Meeting 08/26/2020

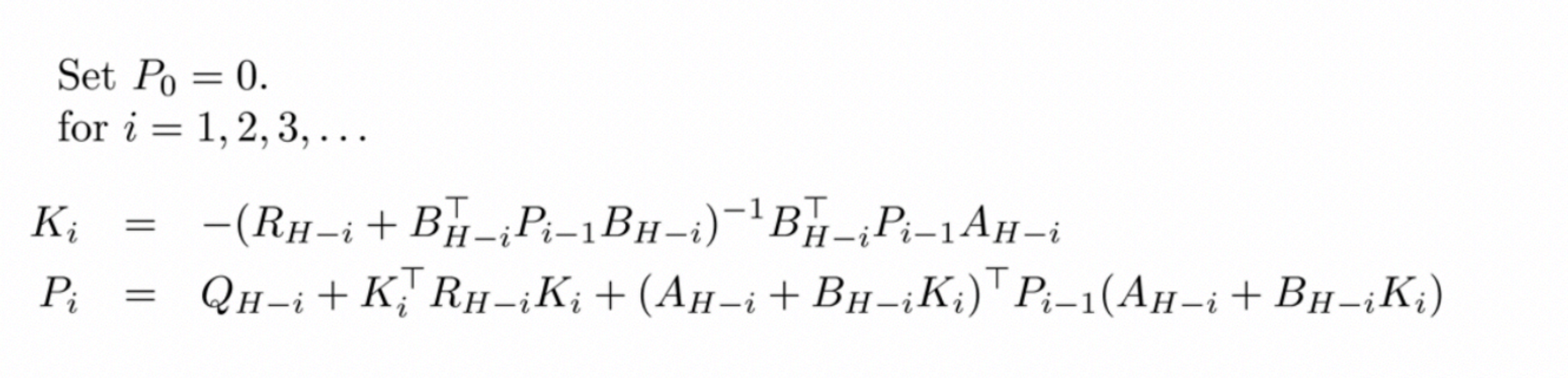
Shuo Zhang

Past week

- Checked the correctness of gradient (Matrix A, B)

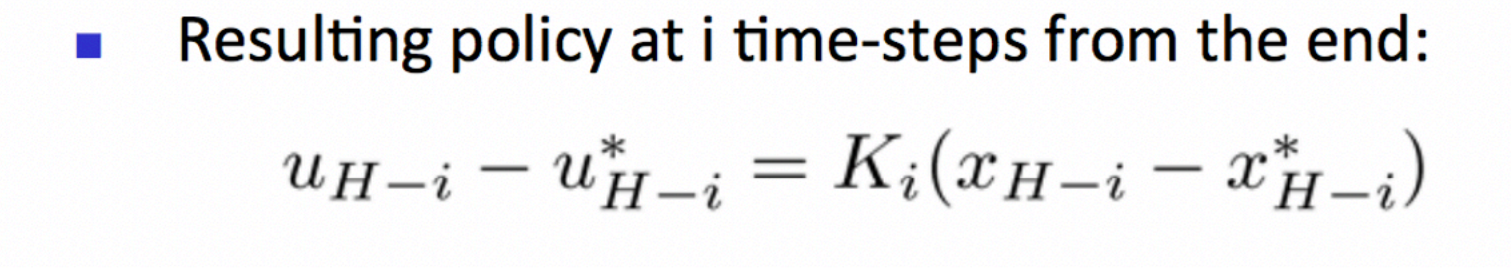
- Calculated Matrix K for all 4 tasks

- Implemented (A*-based)LQR closed-loop control on gazebo hand (5 goal locations)



Set $P_0 = 0$.
for $i = 1, 2, 3, \ldots$

$$\begin{aligned} K_i &= -(R_{H-i} + B_{H-i}^\top P_{i-1} B_{H-i})^{-1} B_{H-i}^\top P_{i-1} A_{H-i} \\ P_i &= Q_{H-i} + K_i^\top R_{H-i} K_i + (A_{H-i} + B_{H-i} K_i)^\top P_{i-1} (A_{H-i} + B_{H-i} K_i) \end{aligned}$$

