Safe and Efficient robot control

Combining learning and trajectory optimization

Andrea Del Prete

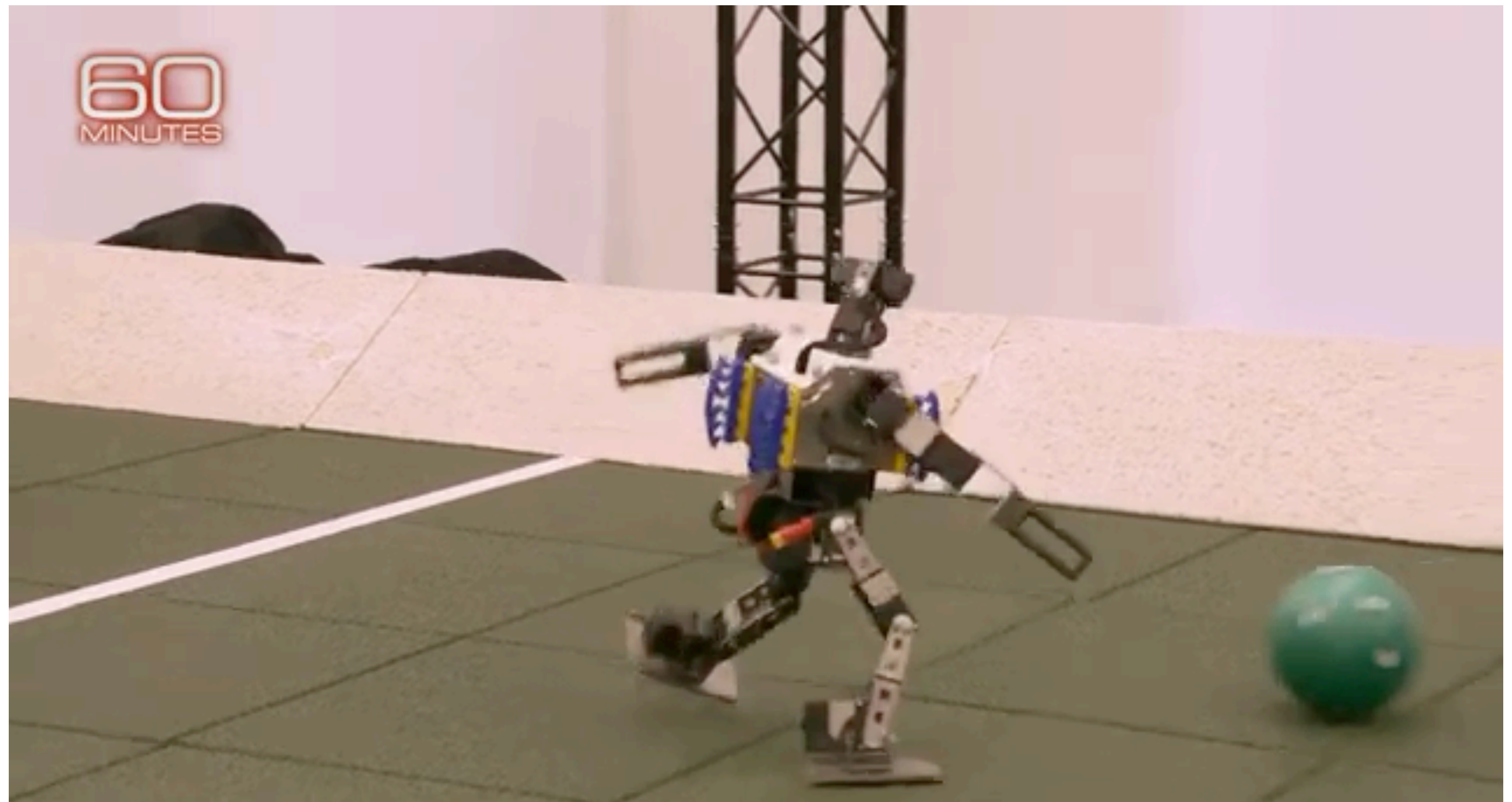
UNIVERSITY OF TRENTO

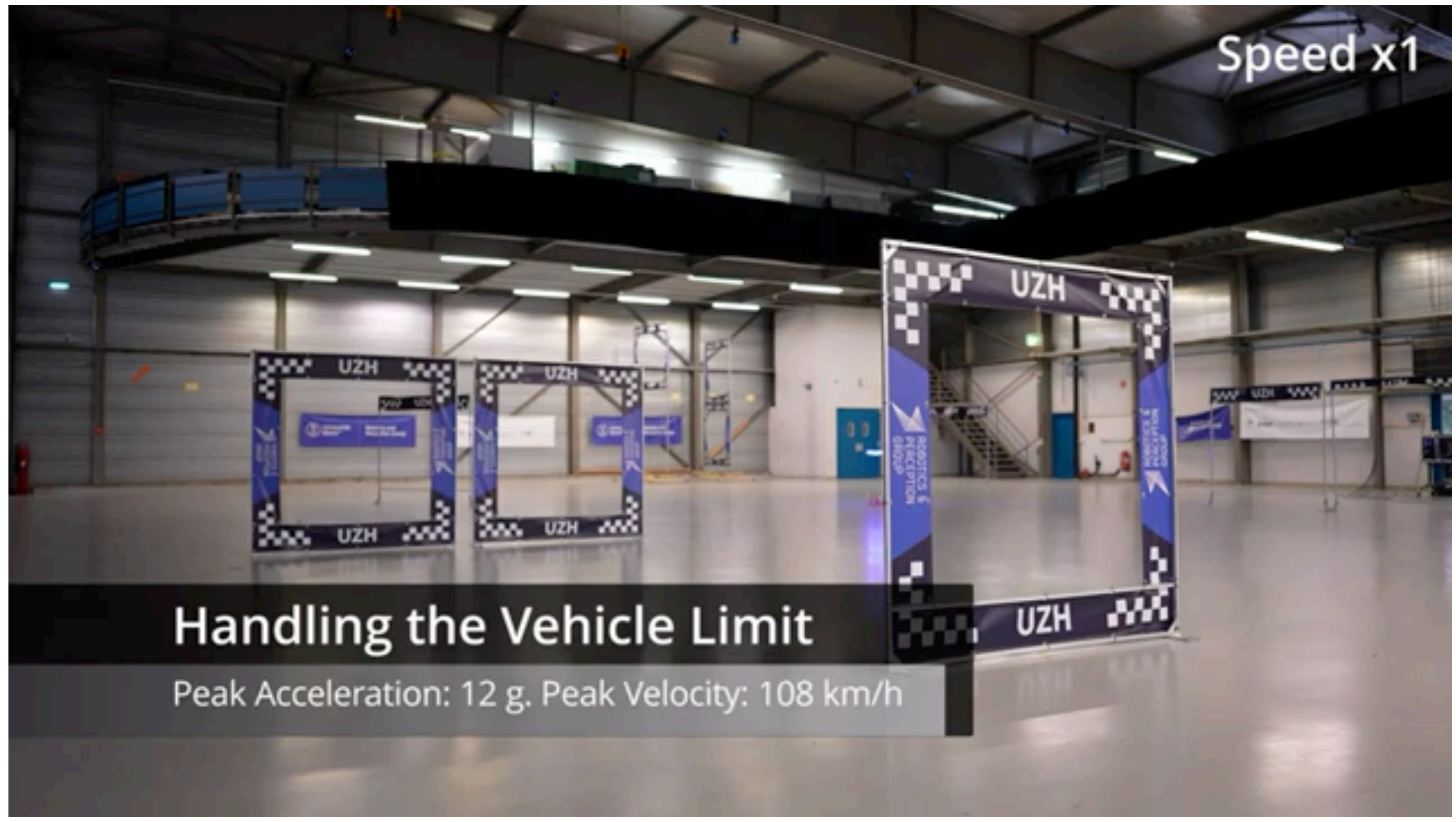
Is there anything RL cannot do?

Is Trajectory Optimization bound to die?

Lee, Hwangbo, Wellhausen, Koltun, Hutter (2020). Learning quadrupedal locomotion over challenging terrain. Science Robotics

Haarnoja, T., Moran, B., Lever, G., Huang, S. H., Tirumala, D., Wulfmeier, M., … Heess, N. (2023). Learning Agile Soccer Skills for a Bipedal Robot with Deep Reinforcement Learning





