



A master thesis presentation over

A cognitive agent to control physical maze via hierarchical learning

- Gaurav Kumar

Index

- Labyrinth Maze Task
- Experiments with DRL (DDPG)
- Motivation and Related Work
- Hierarchical Modeling
 - High-Level Controller (HLC)
 - Low-Level Controller (LLC)
- Other LLC Options

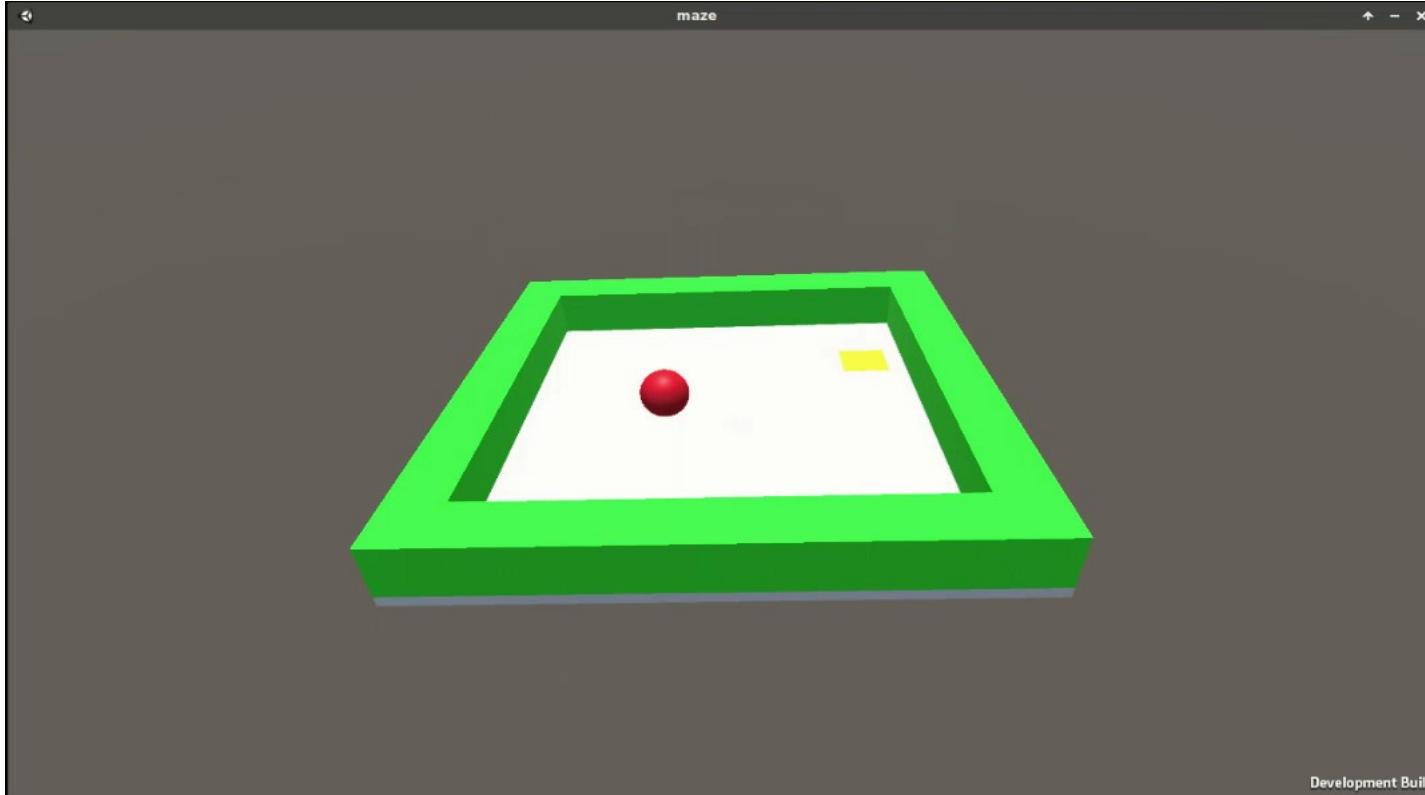
Task

- Environment - Labyrinth
- Agent
 - Ball when Actuation is Angular rotation on (X, Y)-Axis
 - External agent when Actuation is linear velocity on (X, Y)-Axis
- Constraints
 - Walls, Holes etc.

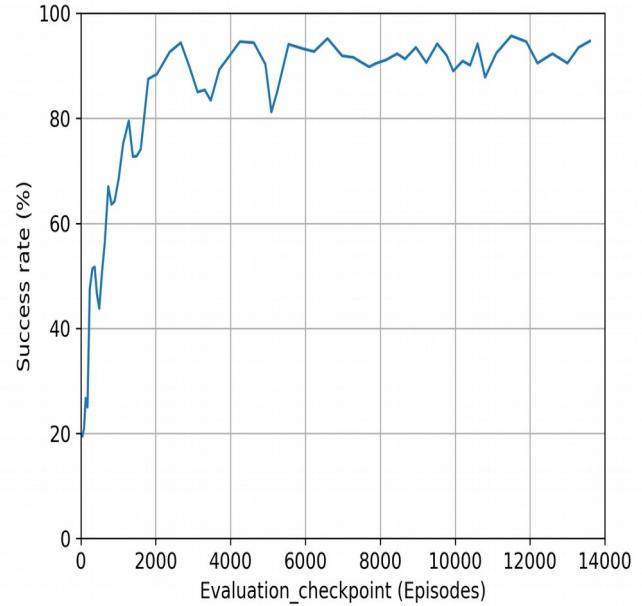
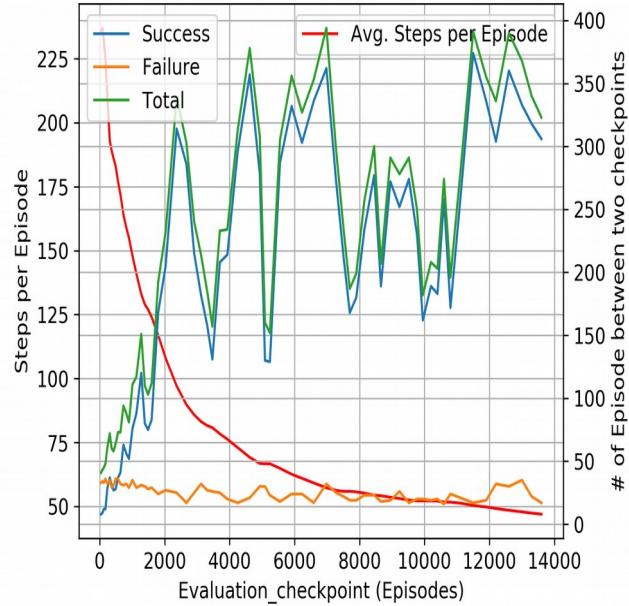
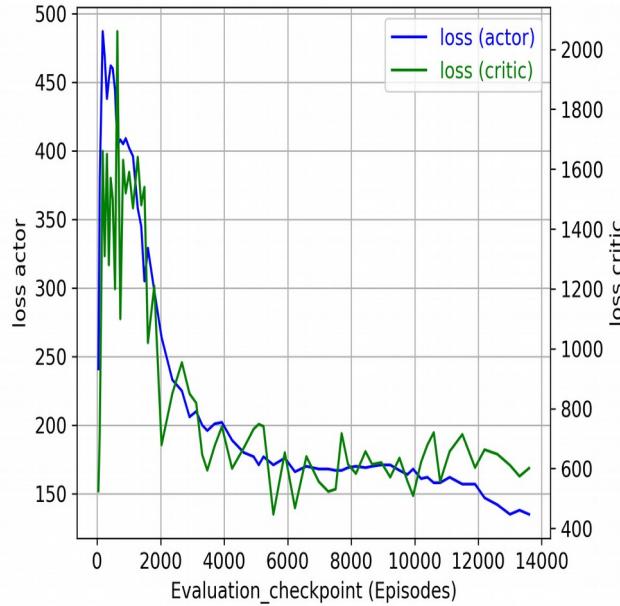
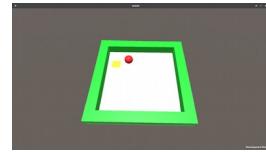


Real Maze

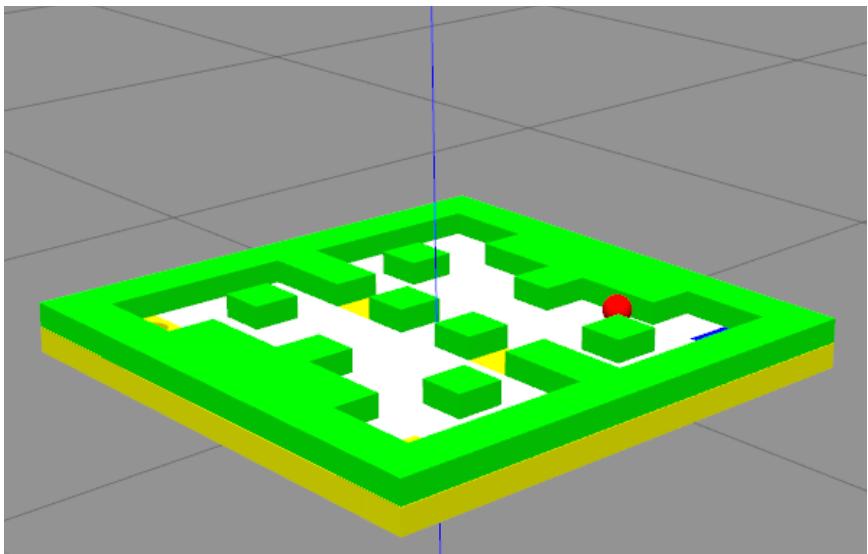
DRL Controller (Unity, DDPG)



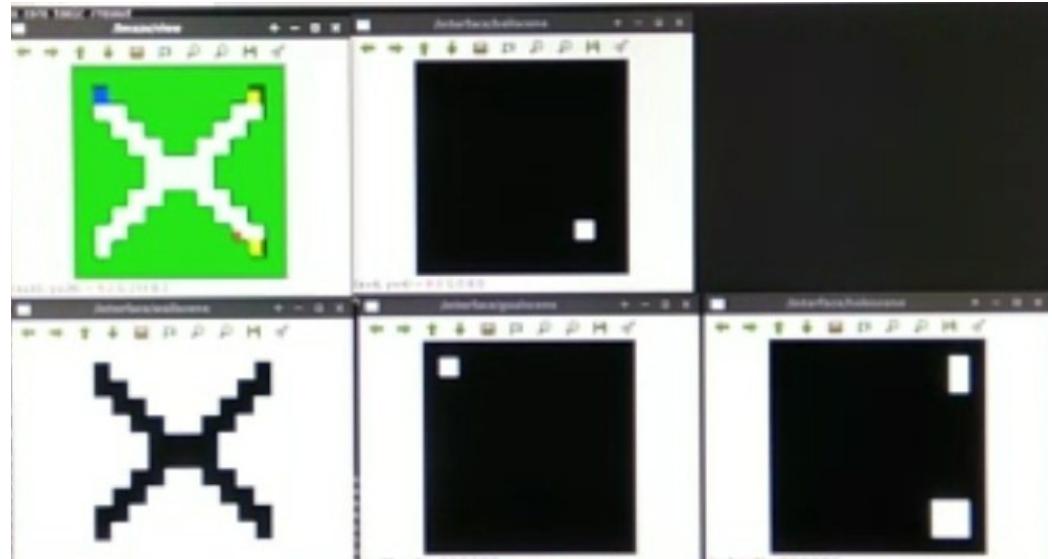
DRL Initial Models (DDPG)