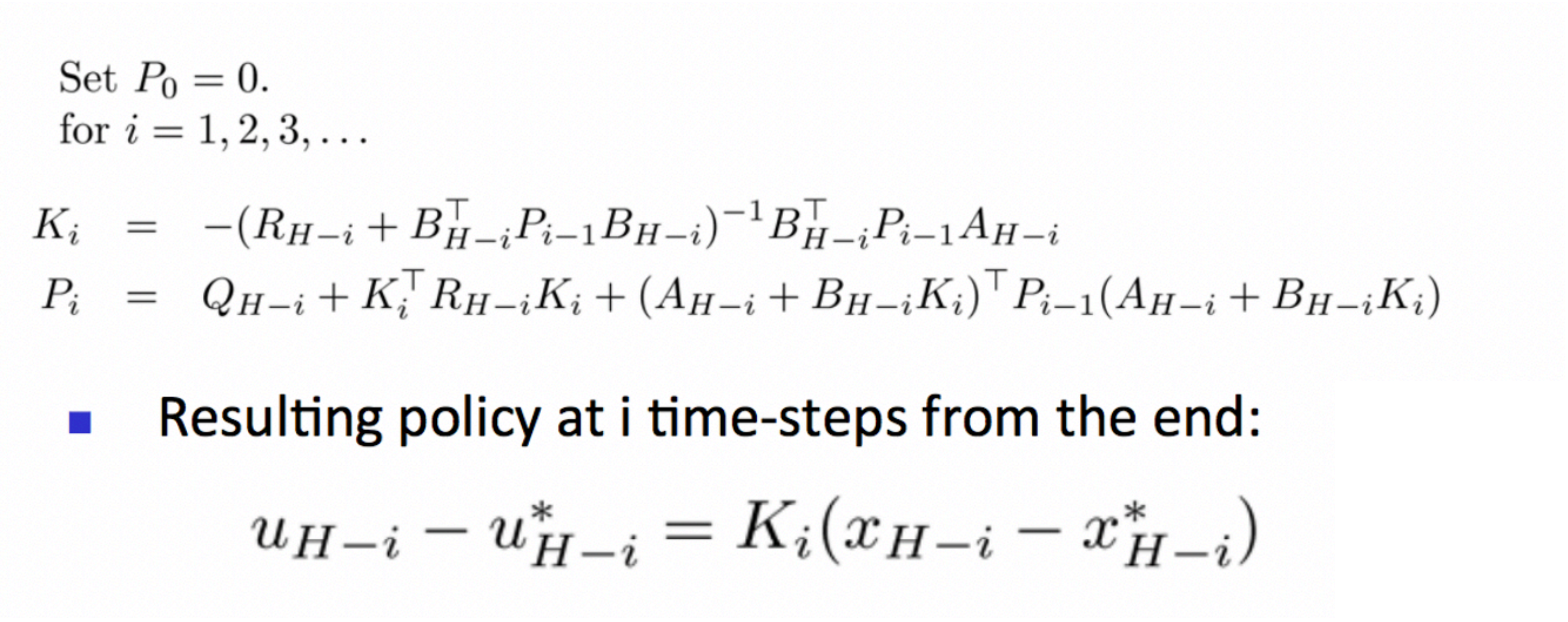


- Resulting policy at i time-steps from the end:

$$u_{H-i} - u_{H-i}^* = K_i (x_{H-i} - x_{H-i}^*)$$

Question

- Acrobot: Actions are discrete(0,1,2).
- How to update u_i?



