Zoom

Sampling-based Motion Planning for Legged Robots

Yanran Ding, Mengchao Zhang, Haoran Tang

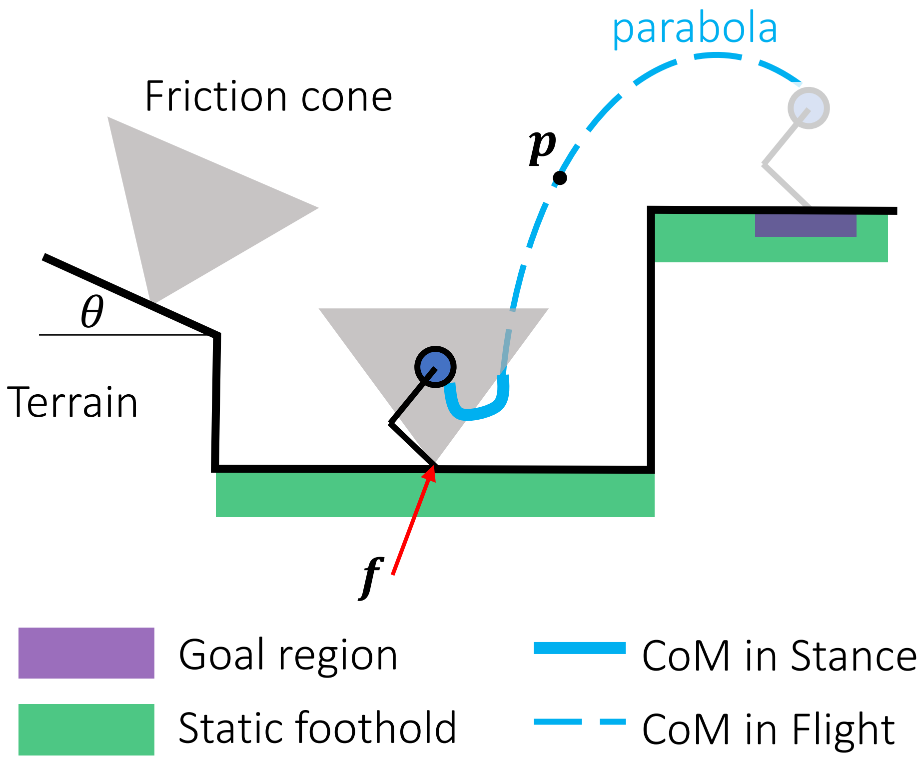
Y M H

Motivation

Zoom

Problem Statement

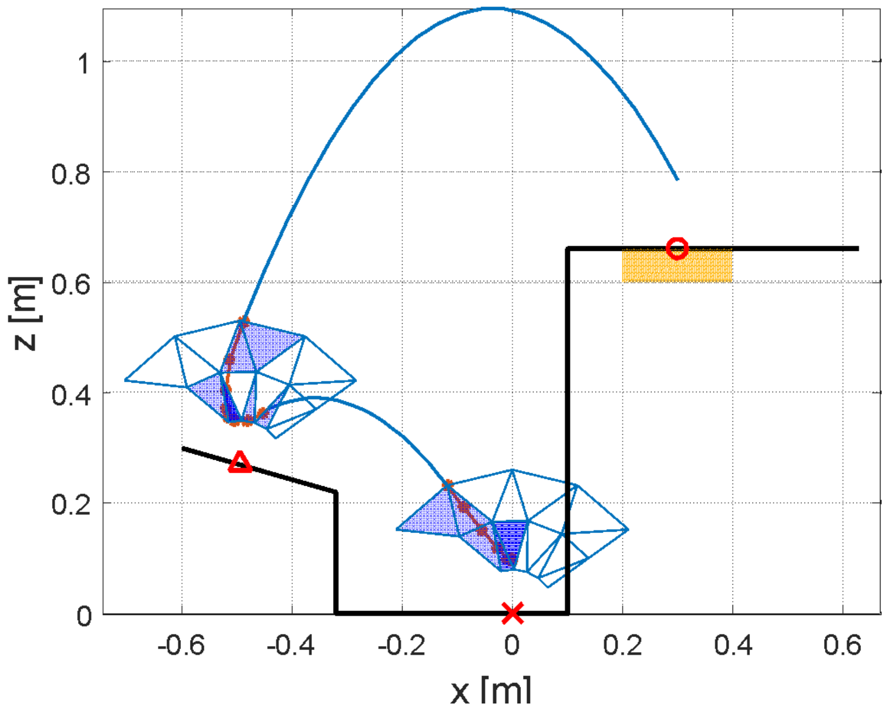
Zoom



- Single leg robot
  - Point-mass
- Reach goal
  - $|x(t_f) - x_g| < \epsilon$
- Subject to constraints
  - $\dot{x}(t) = f(x,u)$
  - $p_c \in E$
  - $x(t) \in \Omega(\mathrm{p_c})$
  - $u(t) \in U(x)$
  - $x(t_i) = x_0$

Existing Method

Zoom



Optimization-based method

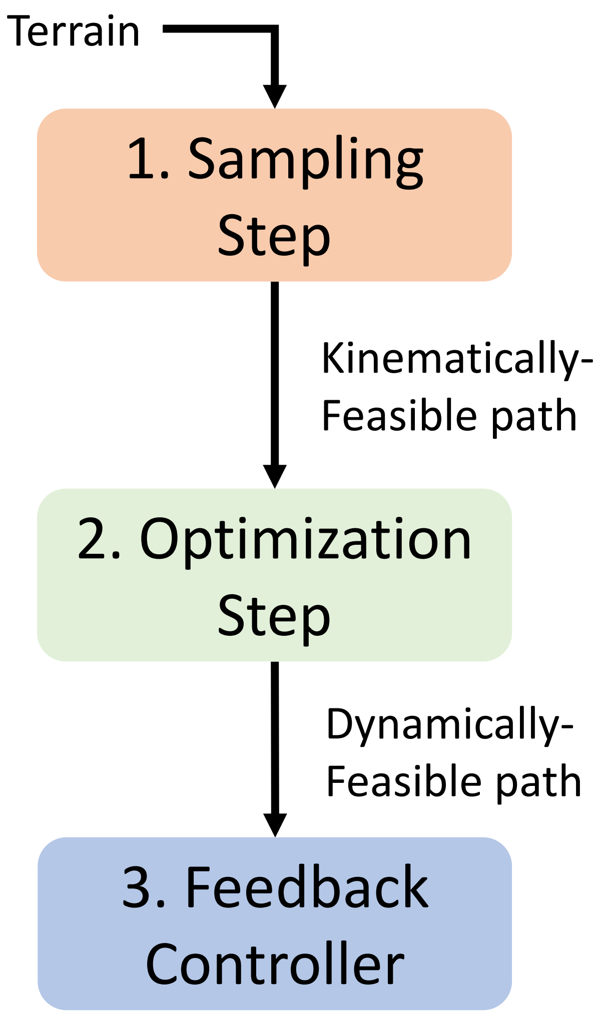
- Pros
  - Respects Dynamics
  - Could handle control constraint
- Cons
  - Long solve time for large N_{step}
  - Collision detection expensive

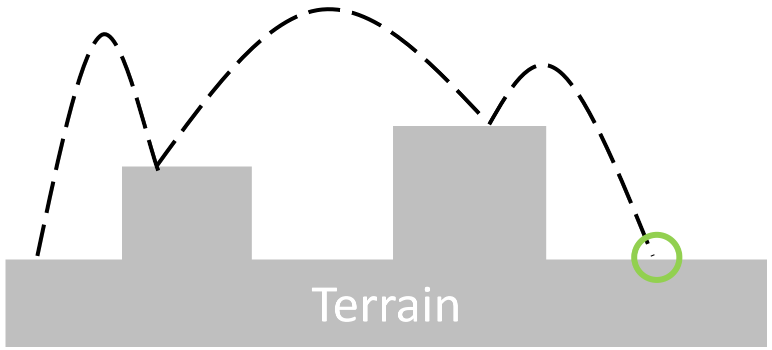
Trajectory Optimization (TO)