Song, Romero, Müller, Koltun, Scaramuzza, (2023). Reaching the limit in autonomous racing: Optimal control versus reinforcement learning. Science Robotics

The issues with RL

My two cents

Poor efficiency

- Data efficiency
- Energy efficiency
- Time efficiency

Poor safety

- No explicit constraints
- No guarantees
- Safety-critical applications

Can we use ideas from Trajectory Optimization to make RL safe and efficient?

Safe and Efficient robot control

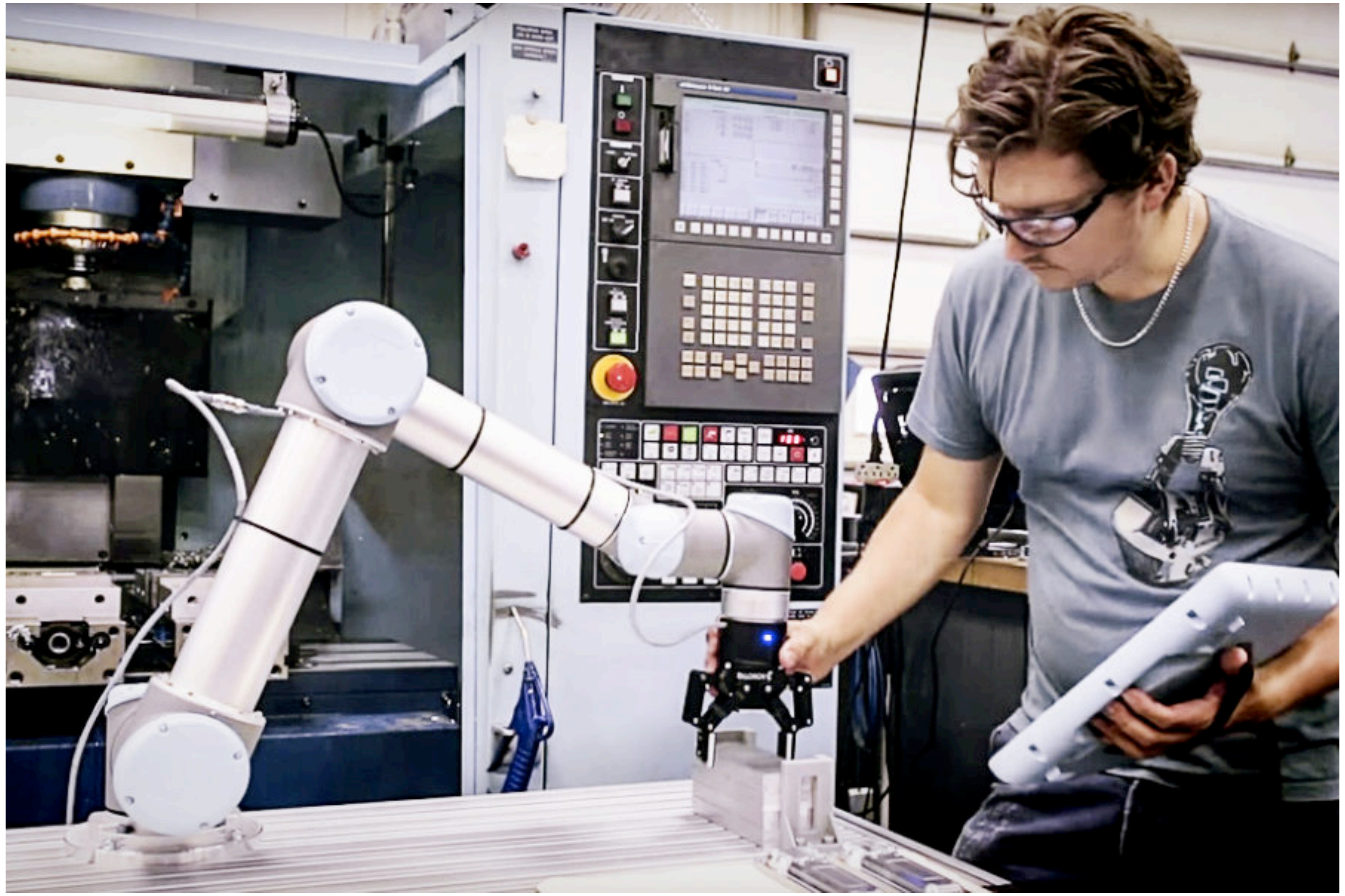
Combining learning and trajectory optimization

