Technion – Israel Institute of Technology



HW5

Autonomous Navigation and Perception

086762

|  |  |  |
| --- | --- | --- |
| Alon Spinner | 305184335 | alonspinner@gmail.com |
| Dan Hazzan | 308553601 | danhazzan@campus.technion.ac.il |

May 21, 2022

# Theoretical Questions – Q1

MCTS is a search method based on sampling states and actions for simulations to terminal states. It works well when we can represent the search space well enough with a small, discretized set where the combination numbers of states and actions are finite. When the search space cannot be granulated efficiently, as in continuous state spaces, we cannot guarantee that the sampling combinations (states and actions) will ever lead to a proper solution (“win”) when running the simulations to the terminal state – At least not in finite time.

# Hands-on – Q1+2

We were asked to simulate a robot autonomously navigating using the following framework at each time step:

Diagram

Description automatically generated

Figure : framework for each timestep

Planning was done with a *modified Sparse Sampling Algorithm* where sigma points are chosen and weighed from an observation distribution generated using a propagated belief instead of averaging over random samples.  
for each sigma point ( we compute the belief transition via Kalman and the partial cost .

The code can be viewed in *02\_plan.jl.*Our implementation for choosing the points and weights follows *[Kalman and Bayesian Filters in Python, Roger R Labbe Jr, 2020]* and can be viewed in function.

Chart, bubble chart

Description automatically generated

Figure Weighted Sigma Points of a 2D gaussian distribution

We ran the simulation with time steps, and a rather low horizon of to allow a reasonable computation time.

Two Scenarios were tested:

1. The goal point lies outisde the beacon grid
2. The goal point lies inside the beacon grid

In both simulations we tested for three cases – three values:

1. The robot goes straight towards the goal ignoring the beacons
2. The robot attempts to each the goal with low uncertainty.
3. The robot won’t step away from the beacon as to ensure low uncertainty.

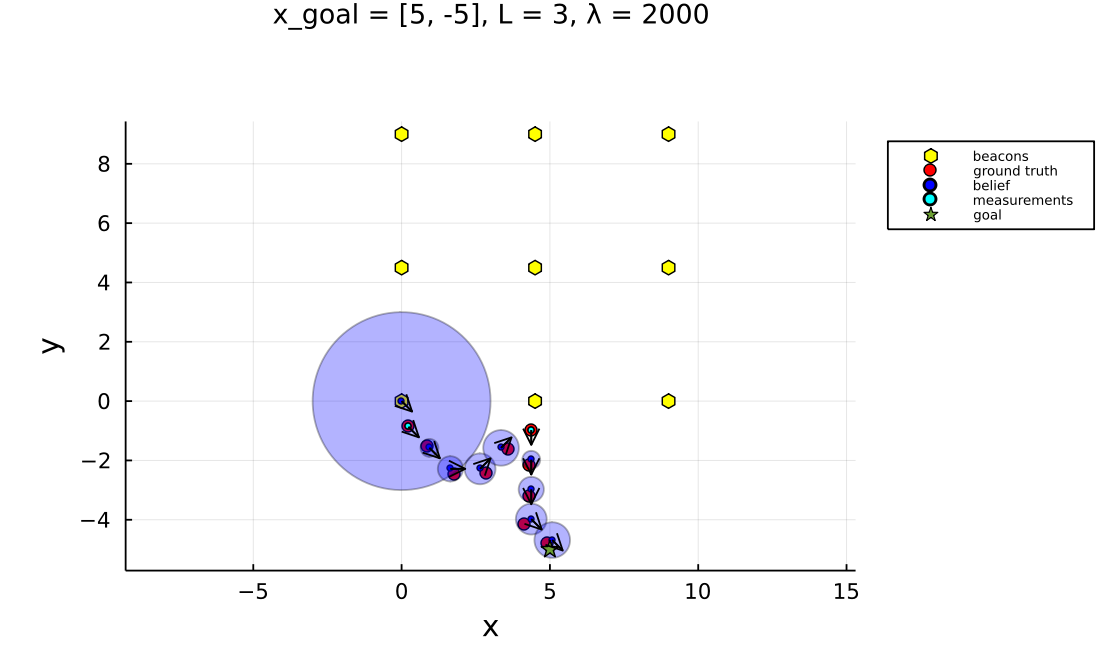
Given the low horizon , we found that works best.  
Many things can be said about the chosen cost function, but we would argue, without showing here, that it is too sensitive to the initial distance to the goal, having us tune per scenario. We would suggest using functions in the cost function to allow for a better ‘local’ or ‘immediate’ decision making.

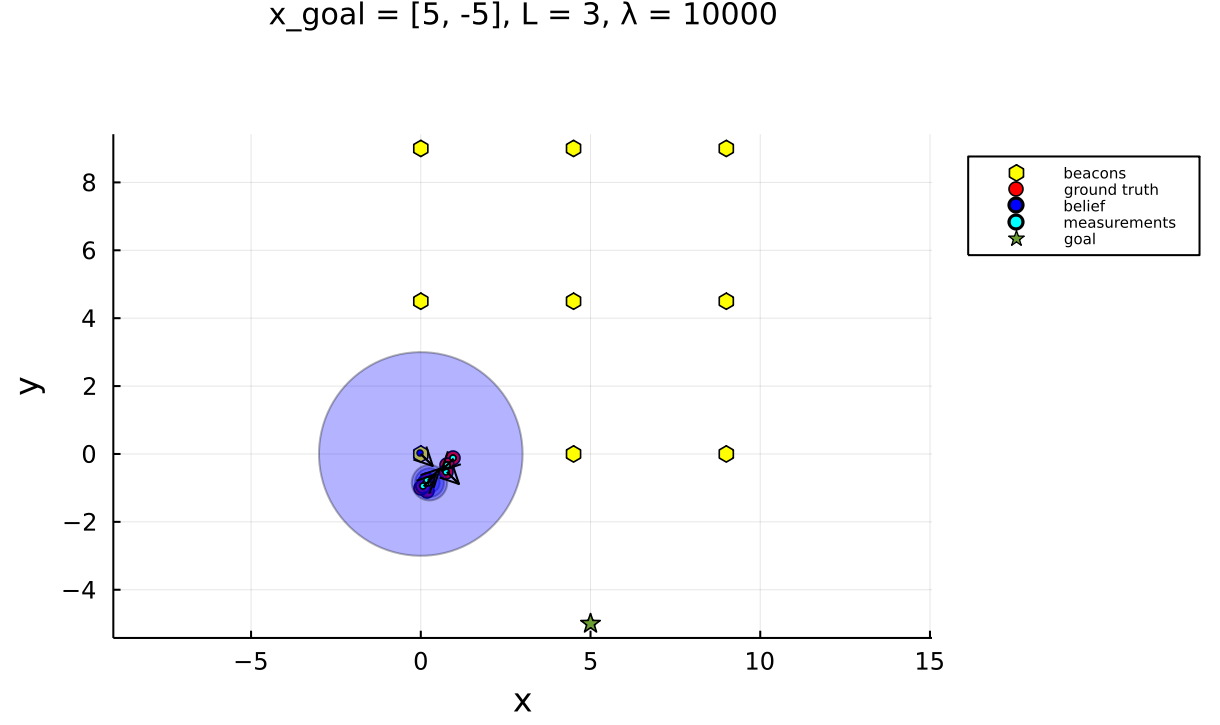
We provide the figures below for each scenario.

## Scenario One - :

Chart, bubble chart

Description automatically generated





## Scenario Two - :