Mixed-Integer Convex Program (MICP)

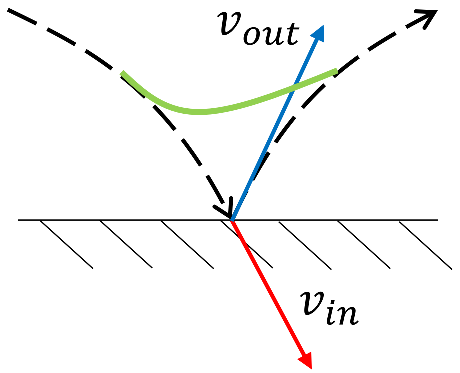
Proposed Framework

Zoom





- Path as a sequence of parabola
- Effective for complex terrain
- Inexpensive to check collision



- Use Optimization to ‘Fuse’ parabola
- Solve an optimization problem
- Solve time scales linearly w.r.t. N_{step}

Sampling Step

Zoom

Terrain

1. Sampling Step

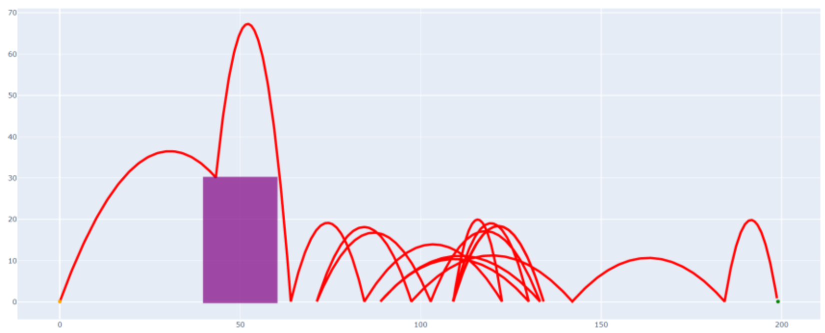
Kinematically-Feasible path

2. Optimization Step

Dynamically-Feasible path

3. Feedback Controller

- Sample Space: Task space
- Data Structure: Path
- Sample Strategy: Reachability guided
- Heuristics
  - Sample in reachable region of current point P_k
  - Sample point not too close to P_k nor P_{k-1}
  - Not too close to obstacle
  - If no obstacle between P_k and goal, always jump towards the goal
  - …
- Find parabola
- Post-processing



Sampling Step

Zoom

Terrain

1. Sampling Step

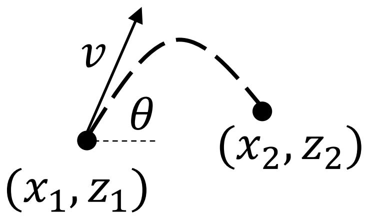
Kinematically-Feasible path

2. Optimization Step

Dynamically-Feasible path

3. Feedback Controller

- Sample Space: State space
- Data Structure: Path
- Sample Strategy: Reachability guided
- Heuristics



- Find parabola

$$
\begin{aligned}
&\mathit{minimize}_{v,\theta,t} \quad t\\
&\mathit{subject\ to}\\
&\quad x_2 = x_1 + vC_\theta t\\
&\quad z_2 = z_1 + vS_\theta t - \frac{1}{2}gt^2\\
&\quad -\mu < \frac{vC_\theta}{vS_\theta - gt} < 0\\
&\quad \frac{\pi}{2} - \arctan(\mu) < \theta < \frac{\pi}{2}\\
&\quad v_{min} < v < v_{max}
\end{aligned}
$$

Formed and solved as a small-scale nonlinear programming (NLP)