Andrea Del Prete

UNIVERSITY OF TRENTO

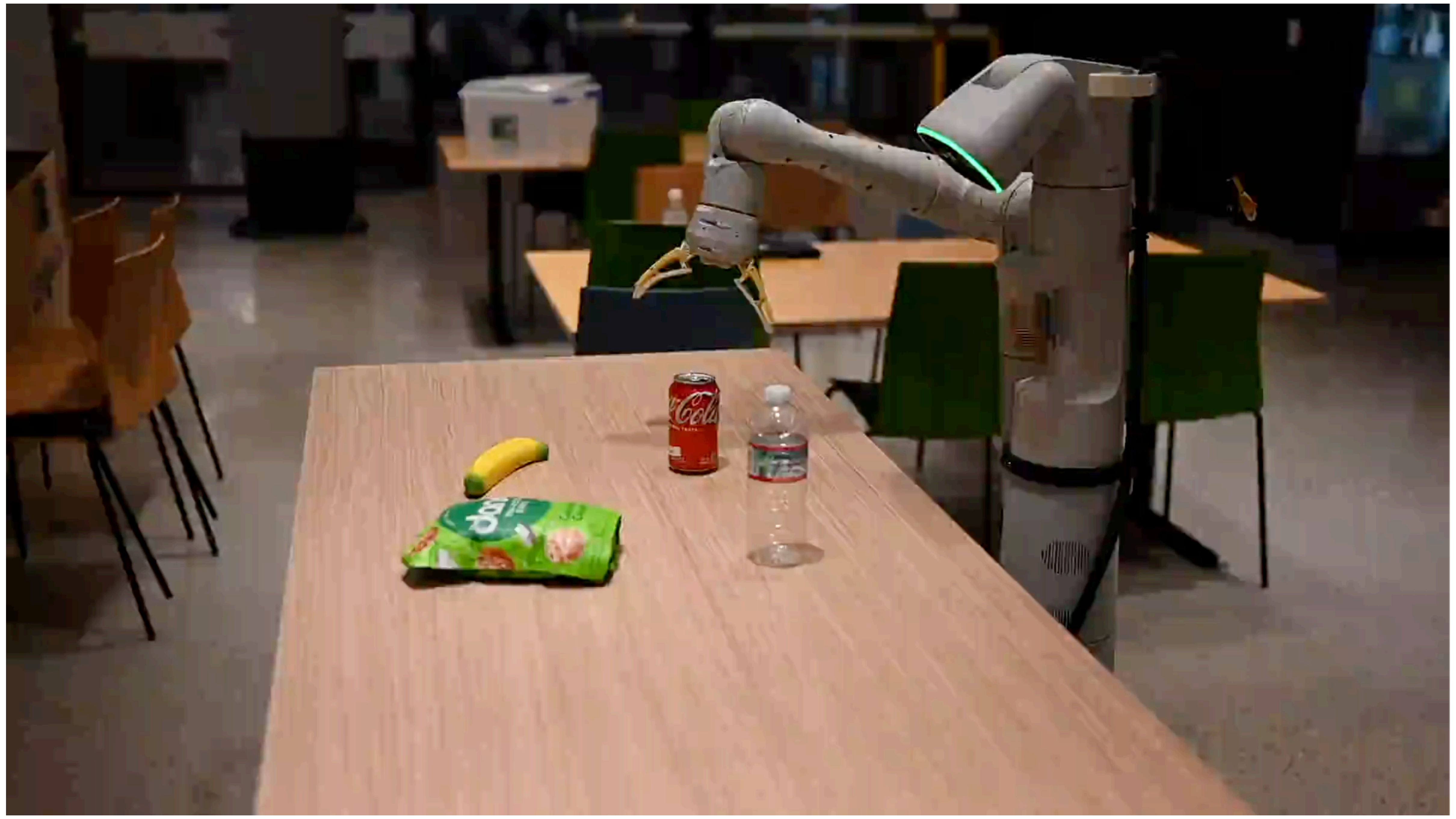
Why Safety?