Set $P_0 = 0$.
for $i = 1, 2, 3, \ldots$

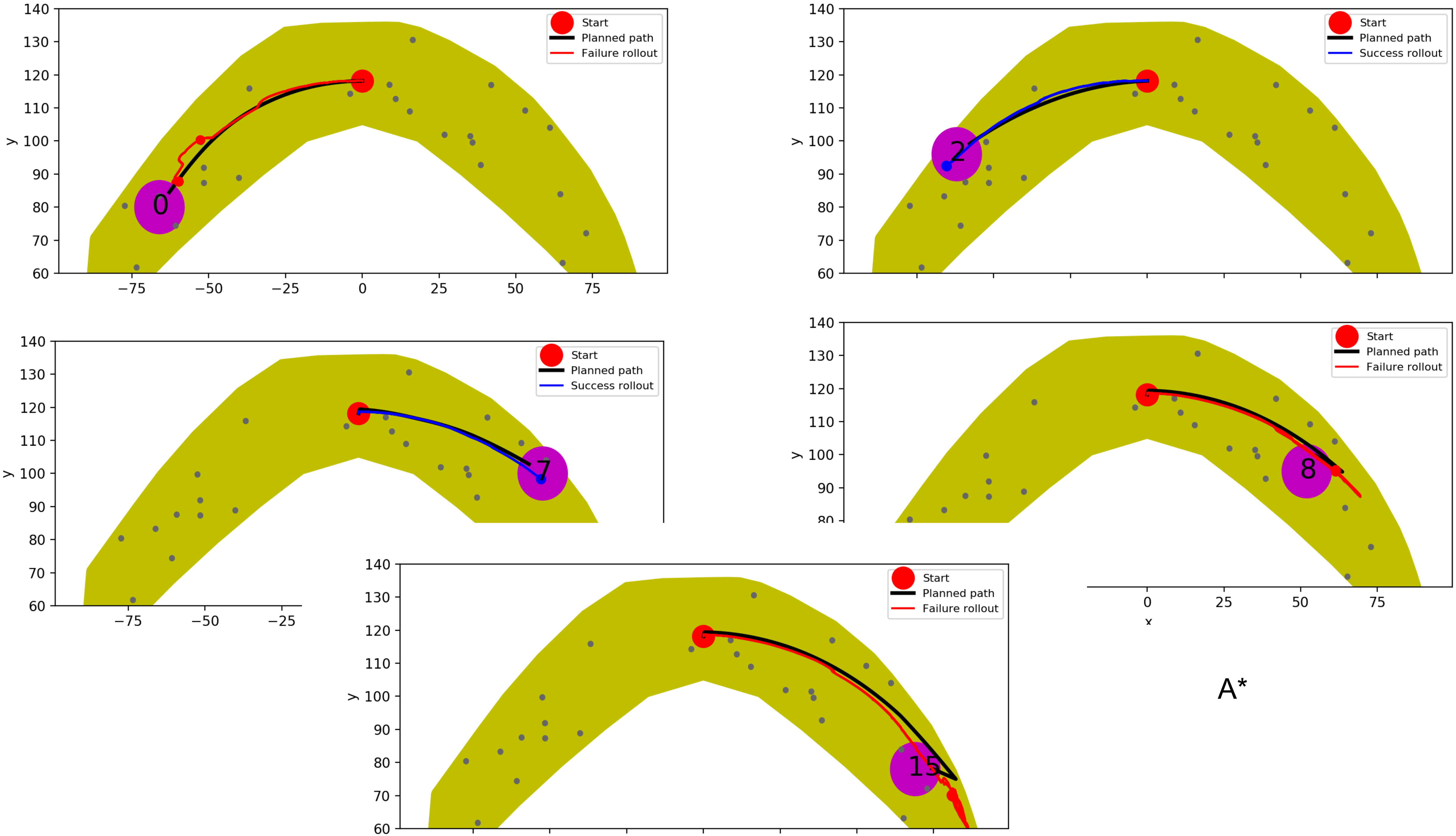
$$\begin{aligned} K_i &= -(R_{H-i} + B_{H-i}^\top P_{i-1} B_{H-i})^{-1} B_{H-i}^\top P_{i-1} A_{H-i} \\ P_i &= Q_{H-i} + K_i^\top R_{H-i} K_i + (A_{H-i} + B_{H-i} K_i)^\top P_{i-1} (A_{H-i} + B_{H-i} K_i) \end{aligned}$$