- Post-processing

Sampling Step

Zoom

Terrain

1. Sampling Step

Kinematically-Feasible path

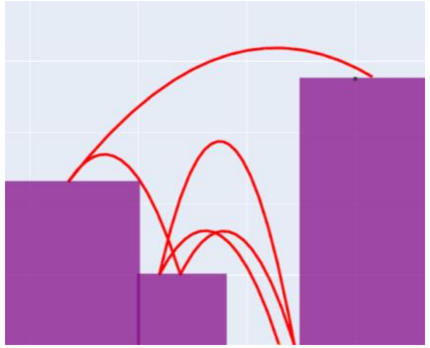
2. Optimization Step

Dynamically-Feasible path

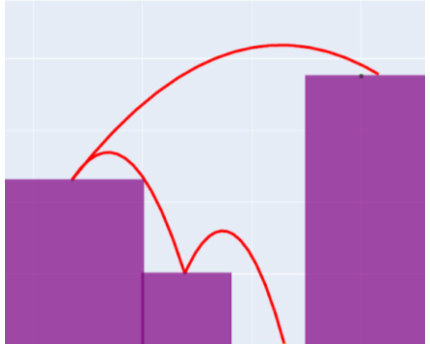
3. Feedback Controller

- Sample Space: State space
- Data Structure: Path
- Sample Strategy: Reachability guided
- Heuristics
- Find parabola
- Post-processing (discard small steps)

