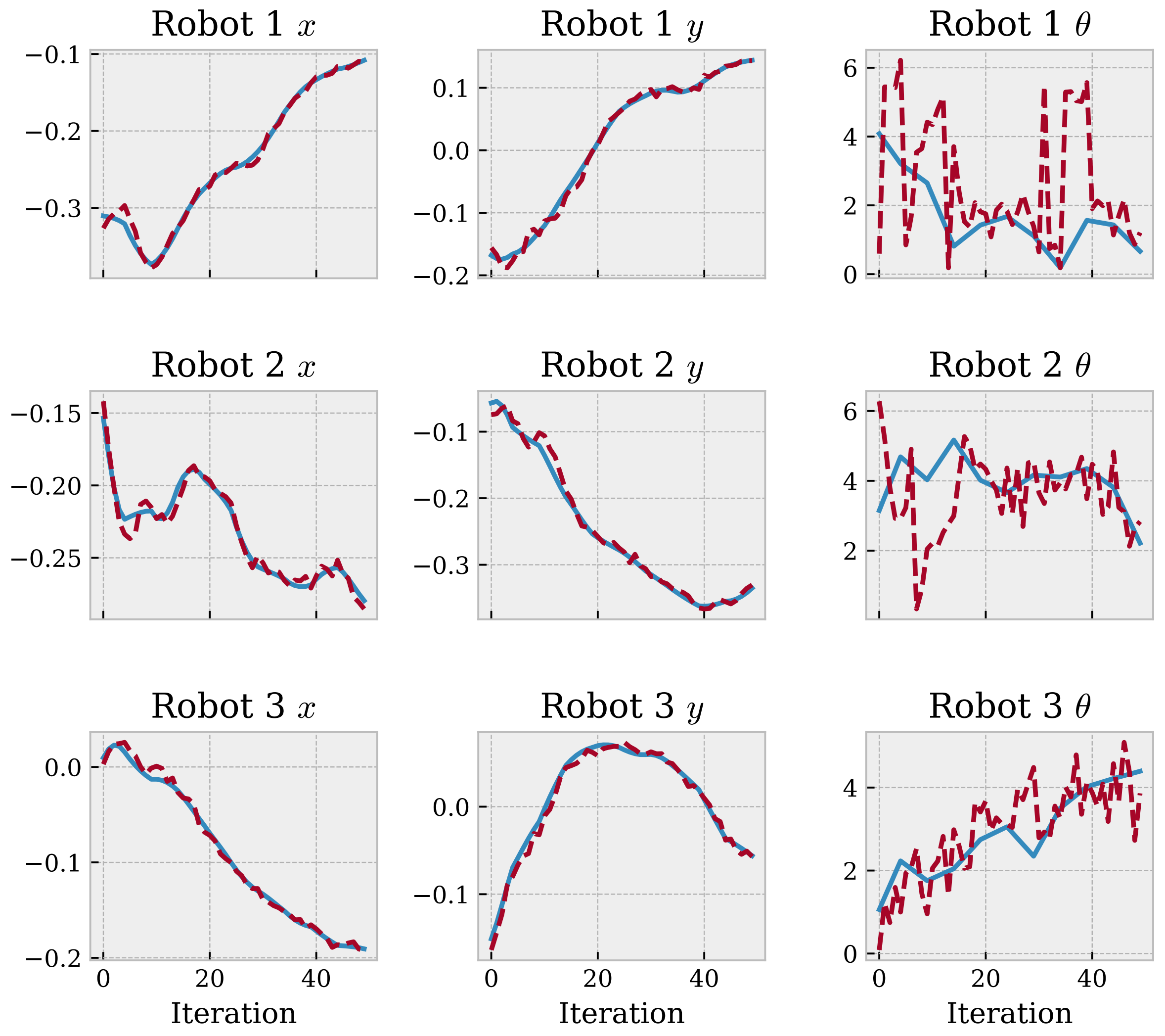


Initial guess → Predict → Move → Measure

Optimize → Update → Predict → ...

Tracking Performance

In the plots below, the tracking error of the pipeline defined above is shown.



Acknowledgements

Thank you to the EE364b course staff for a great course!