Maze with full Constraints



Simulation Environment

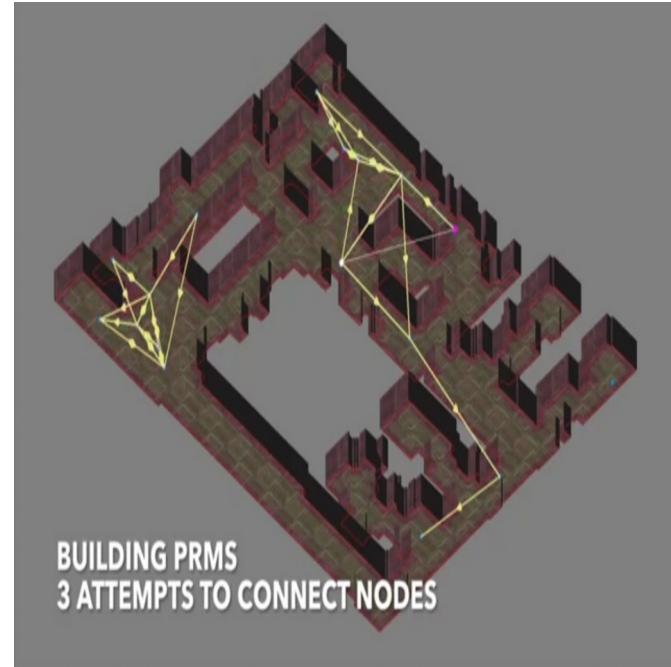
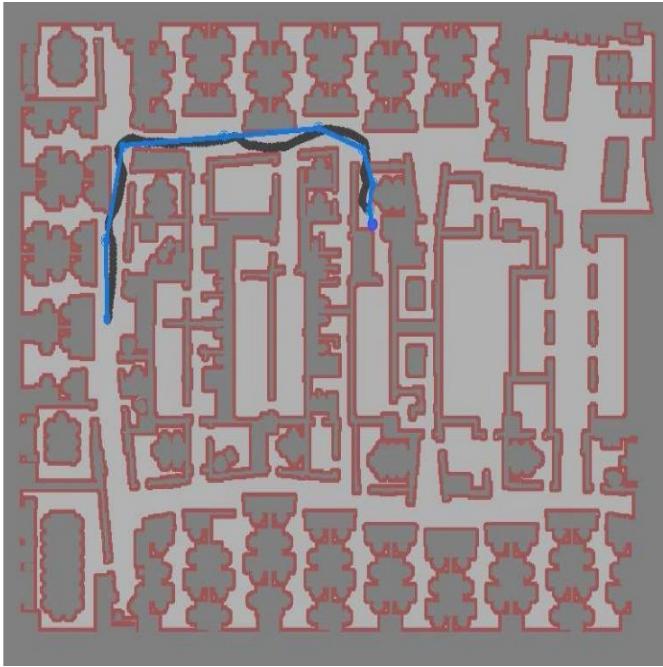


State Representation

Result

- Poor Success Rate
- No Generalization – with different maze layouts

Probabilistic Roadmap-Reinforcement Learning



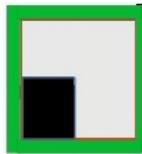
Fovea



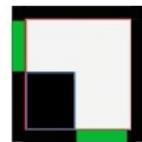
Fovea



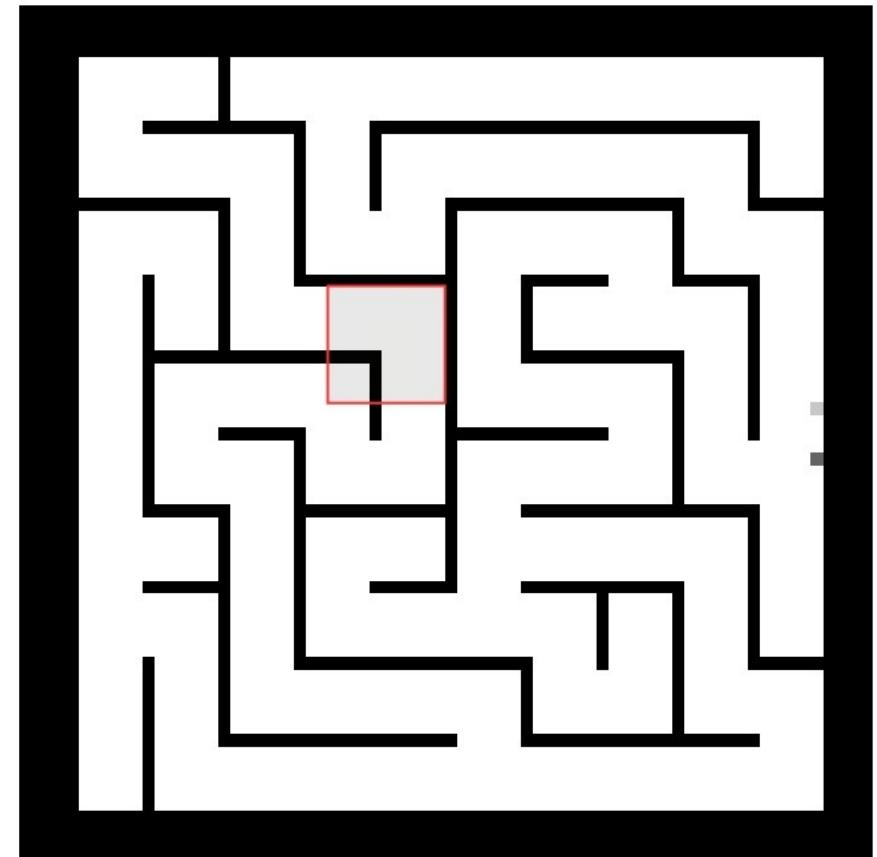
Flood Fill



Boundary Mask

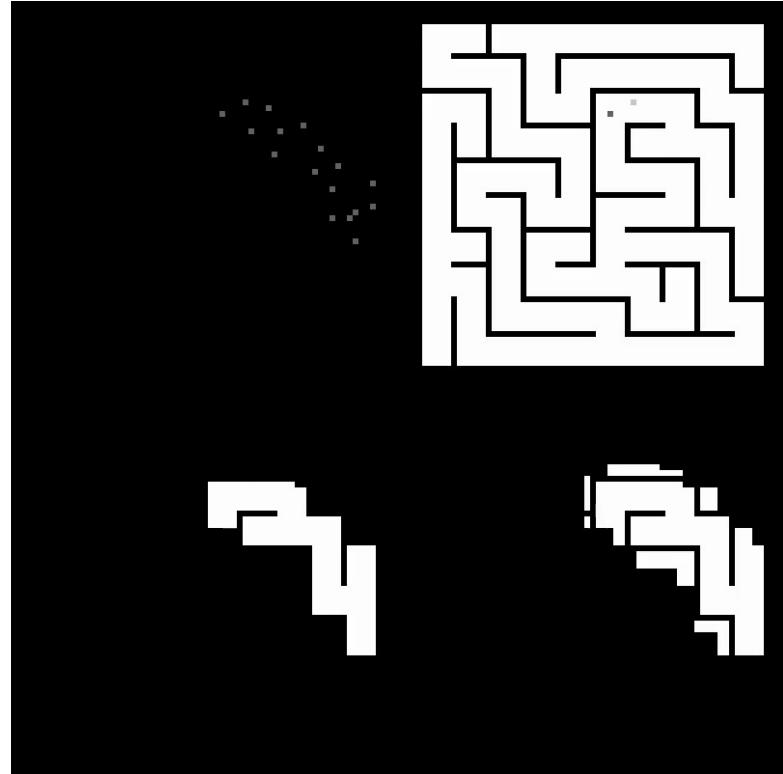


Curiosity Vector



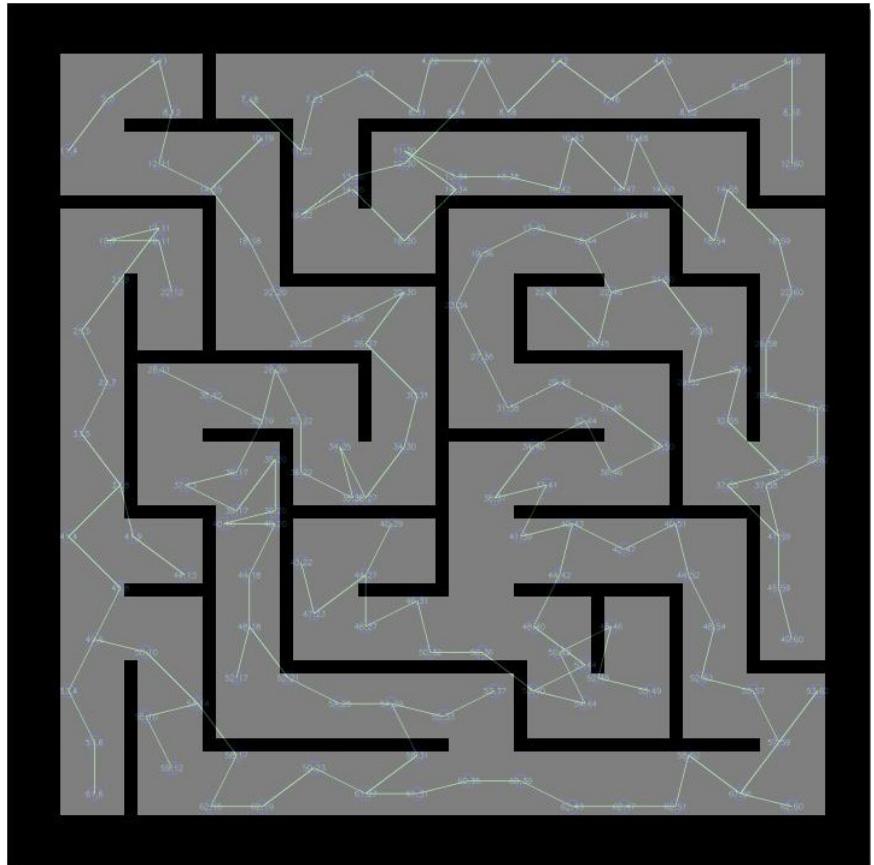
Curiosity Driven Sampling

- Un-necessarily
Populated Graph
- High
Computational
cost at execution



Curiosity Driven Sampling

- Un-necessarily Populated Graph
 - High Computational cost at execution



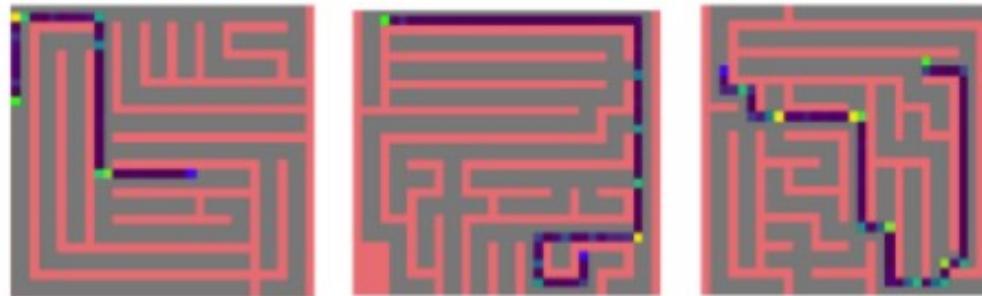
KL-Divergence based Decision Points

Set return,

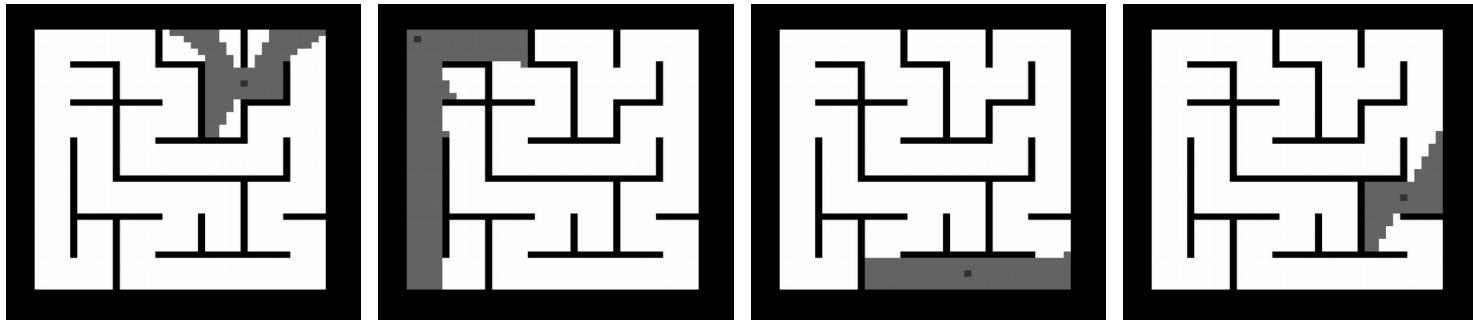
$$y_i = r_i + \beta D_{KL}[p_{enc}(z|s_t, g_t) | q(z|s_t)] \\ + \gamma Q'(s_{i+1}, \mu'(s_{i+1}, Z|\theta^{(\mu+Z)'}) | \theta^{Q'})$$

and update Actor,

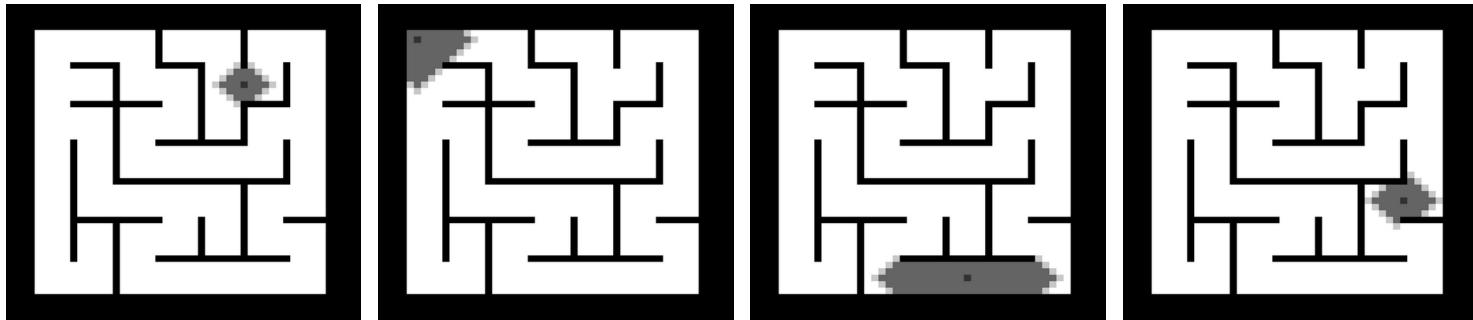
$$\nabla_{\theta^{\mu+Z}} J = \frac{1}{N} \sum_i (\nabla_a Q(s, a | \theta^Q) |_{s=s_i, a=\mu(s_i, Z)} \nabla_{\theta^{\mu+Z}} \mu(s, Z | \theta^{\mu+Z}) |_{s_i} \\ - \beta \nabla_{\theta^{\mu+Z}} D_{KL}[p_{enc}(z|s_t, g_t) | q(z|s_t)])$$