Before



After

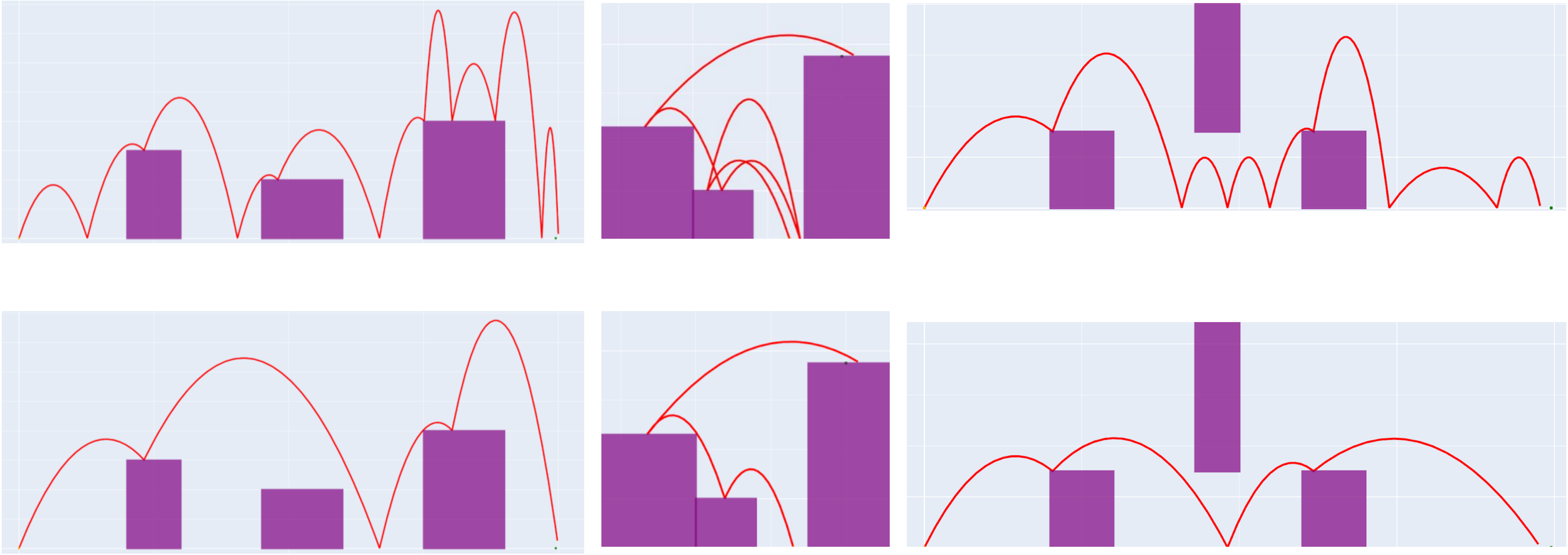


Post Processing

Zoom

before

after

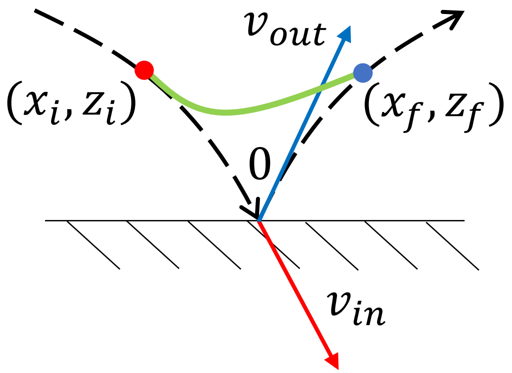


Optimization Step

Zoom

- Optimize local trajectory
- Solve for trajectory given v_{in} and v_{out}
- Problem:

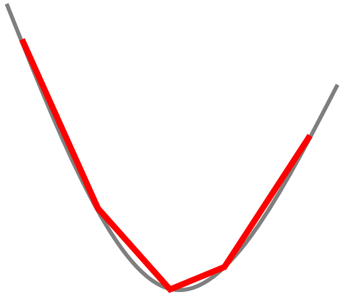