Today
Human-Robot Collaboration in Industry



https://www.therobotreport.com/manufacturing/ria-osha-robot-safety/

Tomorrow
Black-box Data-Driven Control Policies



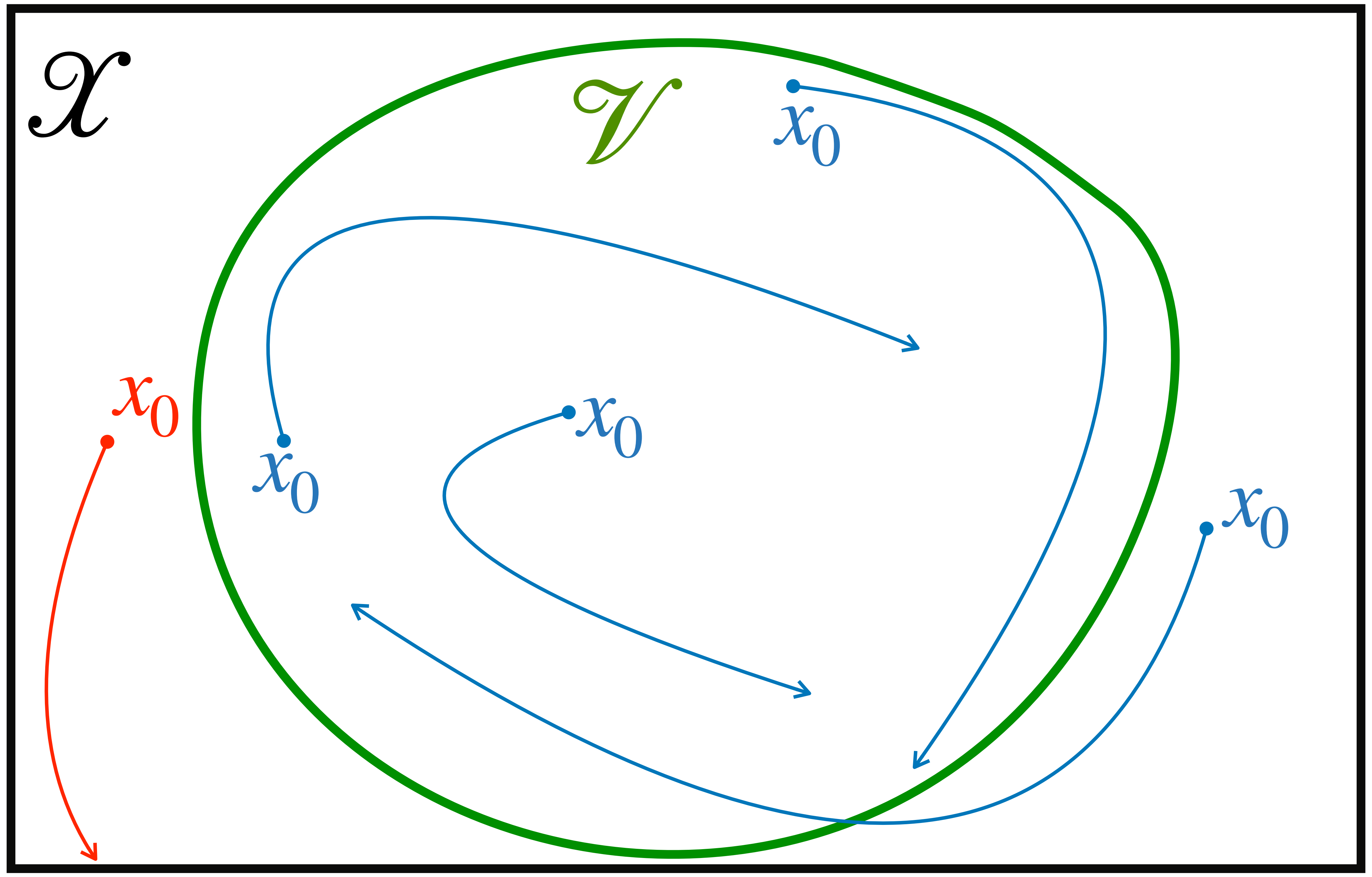
Zitkovich, Brianna, et al. "Rt-2: Vision-language-action models transfer web knowledge to robotic control." Conference on Robot Learning. PMLR, 2023.

Safety via Control Invariant Sets

Constrained discrete-time dynamical system:

$$x_{i+1} = f(x_i, u_i) \qquad x \in \mathcal{X}, \quad u \in \mathcal{U}$$

$\mathcal{V} \subseteq \mathcal{X}$ is a control invariant set ⟷ Once x is in $\mathcal{V}$, it can remain in $\mathcal{V}$



Recursive Feasibility

Model Predictive Control (MPC)

Using a CIS $\mathscr{V}$ as terminal set ensures recursive feasibility in MPC

$$
\begin{aligned}
\underset{\{x_i\}_0^N,\{u_i\}_0^{N-1}}{\text{minimize}} \quad & \sum_{i=0}^{N-1} \ell_i(x_i, u_i) + \ell_N(x_N) \\
\text{subject to} \quad & x_0 = x_{init} \\
& x_{i+1} = f(x_i, u_i) \qquad i = 0 \ldots N-1 \\
& x_i \in \mathcal{X}, u_i \in \mathcal{U} \qquad i = 0 \ldots N-1 \\
& \boxed{x_N \in \hat{\mathcal{V}}}
\end{aligned}
$$

What if the terminal set is an approximation of a CIS $\hat{\mathscr{V}} \approx \mathscr{V}$?

→

MPC problem can become unfeasible using $\hat{\mathscr{V}}$ instead of $\mathscr{V}$!