Options for convexity



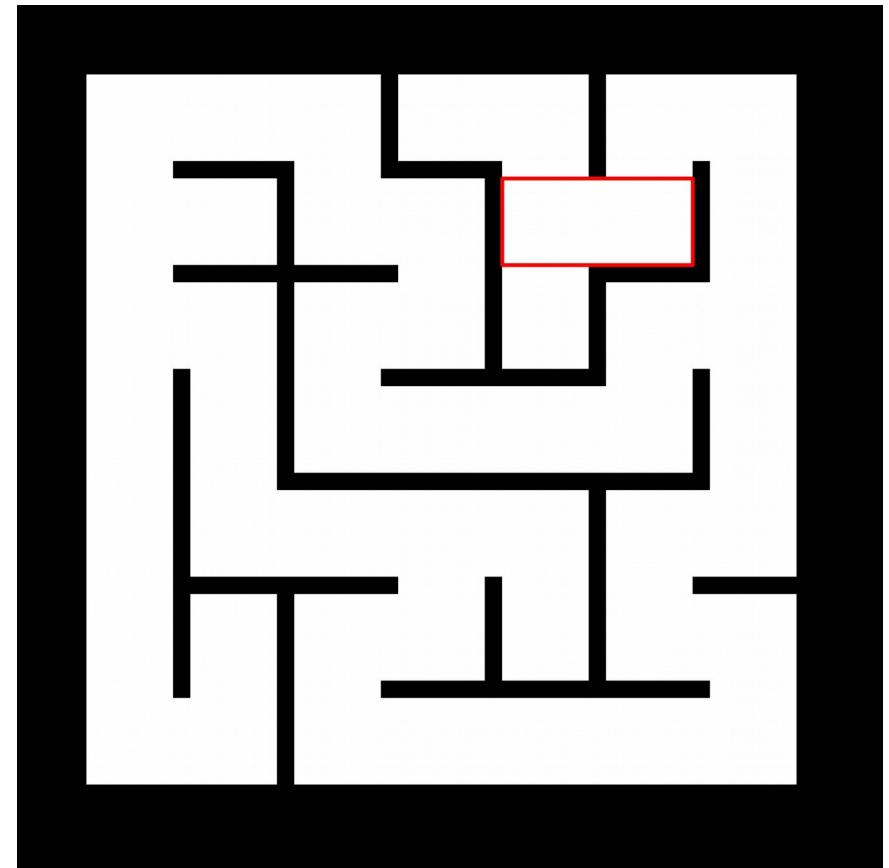
Ray Casting



Convex Flood Fill

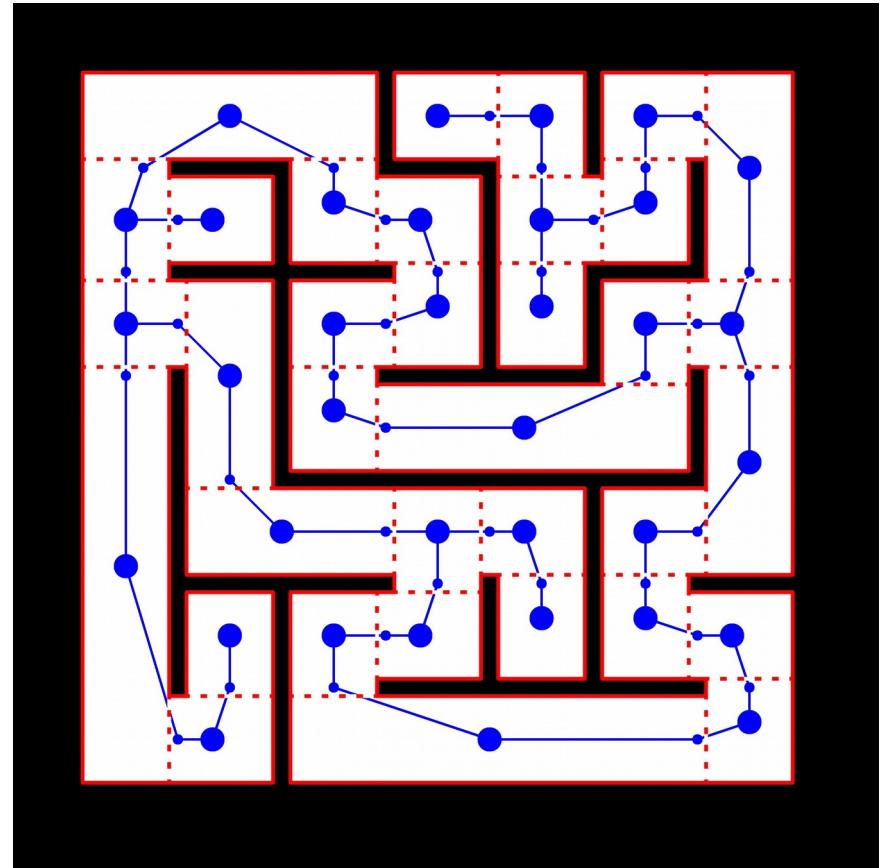
Visual Que – Convex Area

- Rectangular shape
- Convexity violation
- Its Simple !!



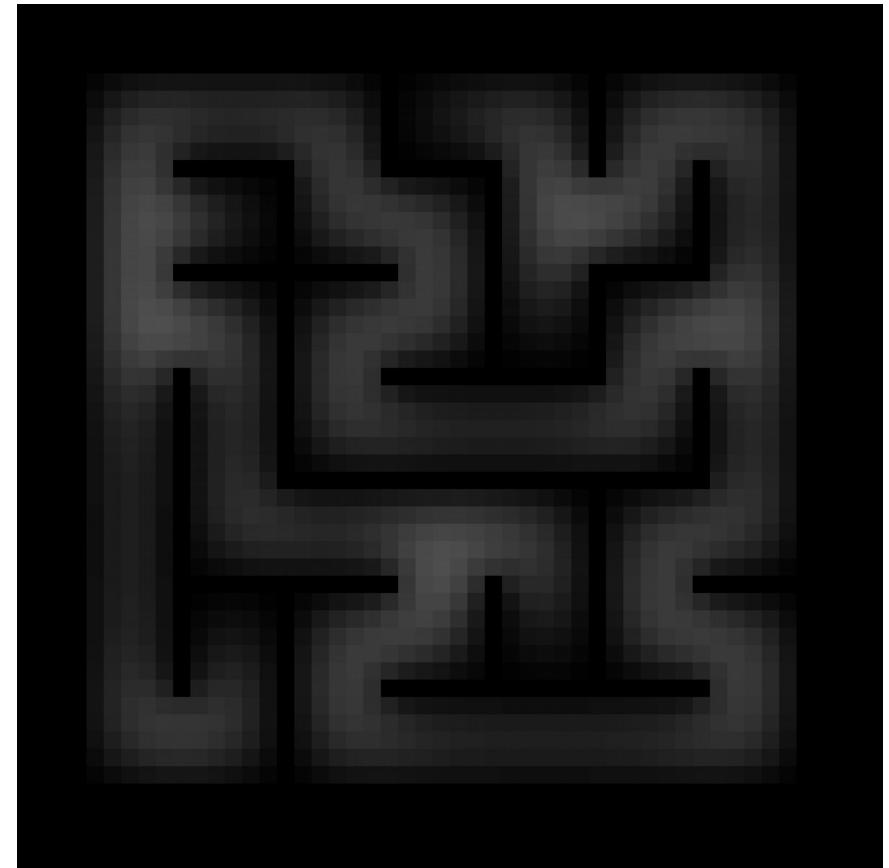
Gateway and Convex-Center

- Center of Convex region
- Gateway Point for adjoining sections
- Sparsely-populated graph
- Resemblance to underlying maze structure



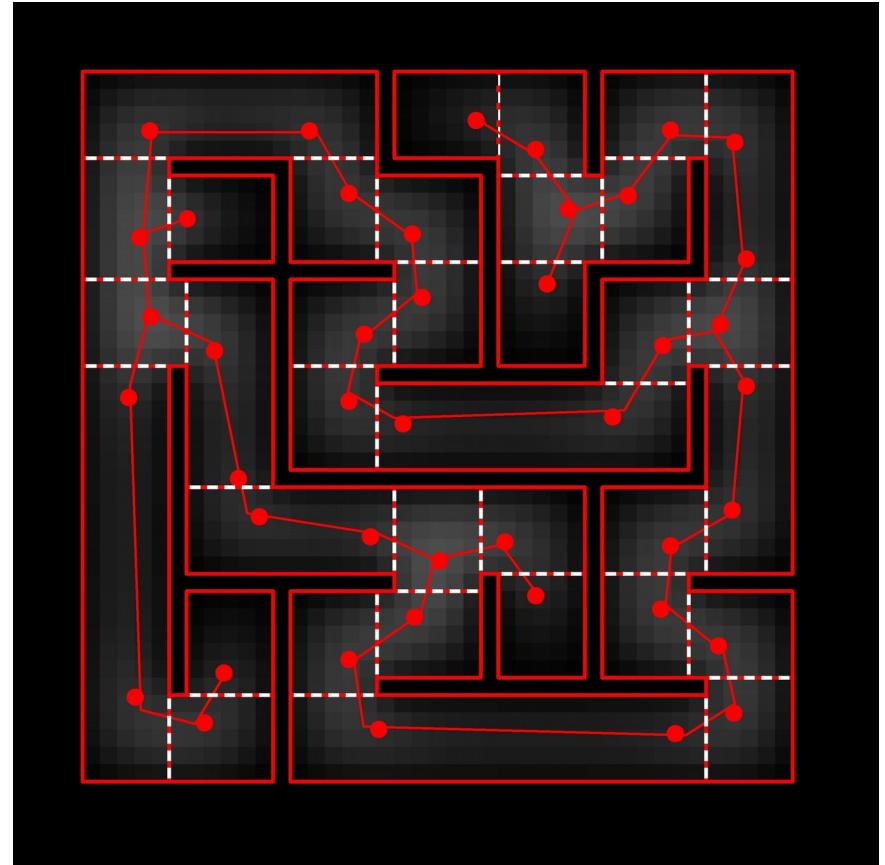
Blurring filter

- Visual Filters
- Blurring reveals interesting property
- Differential brightness