find control trajectory that drives the initial condition on inward parabola to the final condition on the outward parabola



velocity $\quad \dot{x} = v_x, \dot{z} = v_z - g \cdot \frac{x}{\mathrm{v}_x}$

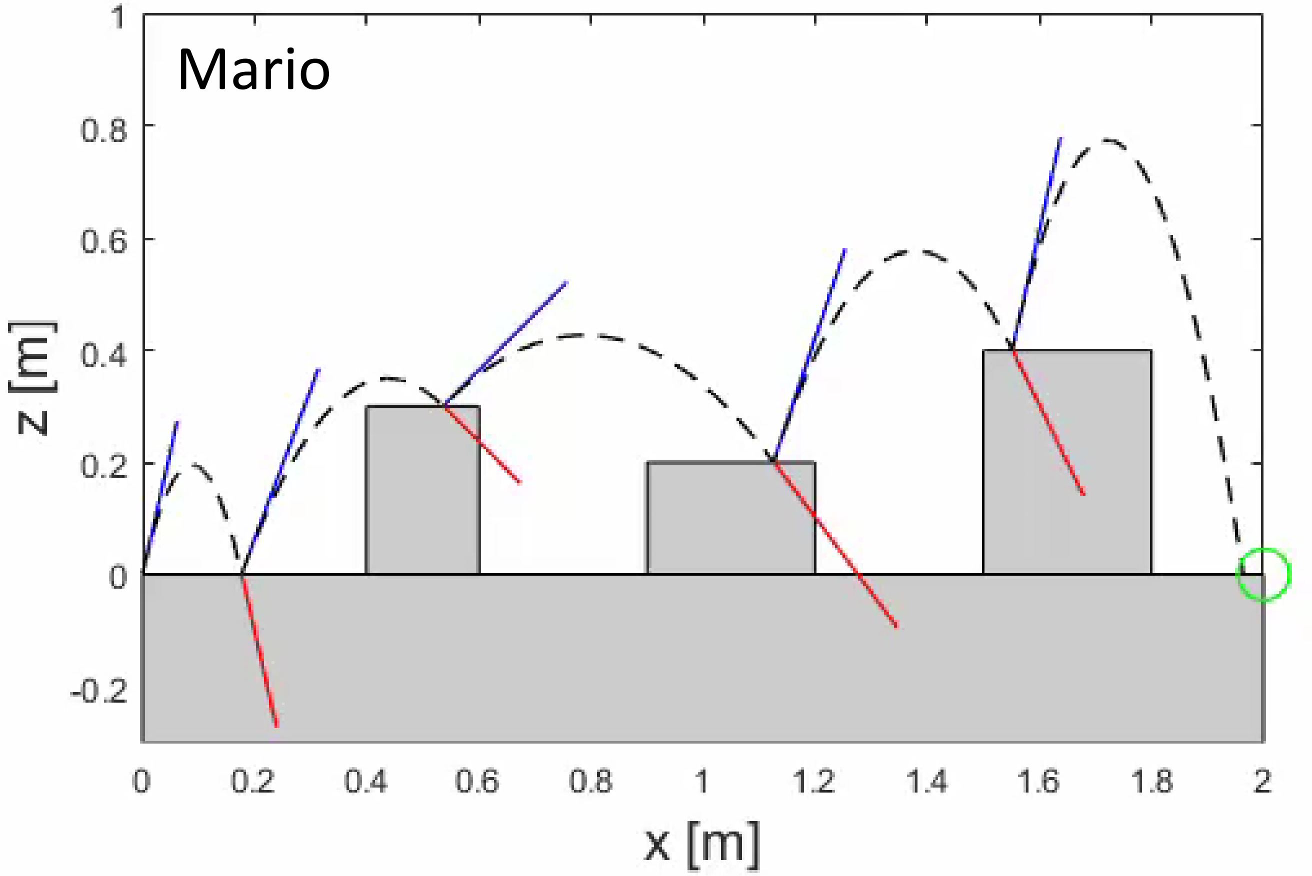
position $\quad z = v_z \frac{x}{\mathrm{v}_x} - \frac{1}{2} g \left(\frac{x}{\mathrm{v}_x}\right)^2$