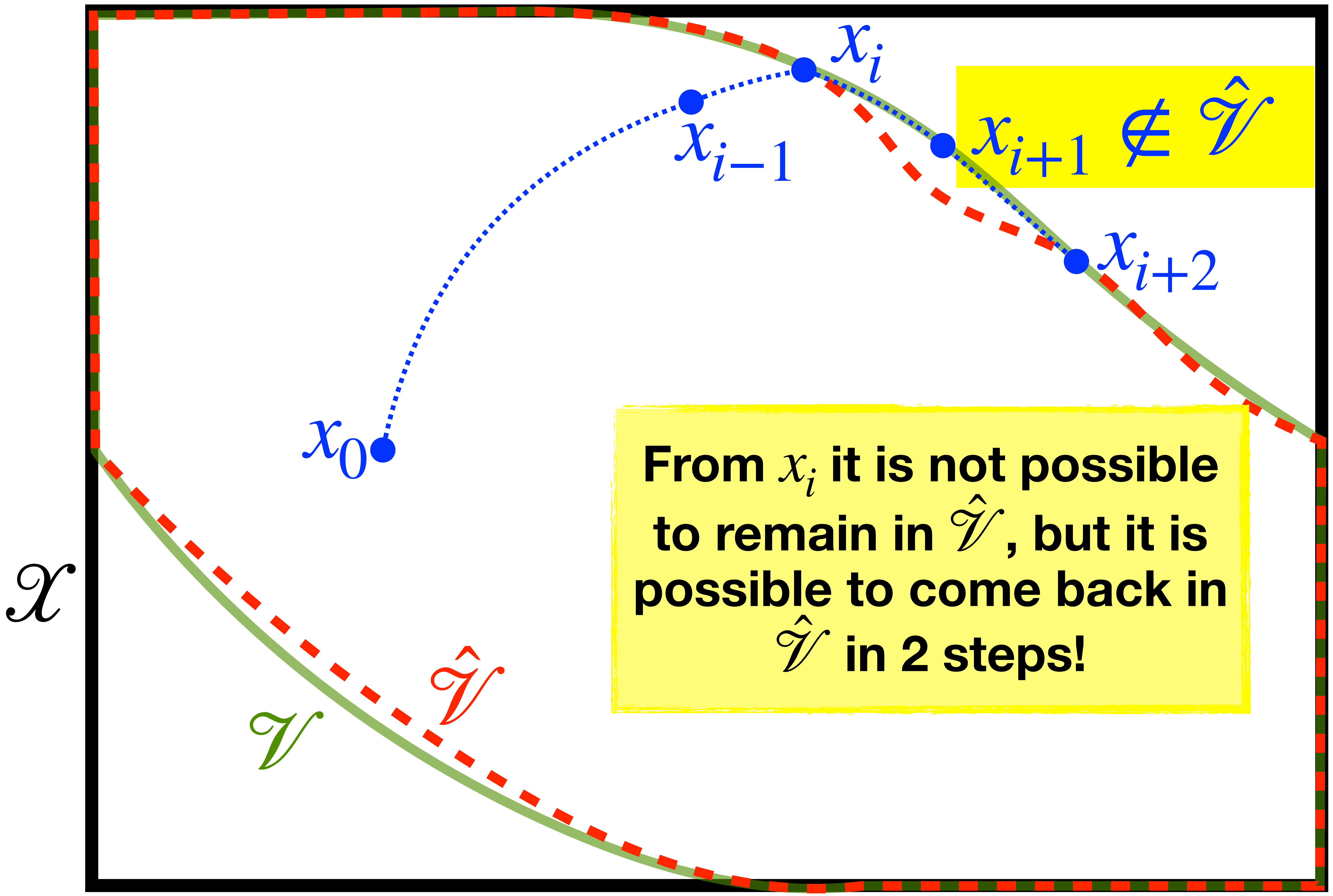
Beyond Control Invariant Sets

- CIS are unknown for nonlinear systems
- Numerical approximation techniques exist, however:
  - They are computationally demanding (curse of dimensionality)
  - A numerical approximation of a CIS is **not** a CIS
    - → all safety guarantees are lost!

Do we really need Control Invariant Sets to ensure safety?