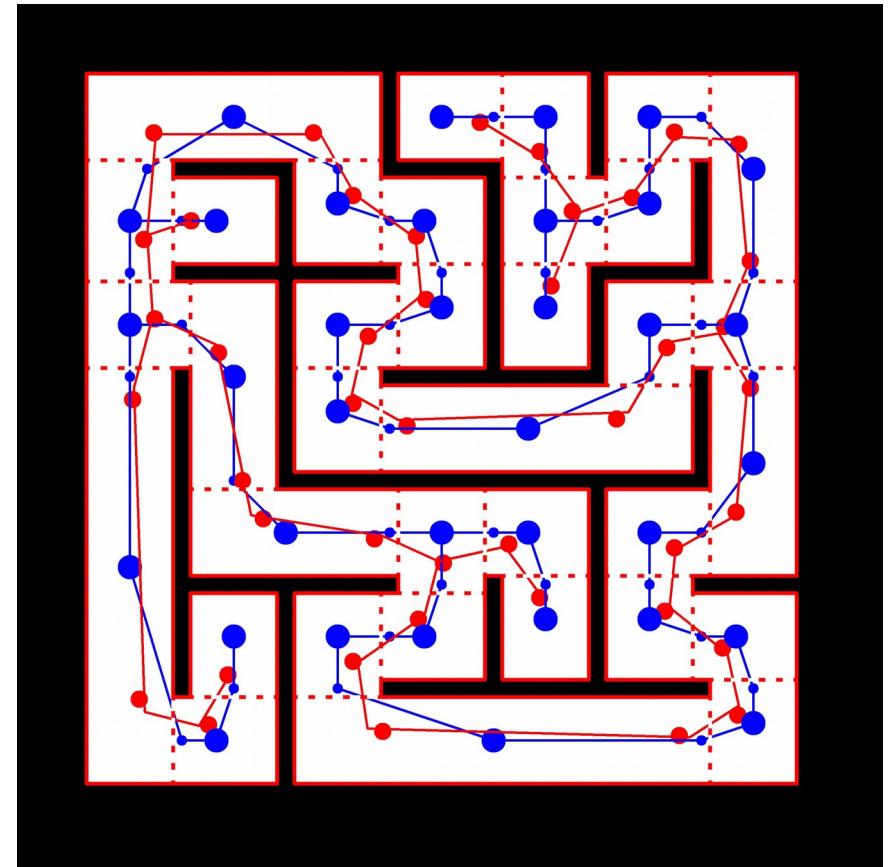
Building the Graph

- But it's a local property !!
- Resemblance to underlying maze structure



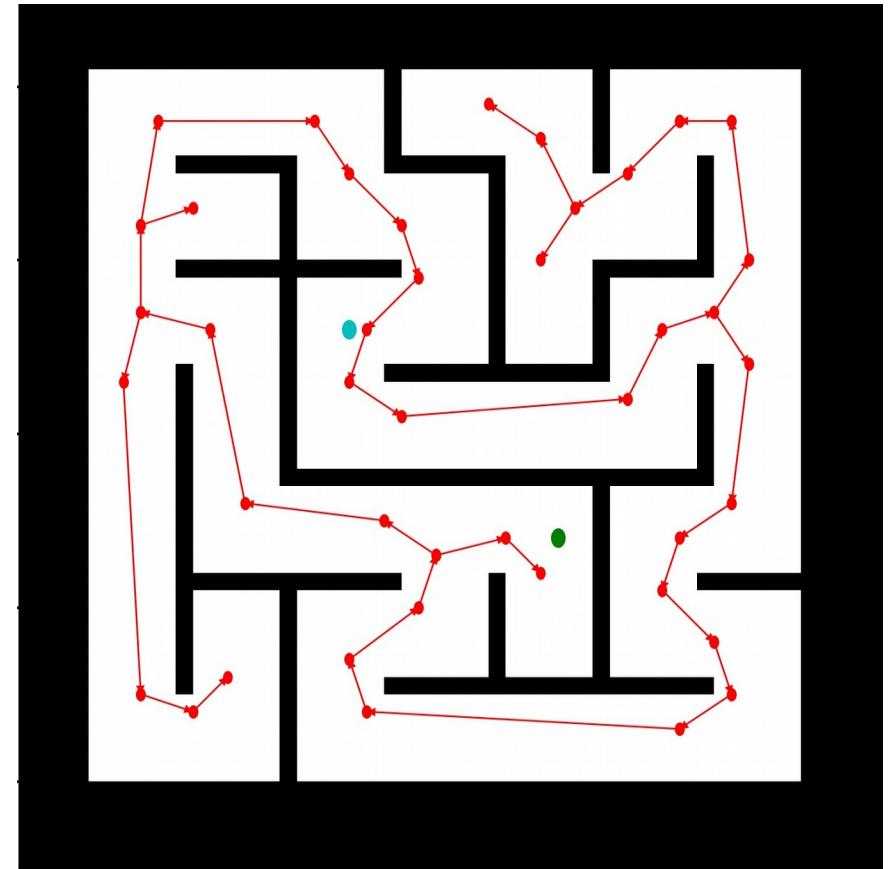
Comparing Approaches

- Blue – Geometric Centers
- Red – Illumination Points
- Lower number of Waypoints
- Better LOS



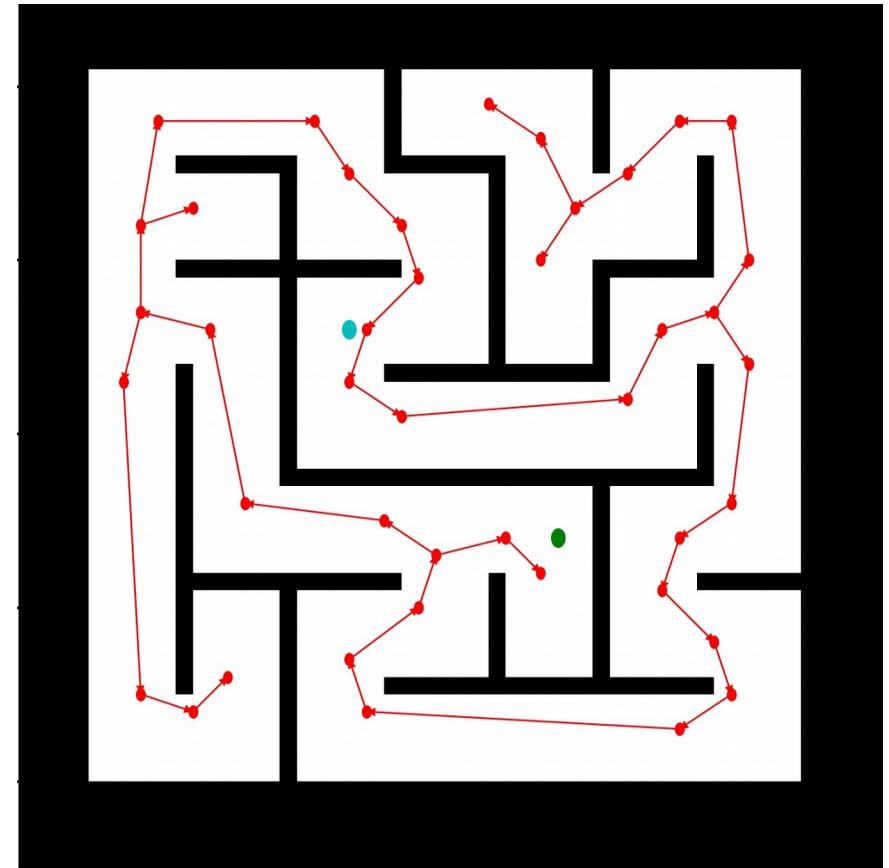
Sample start and goal points

- Cyan – Start
- Green - Goal



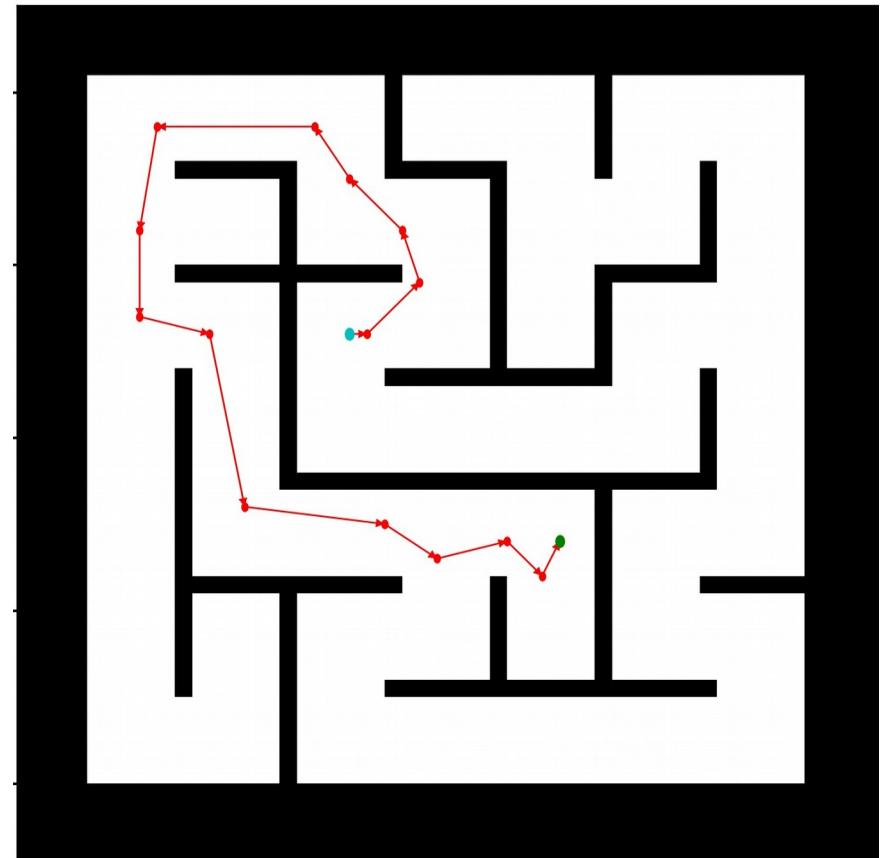
Ask HLC for trajectory

- End-to End Trajectory
- Nearest Neighbor for first and last vertex on graph
- Shortest-Path Search



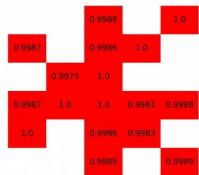
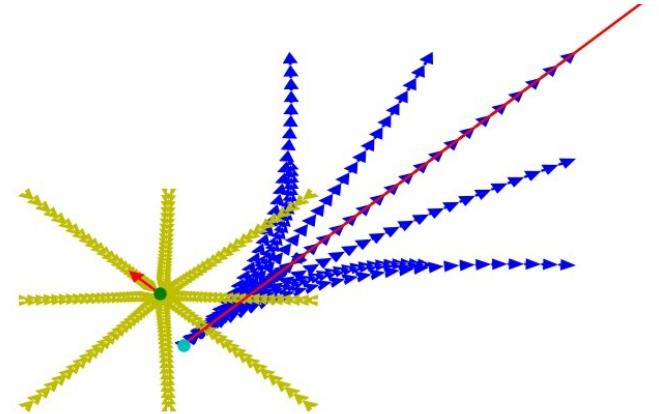
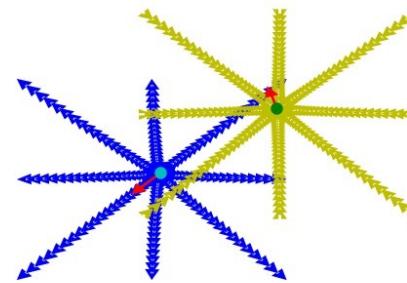
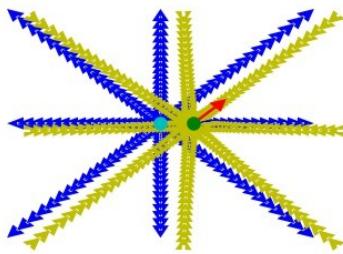
Ask HLC for trajectory

- End-to End Trajectory
- Nearest Neighbor for first and last vertex on graph
- Shortest-Path Search