Symmetric for inward/outward parabola



Position constraint is nonconvex → mixed-integer convex relaxation

Results

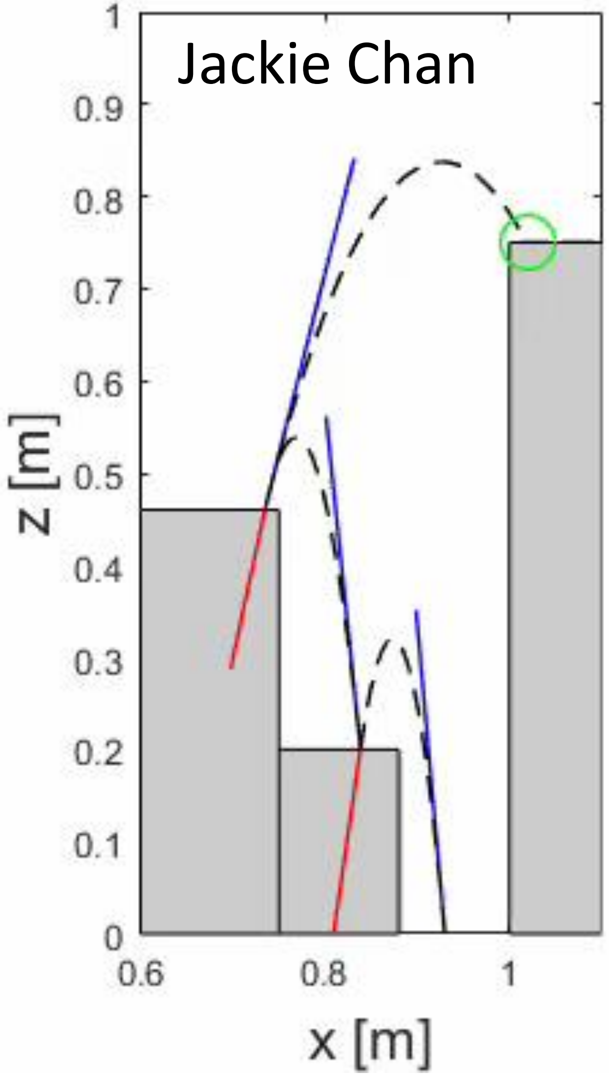


Zoom

Solve time

- Sampling step [s]: 0.1
- Optimization step [s]: 24
  - 1.2, 1.8, 3.5, 12.2, 6.5