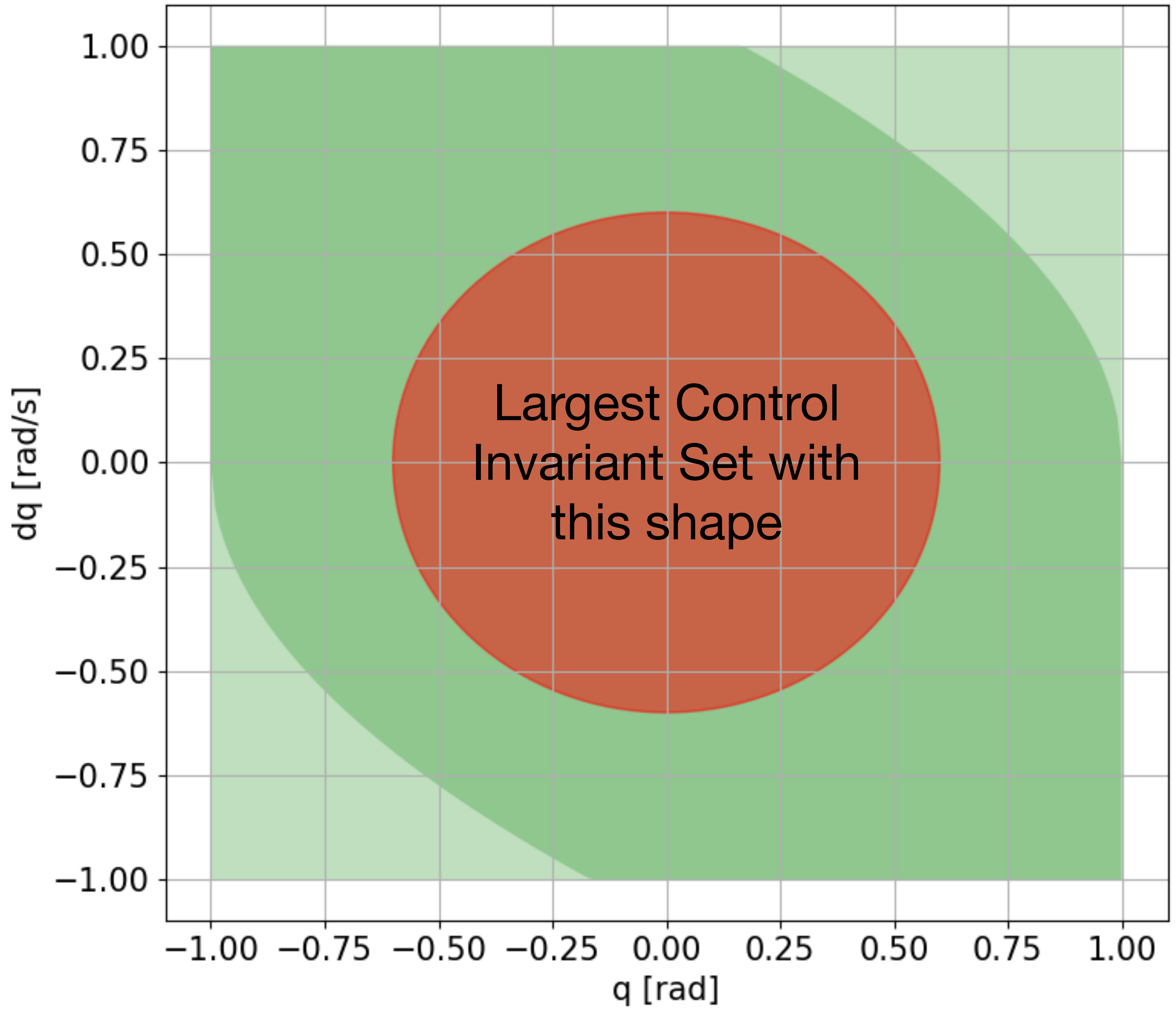
N-Step Control Invariant Set

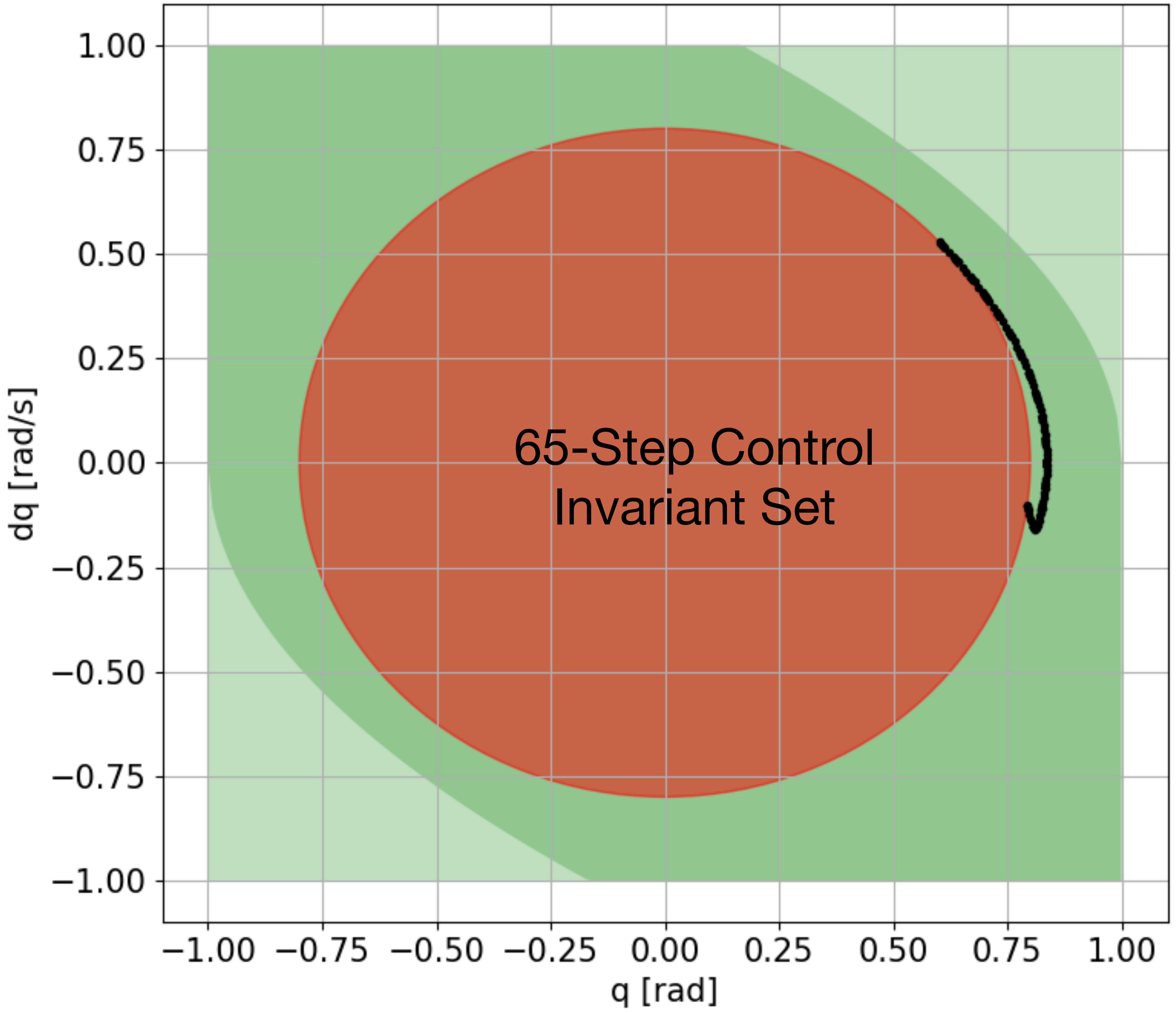
- $\hat{\mathscr{V}}$ is an N-Step CIS iff:
  - For every $x_0 \in \hat{\mathscr{V}}$ it is possible to have $x_k \in \hat{\mathscr{V}}$ for some $k \in [1,N]$
- Weaker condition than classic control invariance
- Possible to guarantee safety with novel MPC schemes