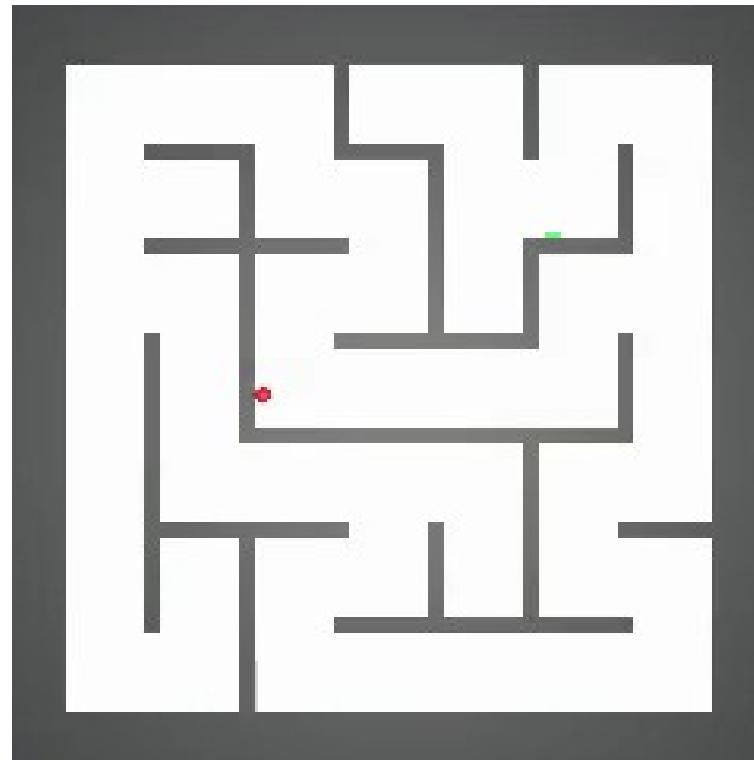


Hierarchical Modeling
LLC

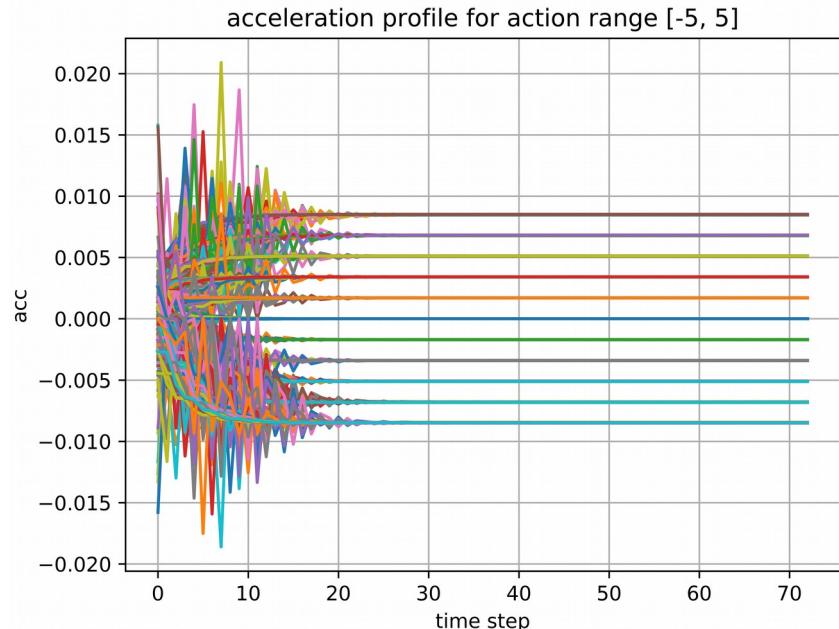
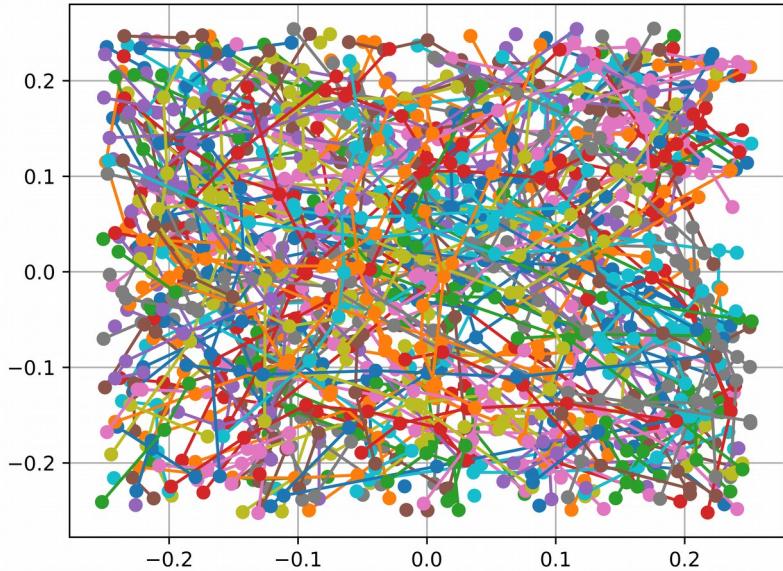
Example Rollout



Rollout Process



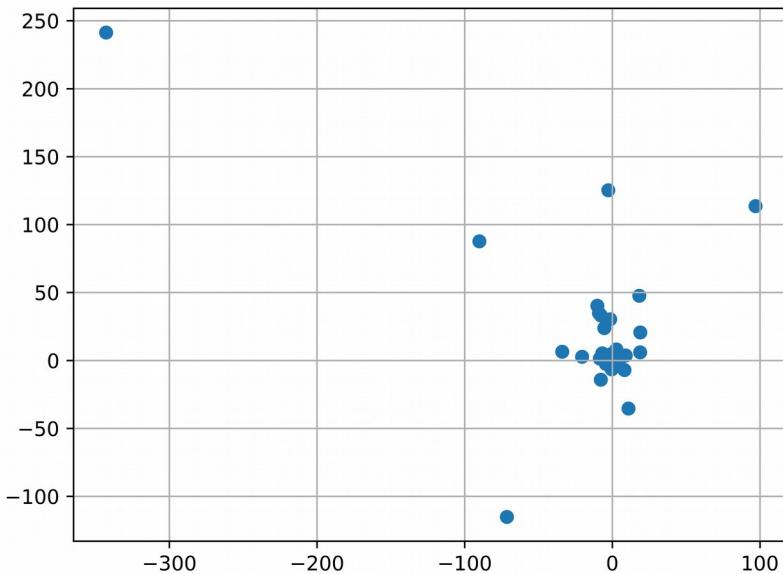
Linear and Angular Velocity



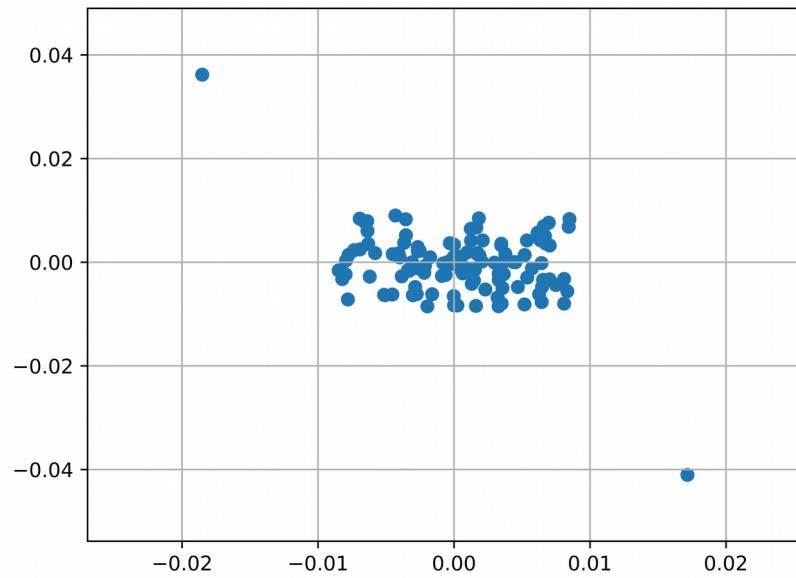
Sampled Data
[Angular_vel_x, Angular_vel_y, Linear_vel_x, Linear_vel_y]

Induced Linear Acceleration Profile

Clustering via k-Means



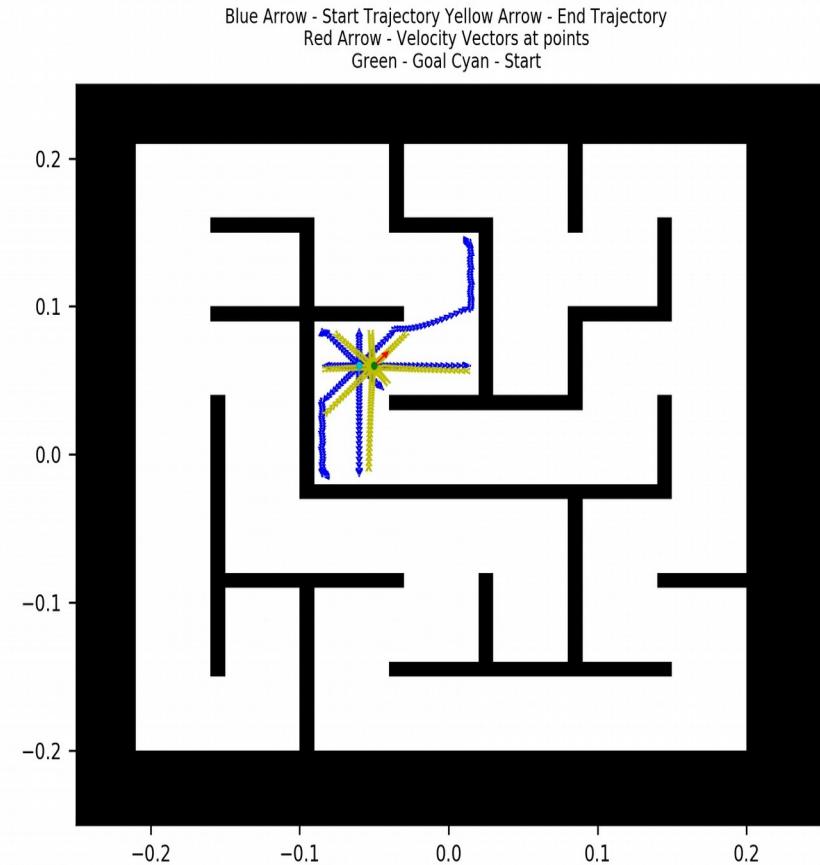
Linear Acceleration Centroid
[Linear_accel_x, Linear_accel_y]



Angular velocity / Command Centroid
[Angular_vel_x, Angular_vel_y]

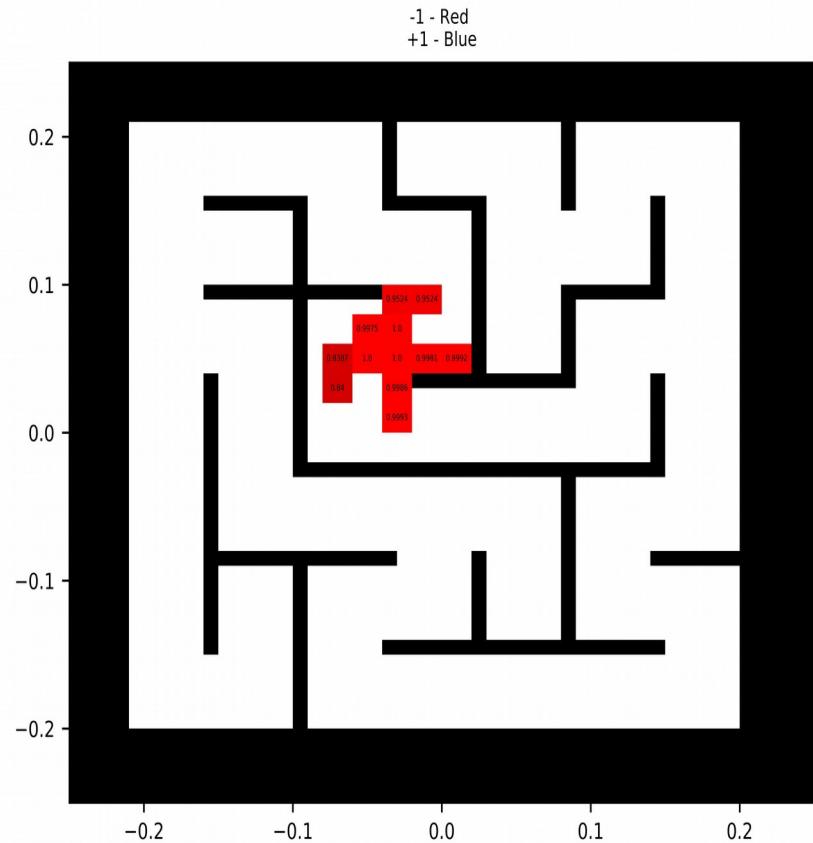
Sampling Waypoint 1

- Blue Rollout – Start
- Yellow Rollout – Goal
- Red Vector – velocity Vector



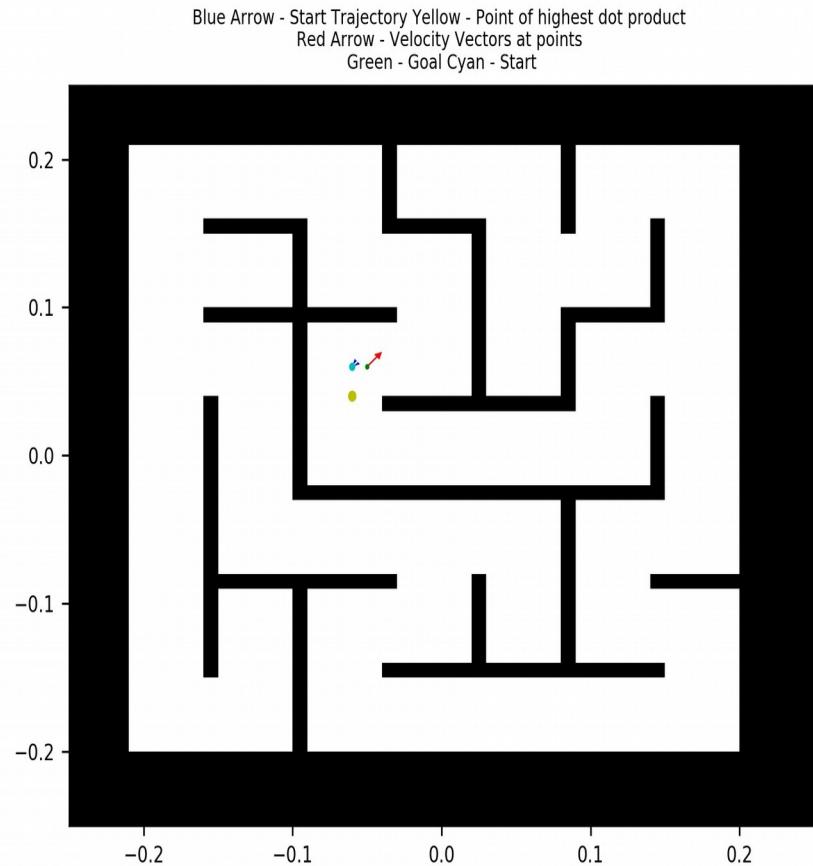
Best Overlap of Trajectories

- Dot Product of rollout vectors
- Best Actions with Highest Dot product
- Point of intersection and steps required to reach there