Results



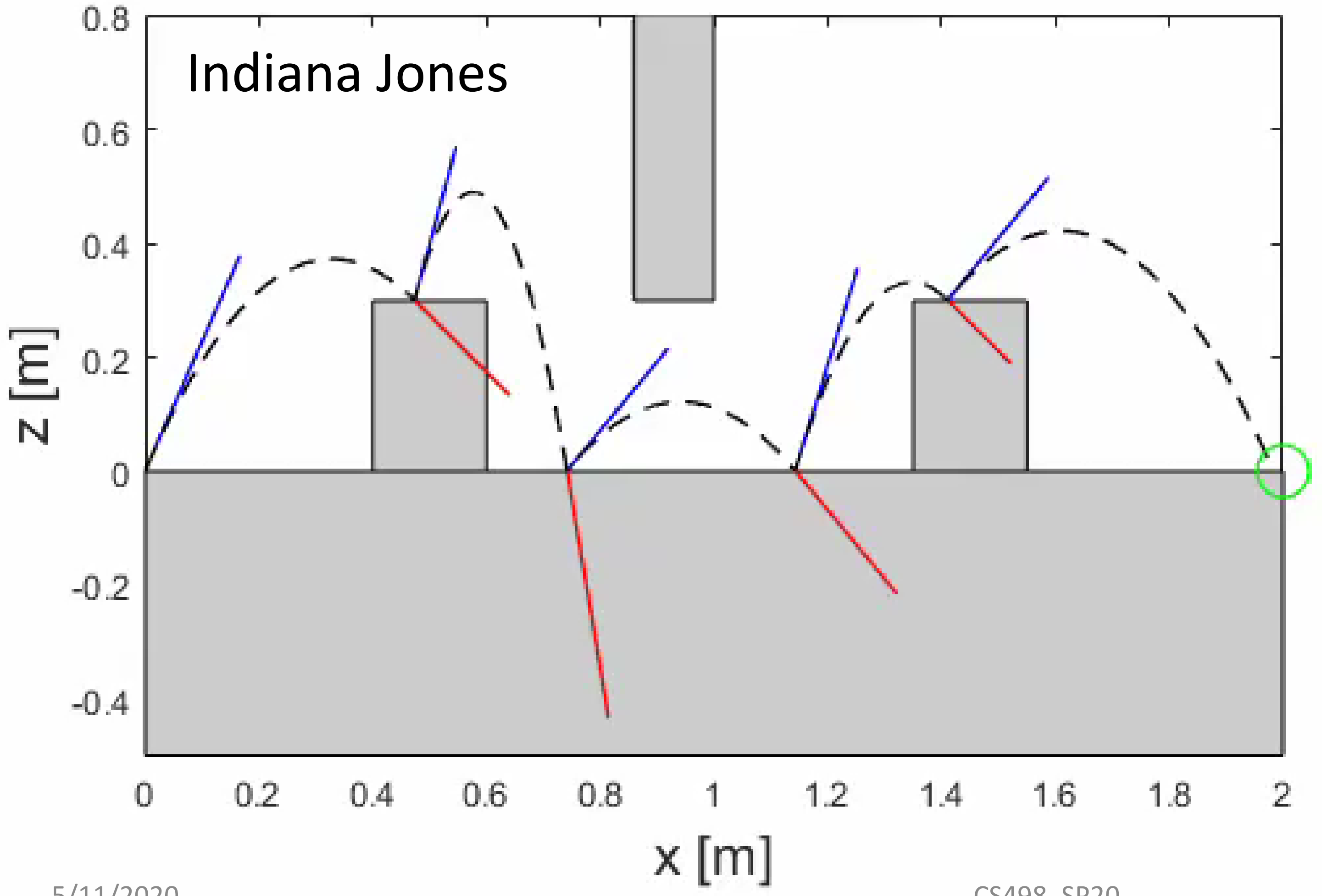
Zoom

Solve time

- Sampling step [s]: 0.3
- Optimization step [s]: 3.6
  - 1.6, 1.3, 0.7

Results

Zoom



Solve time

- Sampling step [s]: 0.3
- Optimization step [s]: 13.8
  - 1.0, 0.5, 10.0, 1.1, 1.2

Conclusion and Future Work

Zoom

Conclusion

- We present a Sampling-Optimization Hierarchical Motion Planning for Legged Robots
- This method exploits the advantages of both methods to produce solution to complex problem with low computational cost

Future Work

- Allow for more complex terrain/ moving obstacles
- Integrate with Feedback controller to handle uncertainty