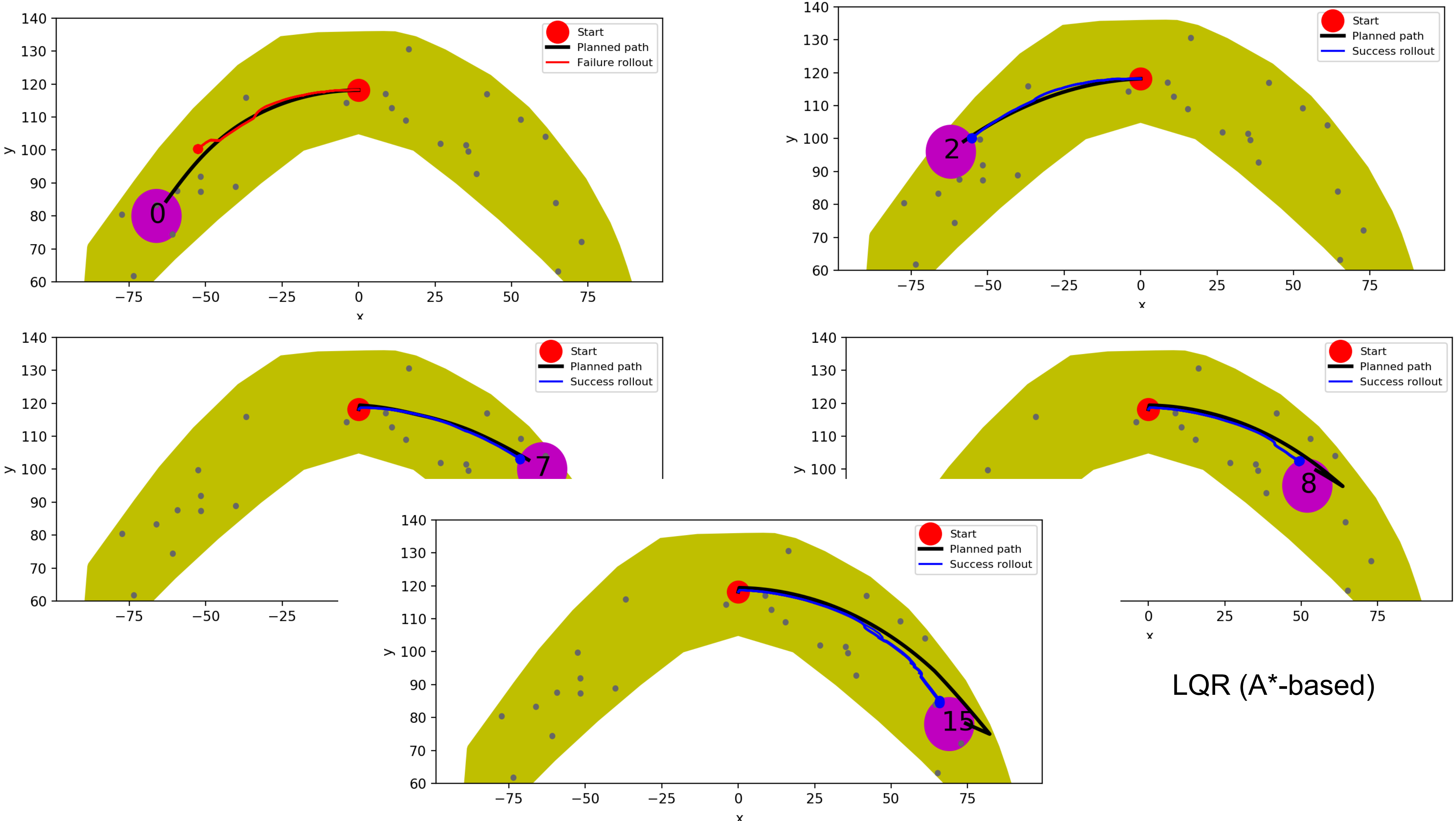
- Resulting policy at i time-steps from the end:

$$u_{H-i} - u_{H-i}^* = K_i (x_{H-i} - x_{H-i}^*)$$

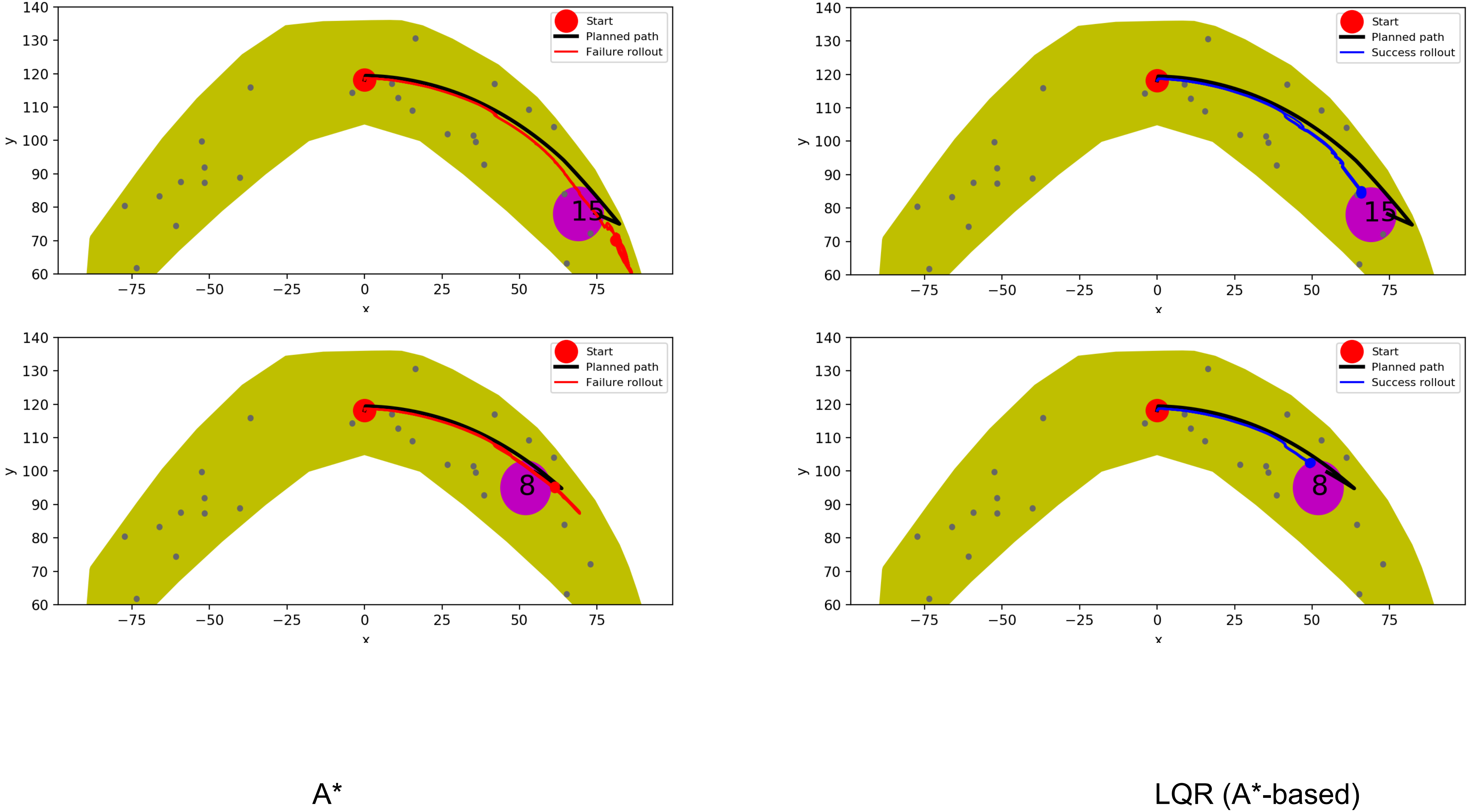
Goal Reach Rate

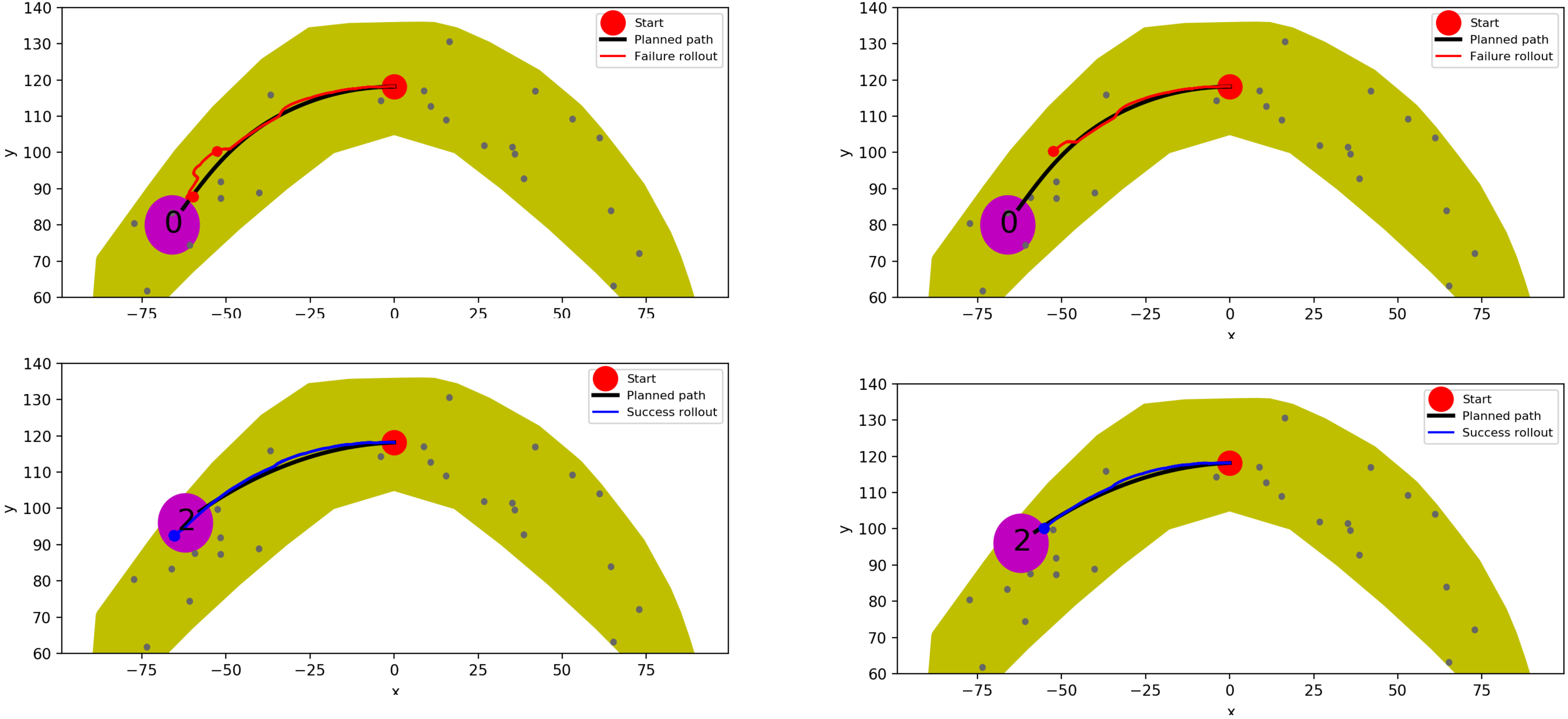
Goal Location	0	2	7	8	15
A*	0%	100%	100%	0%	0%
LQR(A*-based)	0%	100%	100%	100%	100%


Start
Planned path
Failure rollout