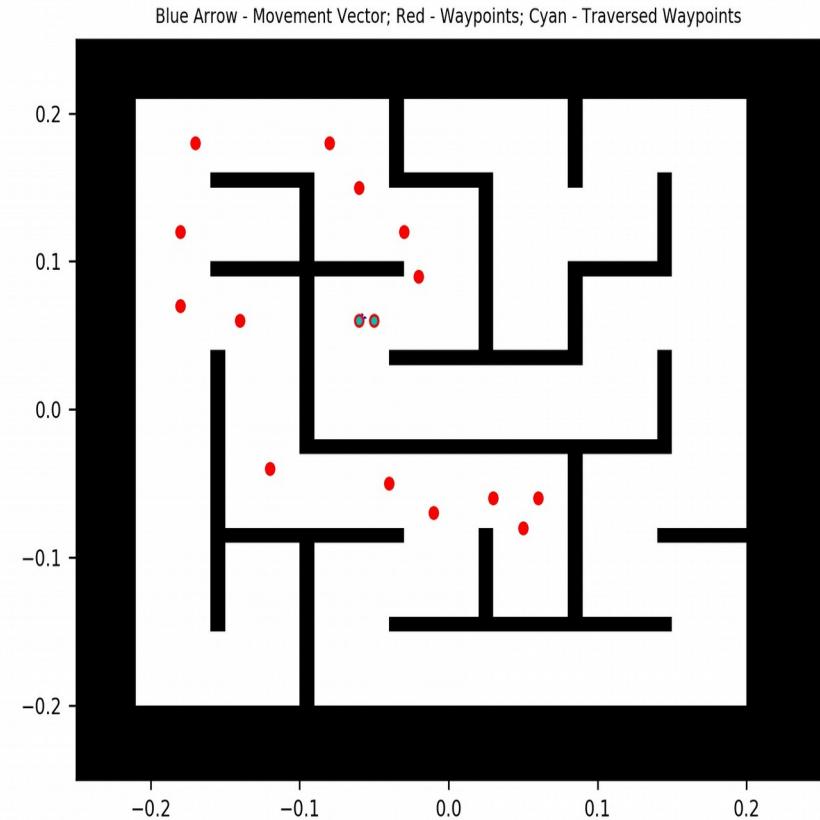
Initial Movement

- Via action giving highest dot product from source rollout
- For predicted number of steps



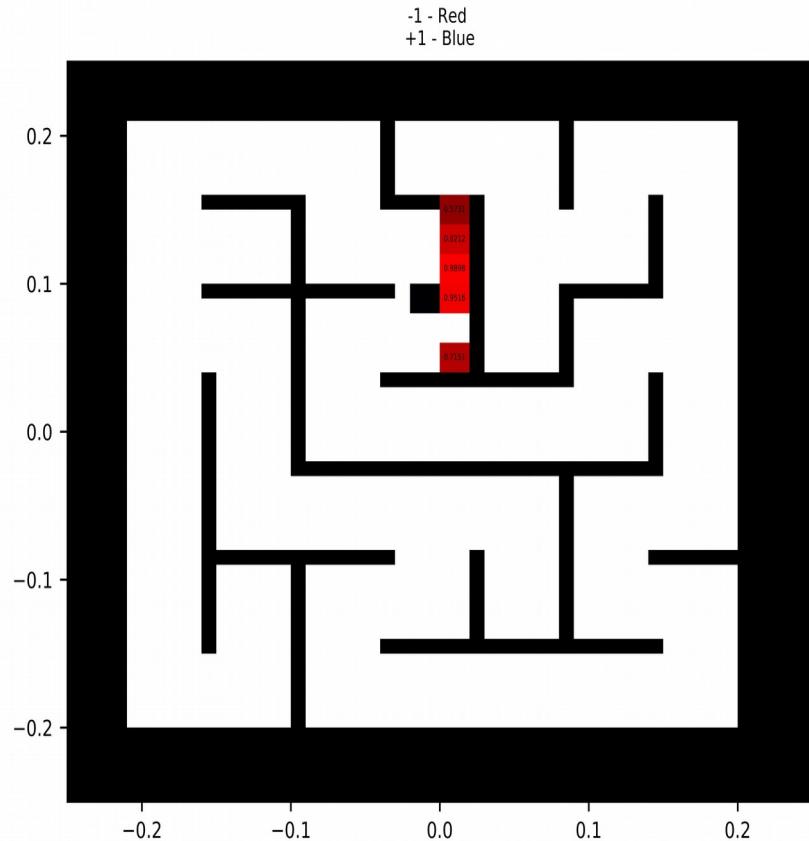
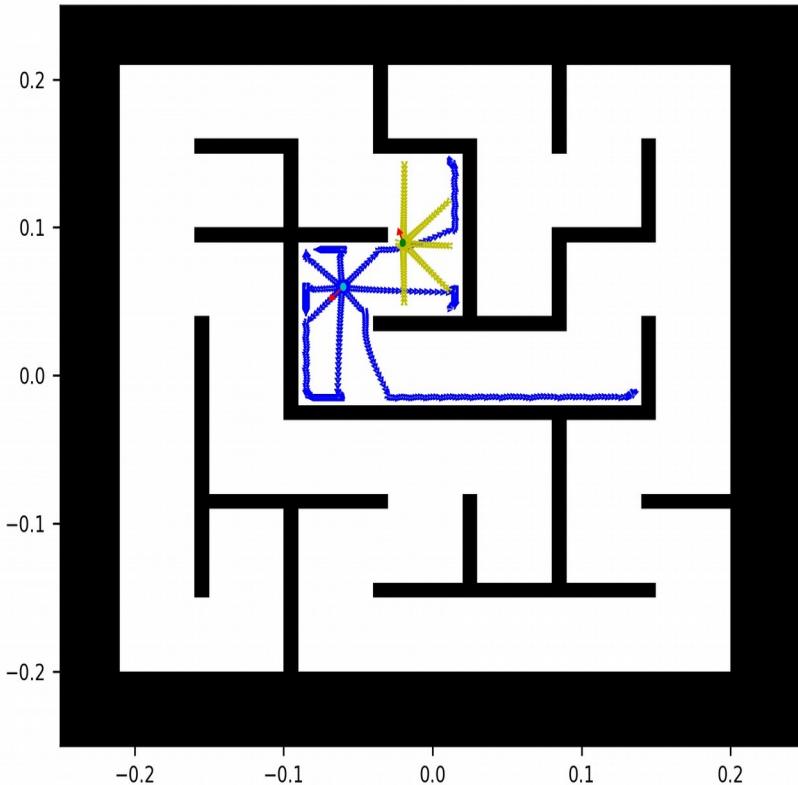
Waypoint Reached

- Check if intermediate state is reached – next waypoint and start by sampling
- If intersection point is reached then correction rollouts



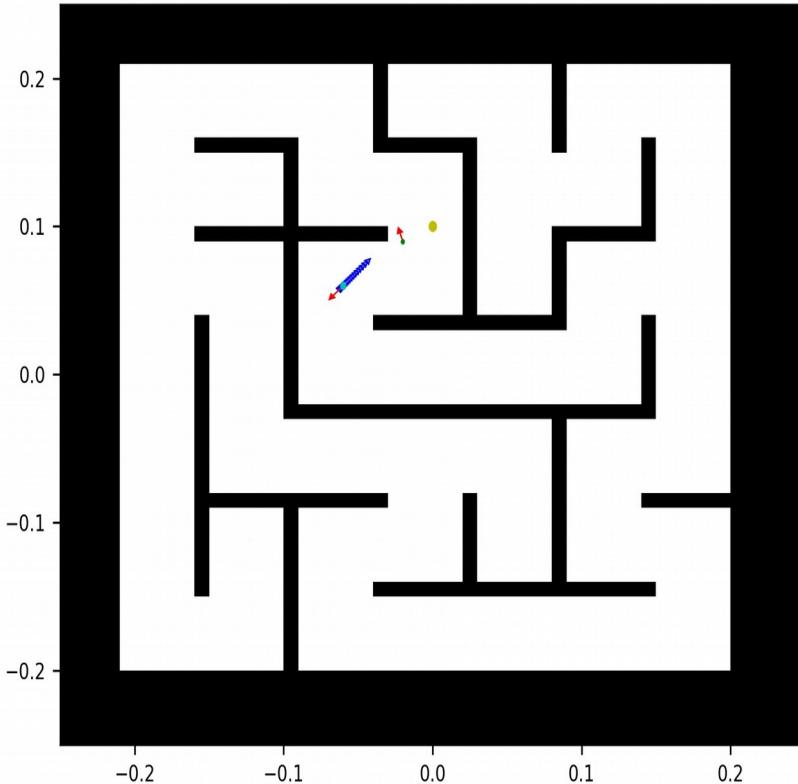
Sampling Waypoint 2 / Best Overlap

Blue Arrow - Start Trajectory Yellow Arrow - End Trajectory
Red Arrow - Velocity Vectors at points
Green - Goal Cyan - Start