N-Step Control Invariant Sets

Double integrator - Circular shape

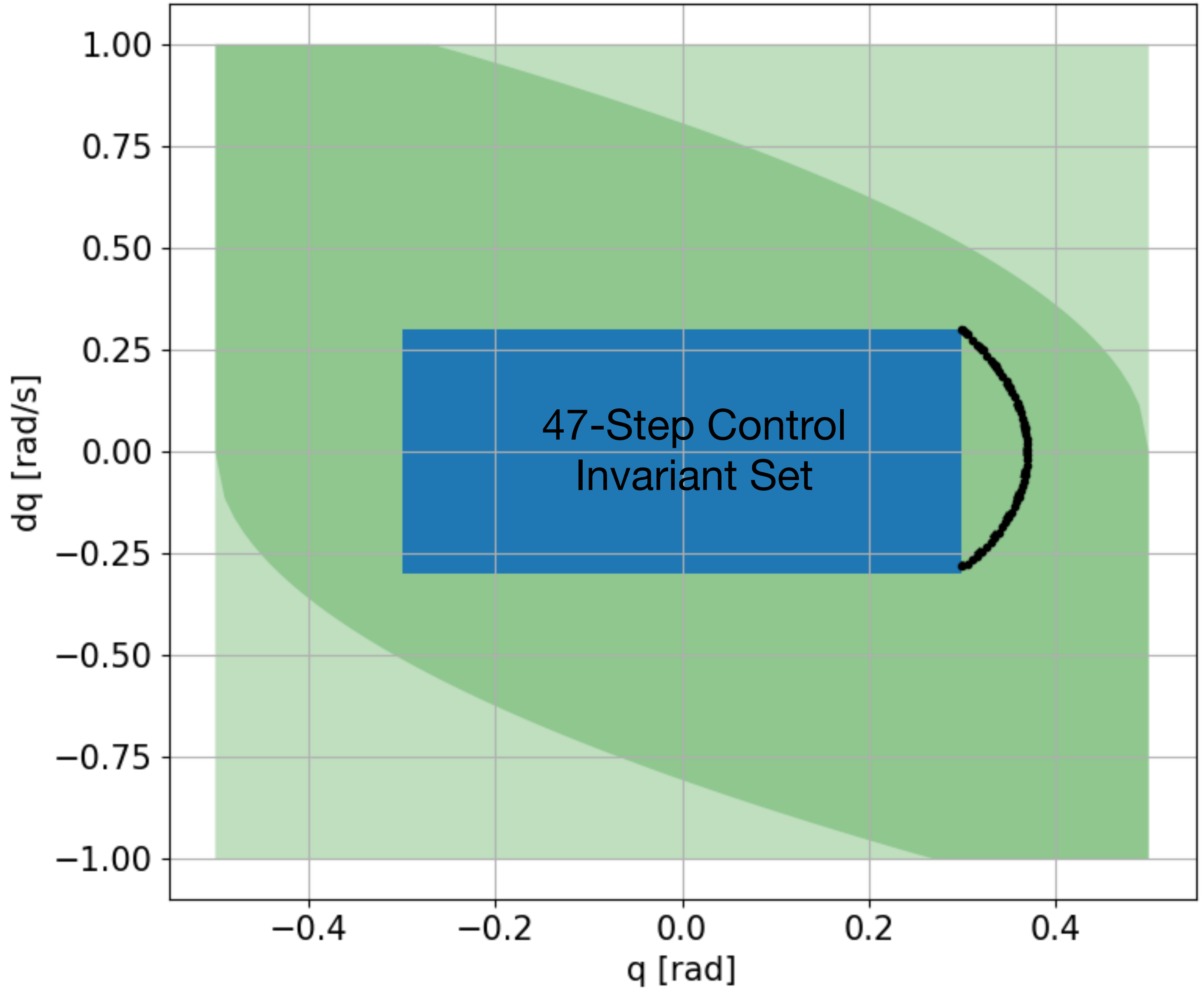




N-Step Control Invariant Set

Double integrator - Rectangular shape

The exists no Control Invariant Set with a rectangular shape!