Zoom

Contribution

Yanran Ding Y

- In charge of optimization step
- Code the optimization in MATLAB
- Brainstorm ideas

Mengchao Zhang M

- In charge of Sampling step
- Code the sampling step in Python
- Generate the test scenarios

Haoran Tang H

- Post-processing of the sampled parabola

Zoom

Q & A

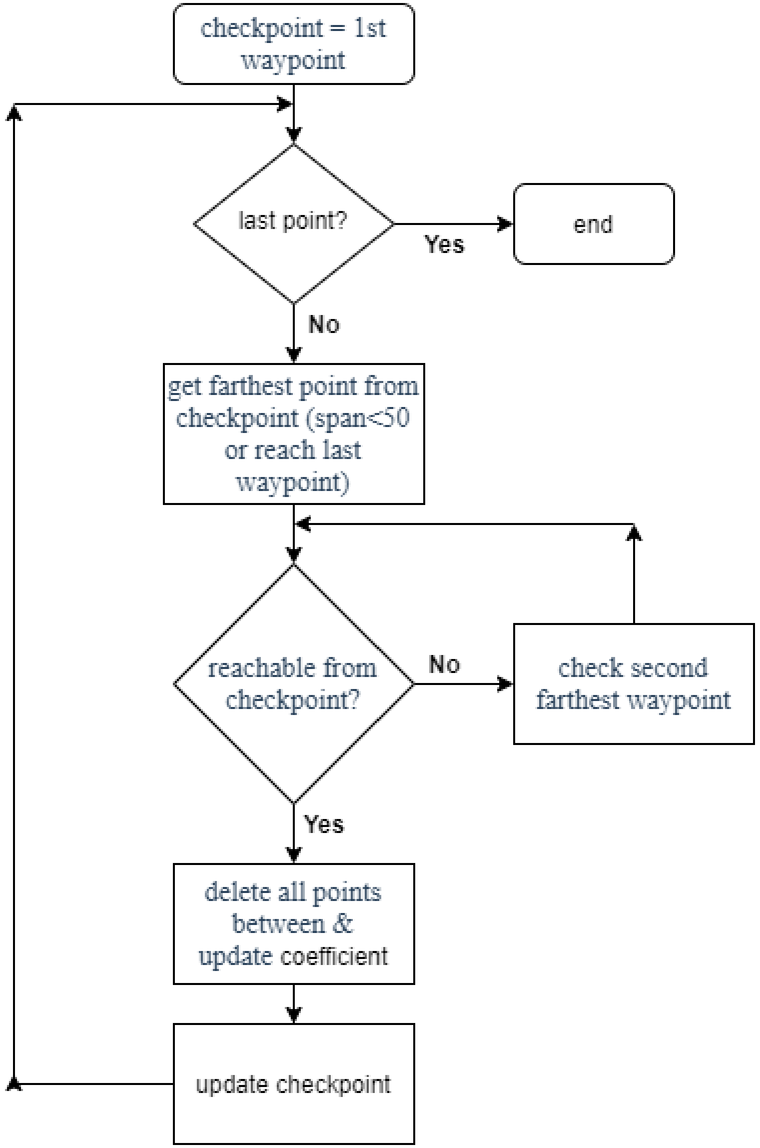
Zoom

Backup Slides

Post Processing

Forward optimizing:
pro: straightforward logic
con: harder to update (need to reconstruct list and calculate new iteration index)----solved by linked list

Backward optimizing:
pro: no linked list, faster and more efficient (append waypoint at the tail)
con: harder to design and develop



Zoom

Sampling Step

Zoom

Terrain

1. Sampling Step

Kinematically-Feasible path

2. Optimization Step

Dynamically-Feasible path

3. Feedback Controller

- Sample Space: State space
- Data Structure: Path
- Sample Strategy: Reachability guided
- Heuristics
- Find parabola
- Post-processing (discard small steps)

Methods	Pro	Con
Forward optimizing	Logic straightforward	More difficult to update
Backward optimizing	More efficient	More difficult to design