Success rollout
0
2
7
8
15
y
x
140
130
120
110
100
90
80
70
60
−75
−50
−25
0
25
50
75
A*


Start
Planned path
Failure rollout
Success rollout
0
2
7
8
15
x
y
140
130
120
110
100
90
80
70
60
−75
−50
−25
0
25
50
75

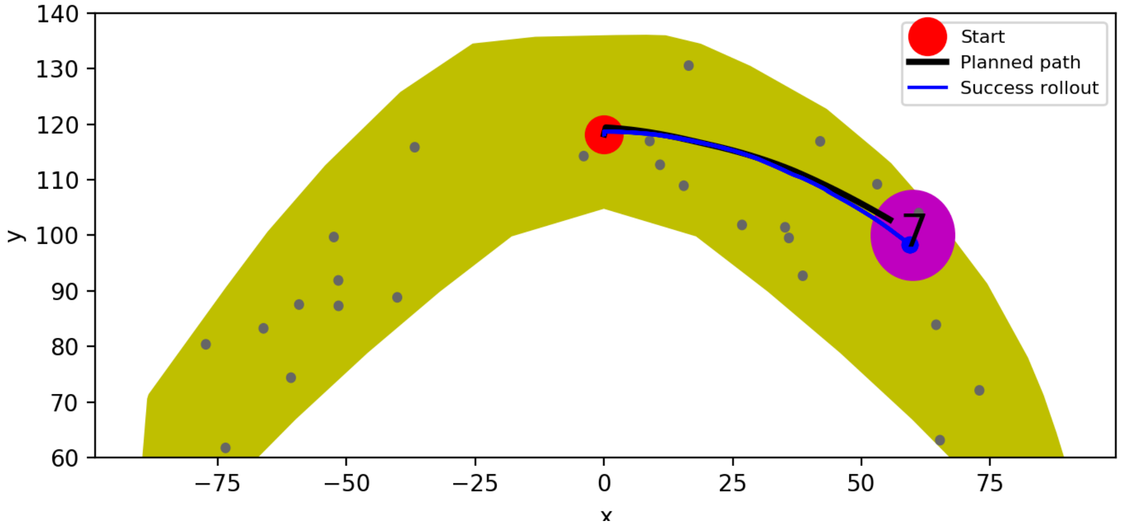
LQR (A*-based)