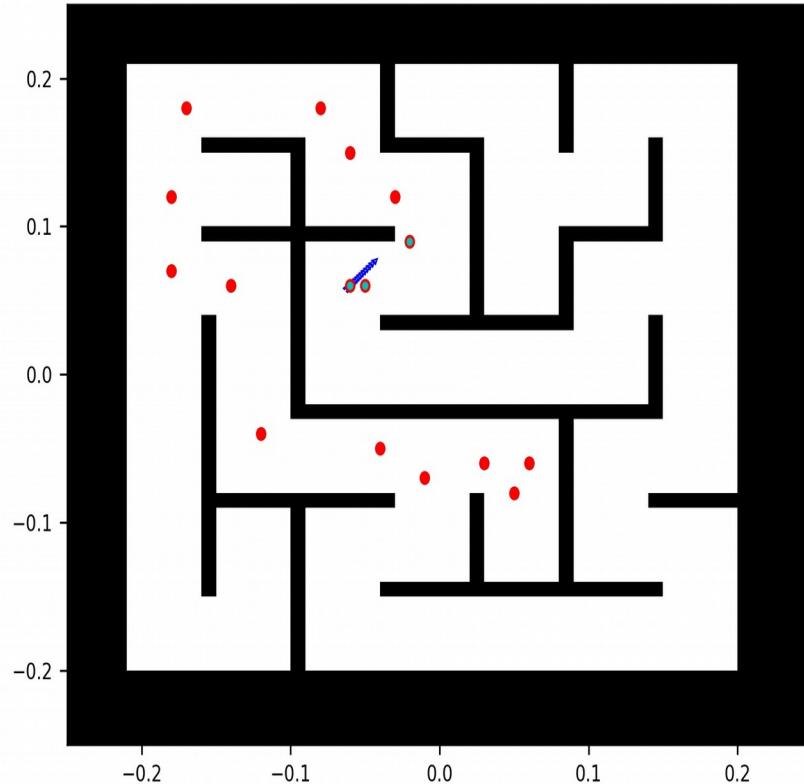


Initial Movement / Waypoint Reached

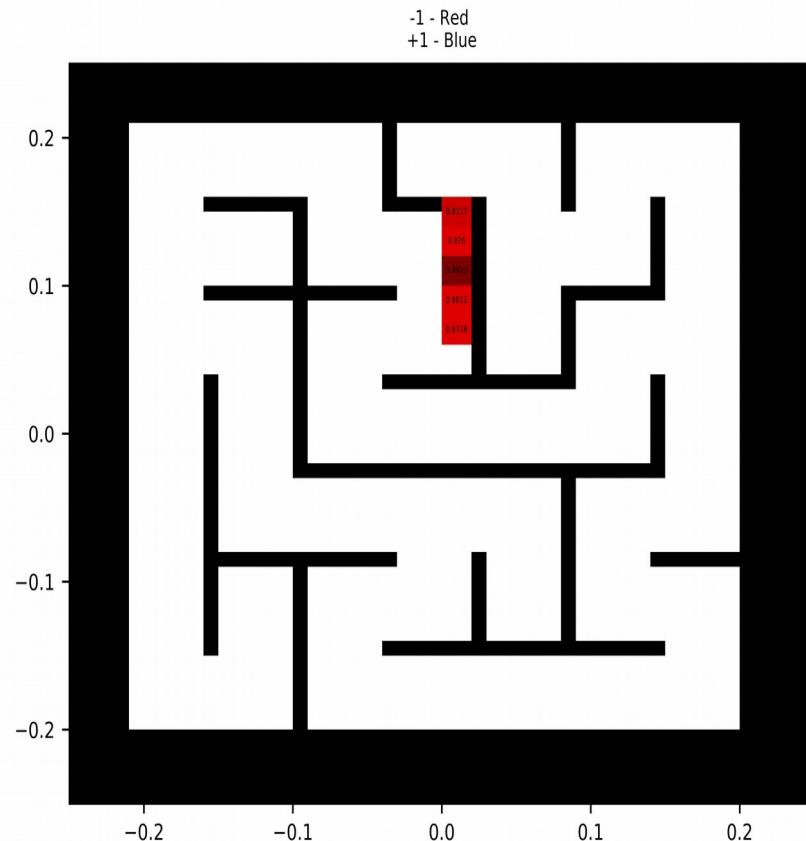
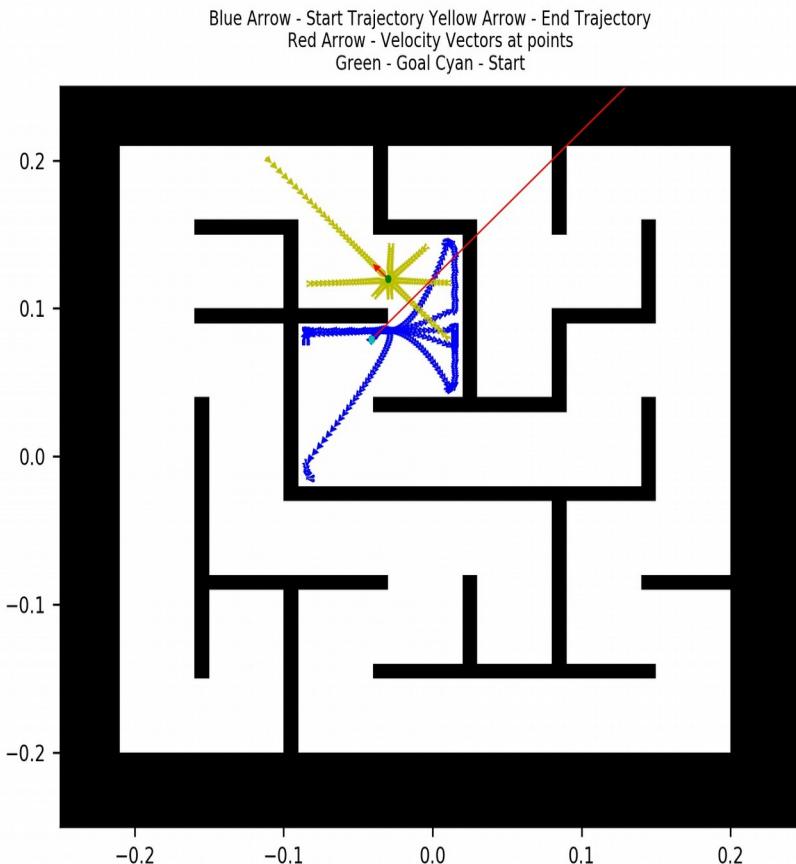
Blue Arrow - Start Trajectory Yellow - Point of highest dot product
Red Arrow - Velocity Vectors at points
Green - Goal Cyan - Start



Blue Arrow - Movement Vector; Red - Waypoints; Cyan - Traversed Waypoints

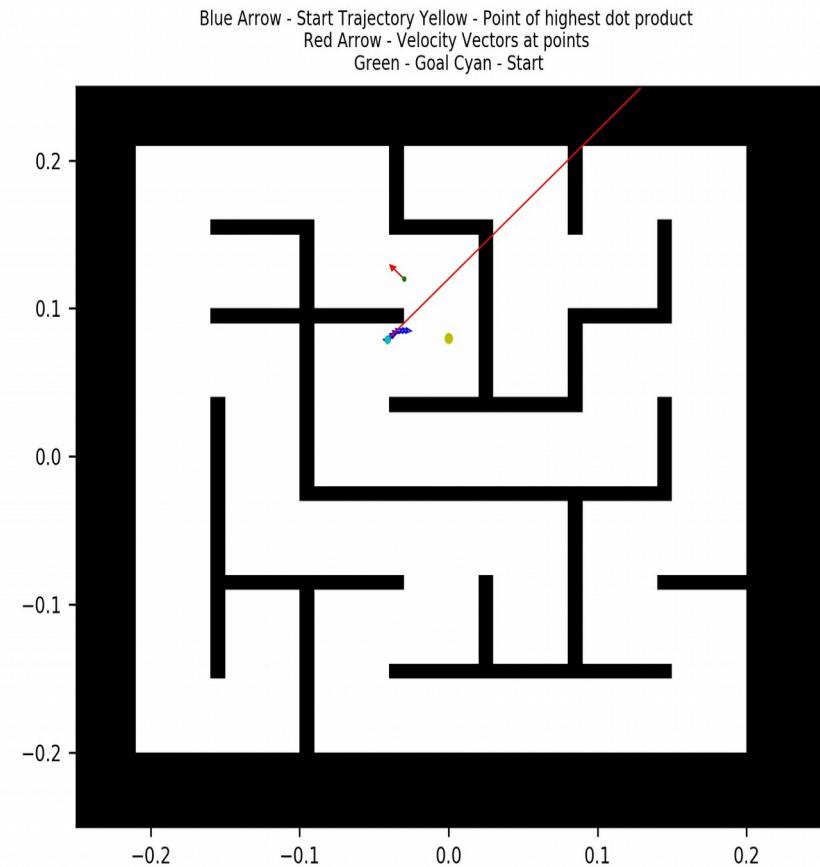


Next Sampling 3 / Best Overlap



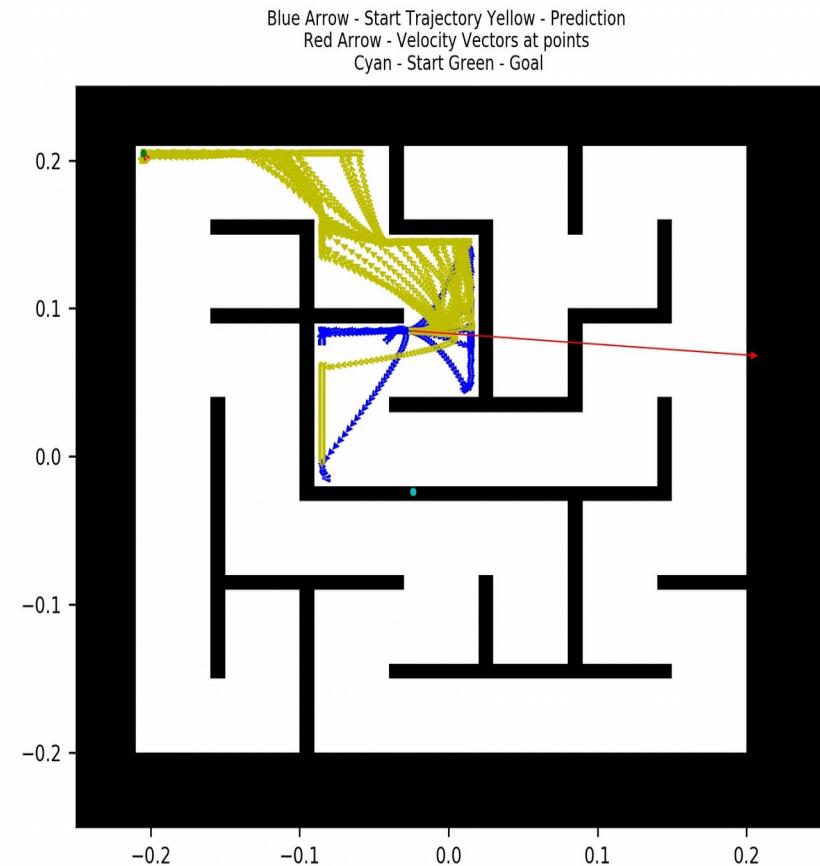
Initial Movement

- intermediate state is not reached in predicted number of steps



Midway Correction

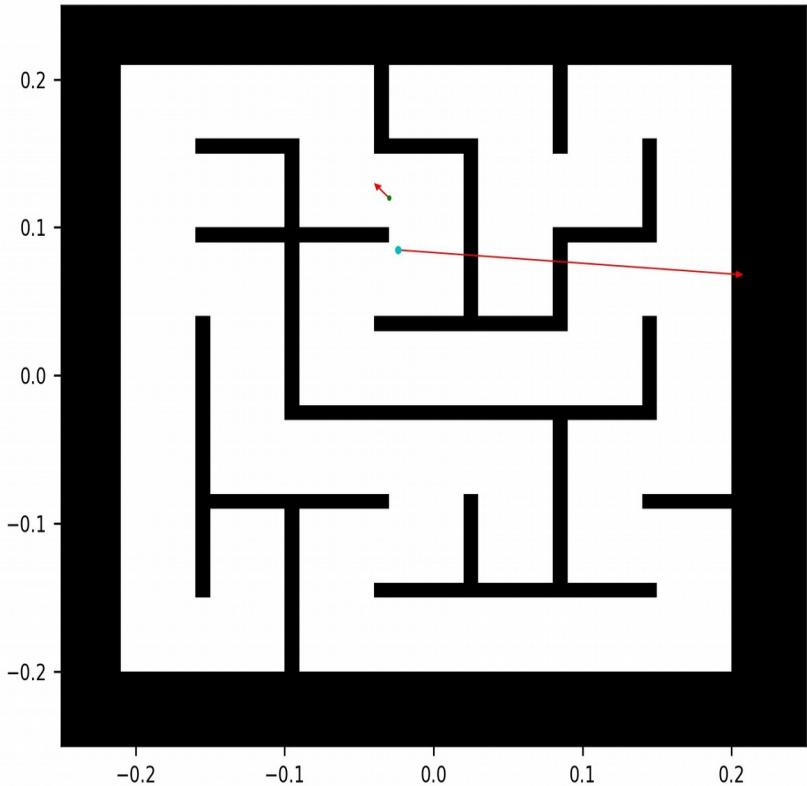
- Noisy rollouts after the current point
- Noise introduced by uniform dist. around next velocity
- Choose the action with noise giving best approach to goal



Final Execution

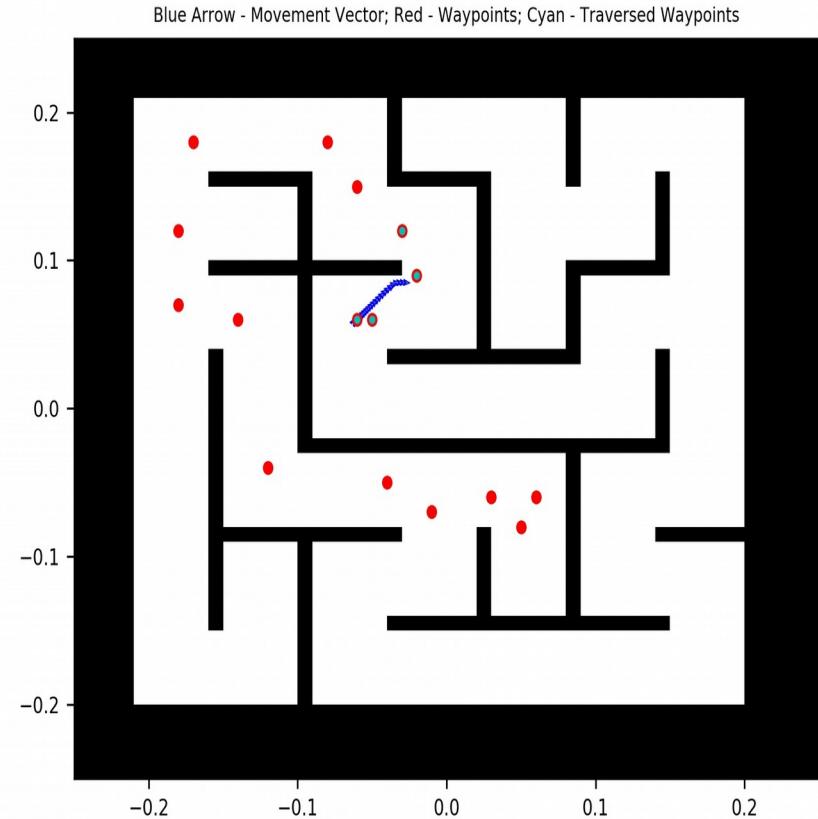
- Execute final action

Blue Arrow - Start Trajectory Yellow - Prediction
Red Arrow - Velocity Vectors at points Cyan - Start Green - Goal
Dot Product : -0.001856580037454011 Distance to goal : 0.024975735703019617