140
130
120
110
100
90
80
70
60
y
−75
−50
−25
0
25
50
75
x
Start
Planned path
Failure rollout
Success rollout
15
8
A*
LQR (A*-based)

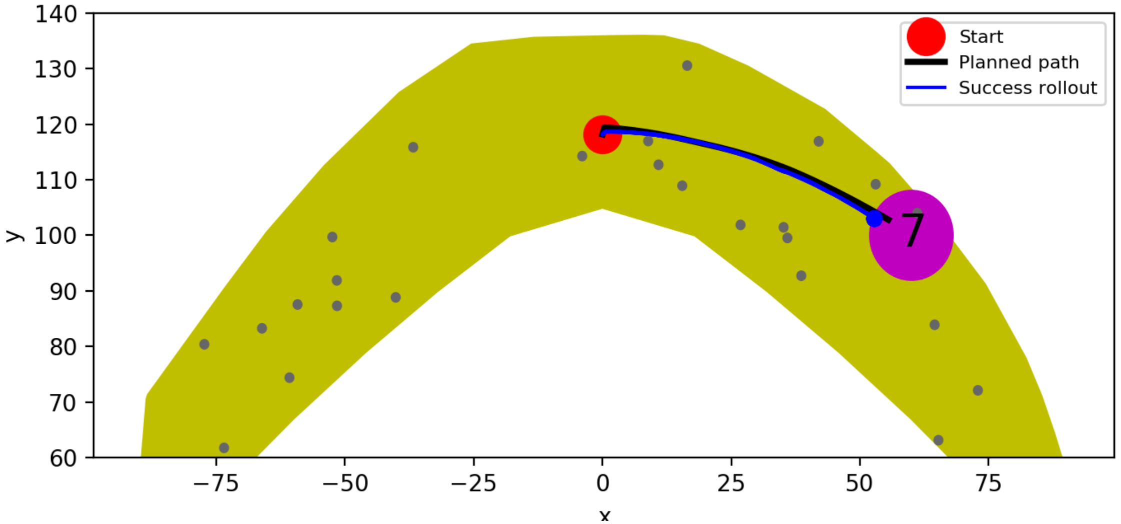

Start
Planned path
Failure rollout
Success rollout
0
2
y
x
140
130
120
110
100
90
80
70
60
−75
−50
−25
0
25
50
75

A*

LQR (A*-based)


Start
Planned path
Success rollout
7
140
130
120
110
100
90
80
70
60
y
−75
−50
−25
0
25
50
75
x

A*


Start
Planned path
Success rollout
7
140
130
120
110
100
90
80
70
60
y
−75
−50
−25
0
25
50
75
x

LQR (A*-based)

Next plans

After we get LQR solutions(Matrix K_i):

1) LQR on other tasks(mujoco)

2) Should I also try closed loop control using PPO (trained from model) on real environment?

3) Learn more theory about conjugate gradient/line search/two-constraints(lambda-eta) optimization. Then, derive new equations, objective function and optimization procedures for our AIP?

4) Implement AIP

AIP Implementation (against TRPO,PPO)

Difference 1 Policy Network:

TRPO/PPO: u_final=pi_theta([x])

AIP: u_final=pi_theta([x, u_controller])

Difference 2 Constraint:

TRPO:

KL(pi_theta_new || pi_theta_old)<epsilon. (CG+Line Search)

PPO:

No constraint

Constraint (KL(pi_theta_new || pi_theta_old)<epsilon) combined into objective function

AIP:

KL(pi_theta_new || pi_theta_old)<epsilon

??KL(pi_theta || controller)<omega??

(Need to derive new equations and objective for optimization?)

Task	Open Loop A* +Rollout	Open Loop PPO + Rollout	Closed Loop PPO	Cloes Loop LQR based on A*	AIP (3 options)
Reacher (0.1% model)	Done + Done	Done + Done	Not yet	Not yet	Not yet
Gazebo Hand (0.1% model)	Done + Done	Done + Not yet	Not yet	Done	Not yet
Acrobot (100% model)	Done + Done	Done + Done	Not yet	Not yet	Not yet
Real Hand (100% model)	Done + Not yet	Done + Not yet	Not yet	Not yet	Not yet