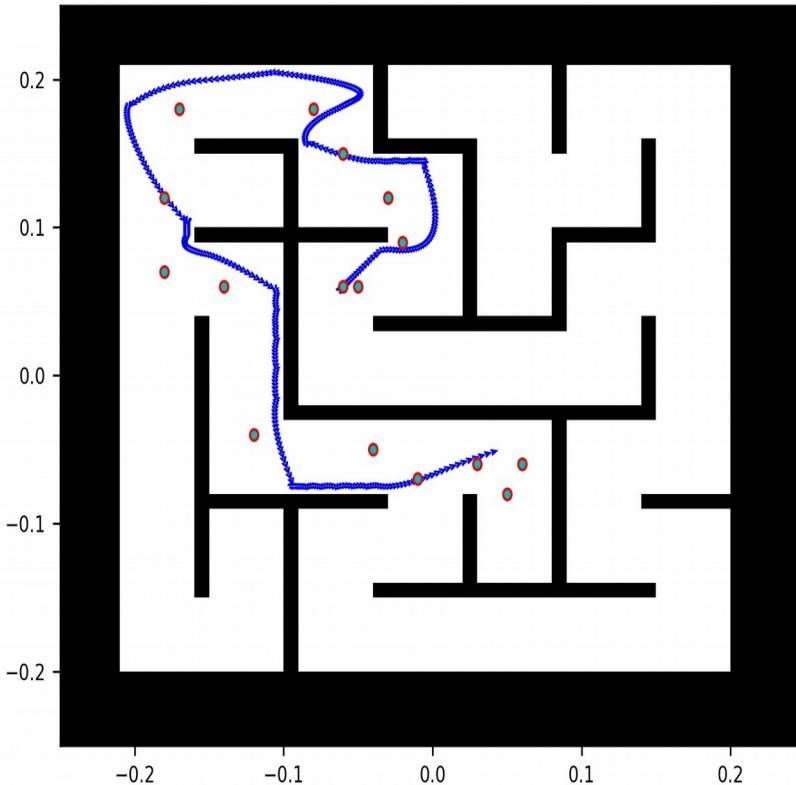
Waypoint Reached

- Next way point if reached
- Else replan from current state until intermediate state

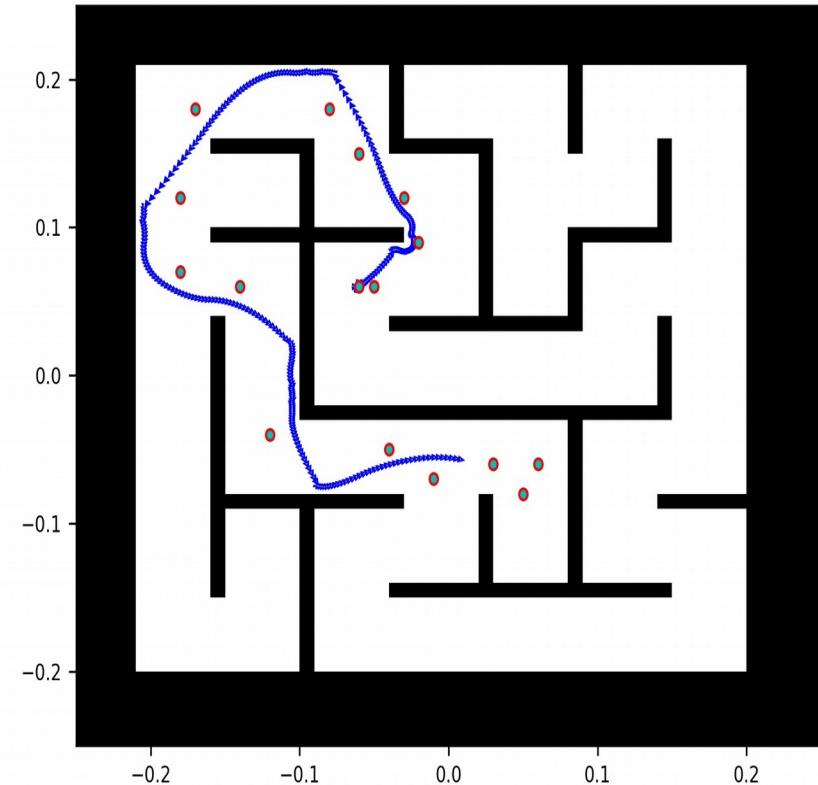


Final Waypoint Reached

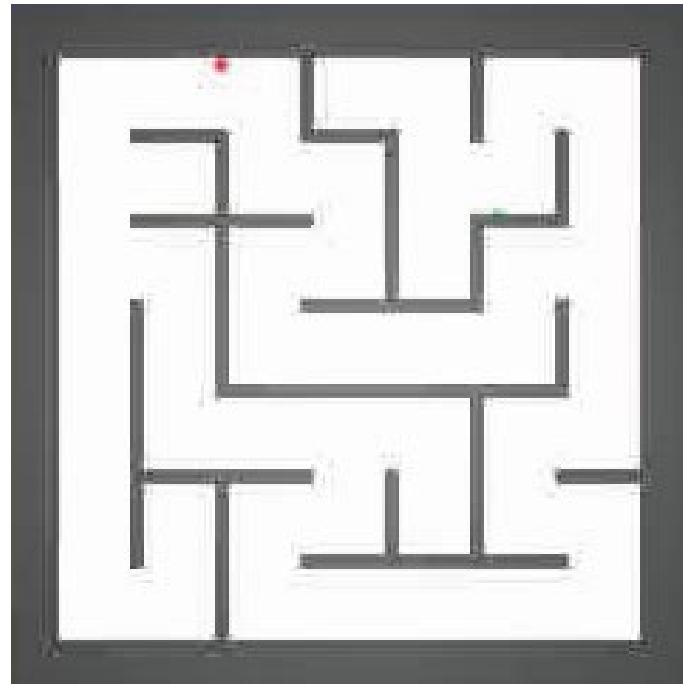
Blue Arrow - Movement Vector; Red - Waypoints; Cyan - Traversed Waypoints



Blue Arrow - Movement Vector; Red - Waypoints; Cyan - Traversed Waypoints



Final Execution



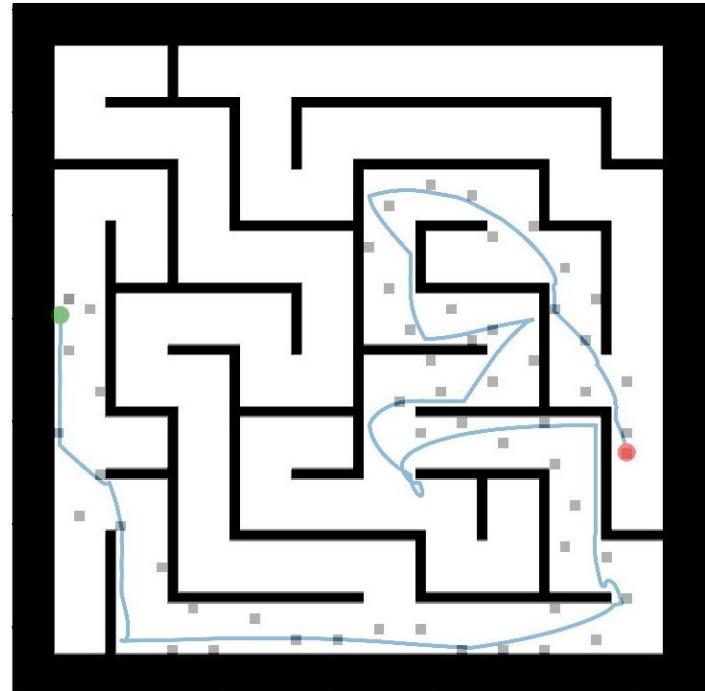
Final Execution

Other LLC Options

- Direct Move Controller
- K-Means Controller
- PID Controller

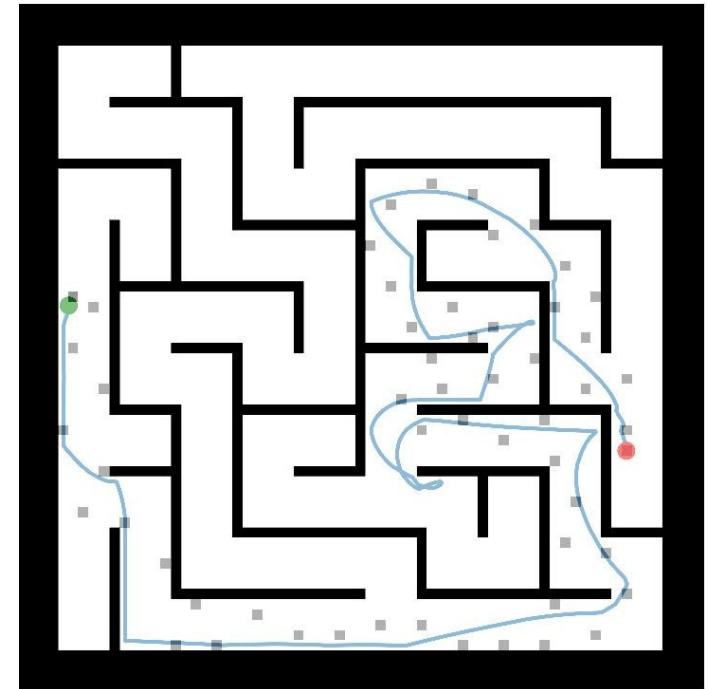
Direct Move Controller

- Translate the displacement vector between two location to command using a fixed Multiplier
- No response to Inertia !!
- High Turn around time !!



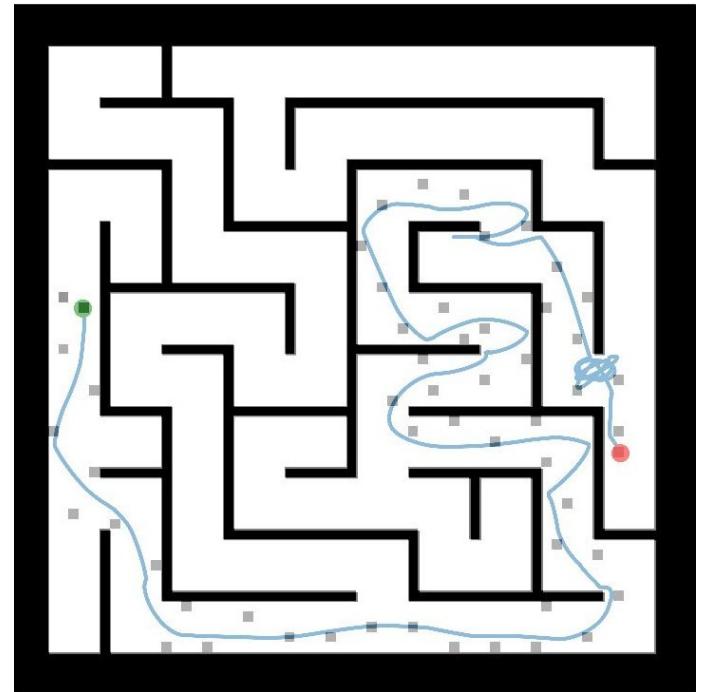
K-means Controller

- Use K-Nearest Neighbors over clusters to find command
- Still no response to Inertia !!
- Variable magnitude of commands !!
- Turn around time !!



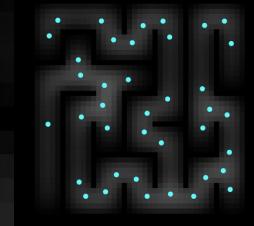
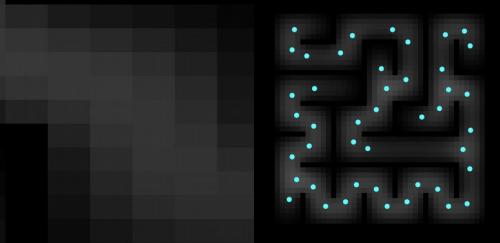
PID controller

- Feed the command from previous setup to a PID controller
- Poor response to Inertia !!
- Turn around time !!



Future Work

- Test the pipeline on a real robot.
- Test on different materials maze.



Thank you for your attention !!