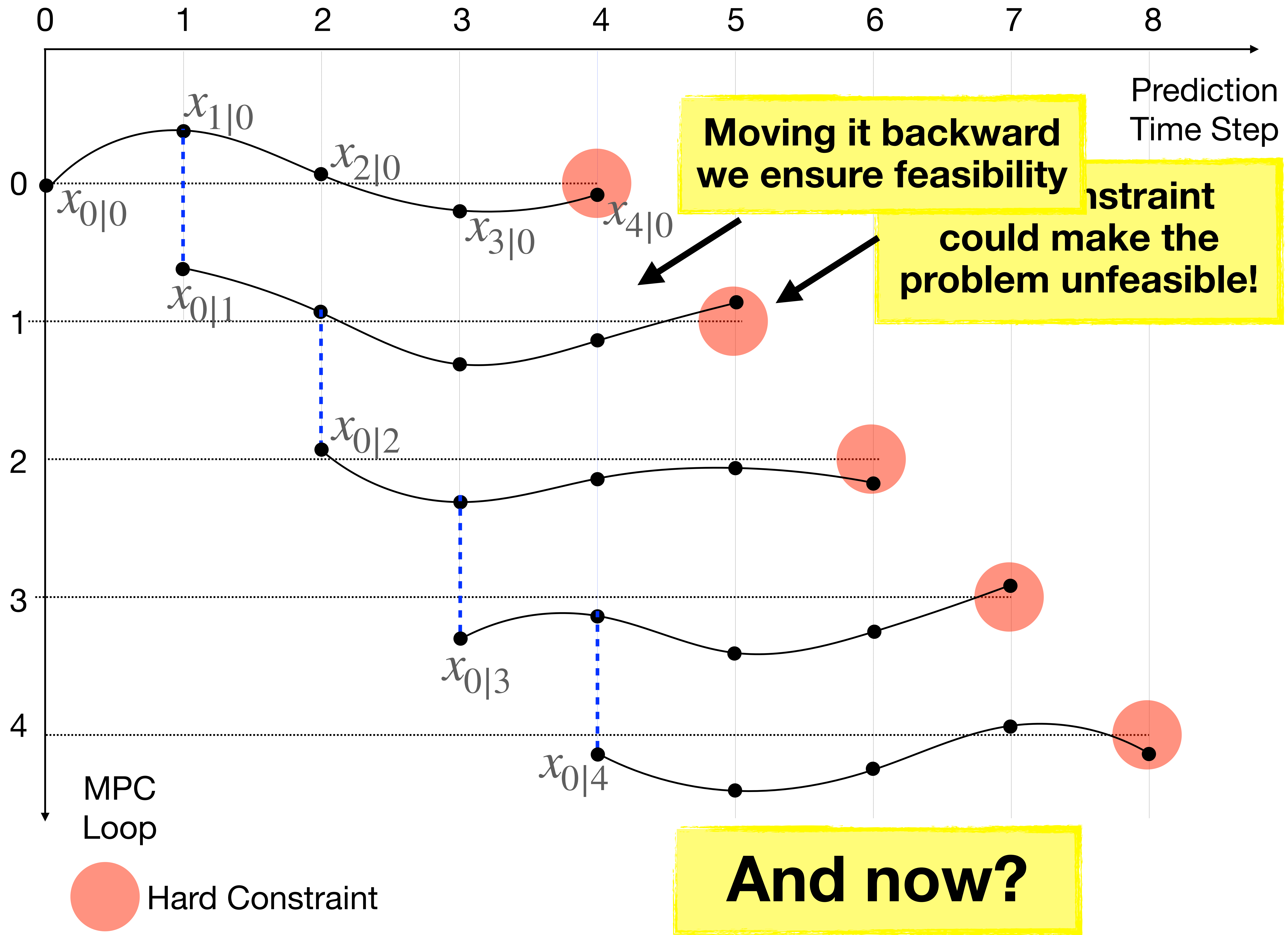
UNIVERSITY OF TRENTO

Receding-Constraint Model Predictive Control

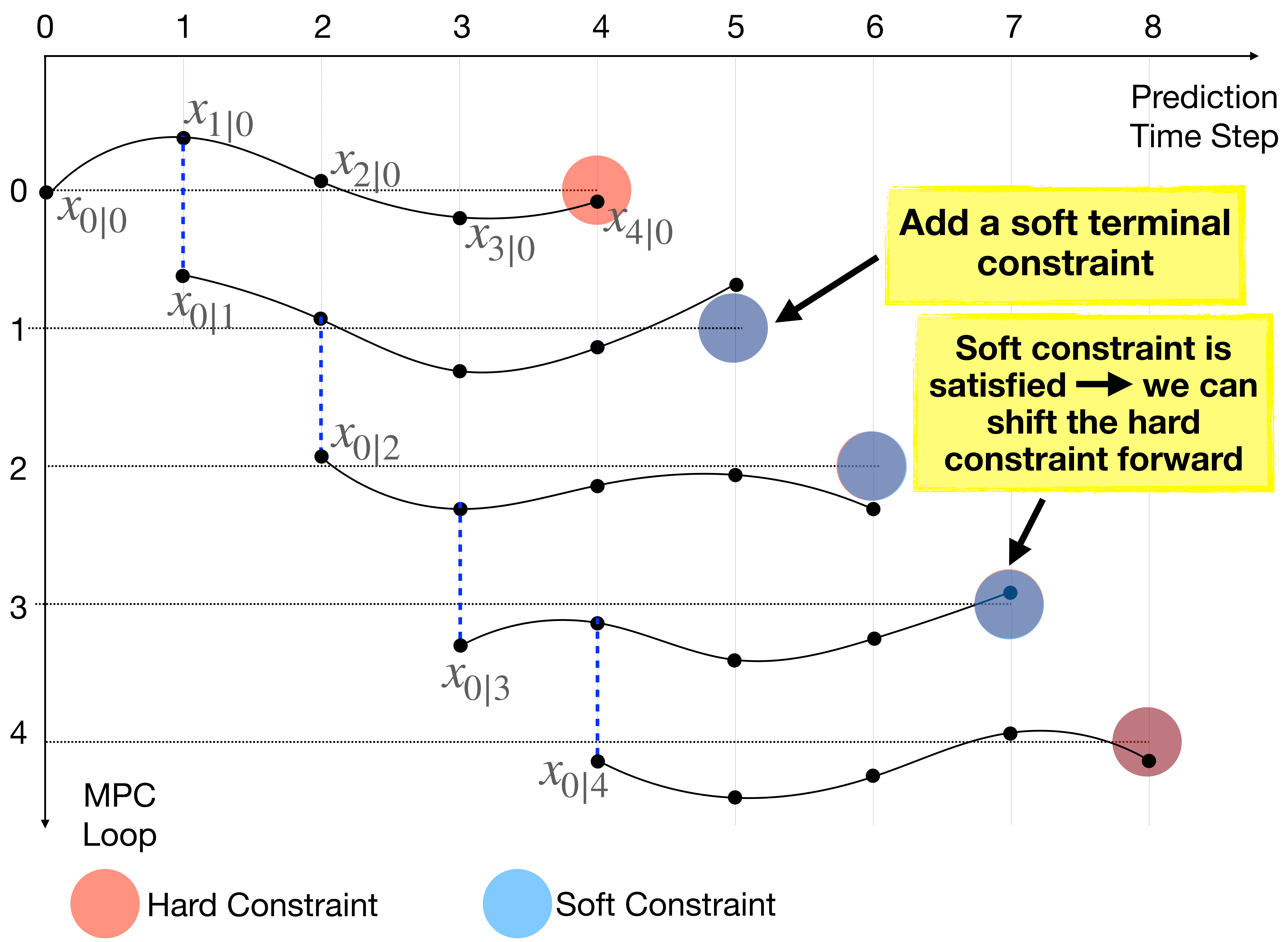
Gianni Lunardi
Asia La Rocca
Matteo Saveriano
Andrea Del Prete



Lunardi, La Rocca, Saveriano, Del Prete (2024). Receding-Constraint Model Predictive Control using a Learned Approximate Control-Invariant Set. IEEE ICRA.



0
1
2
3
4
5
6
7
8
Prediction
Time Step
MPC
Loop
$x_{0|0}$
$x_{1|0}$
$x_{2|0}$
$x_{3|0}$
$x_{4|0}$
$x_{0|1}$
$x_{0|2}$
$x_{0|3}$
$x_{0|4}$
Moving it backward we ensure feasibility
nstraint could make the problem unfeasible!
And now?
Hard Constraint


0
1
2
3
4
5
6
7
8
Prediction Time Step
0
1
2
3
4
MPC Loop
$x_{0|0}$
$x_{1|0}$
$x_{2|0}$
$x_{3|0}$
$x_{4|0}$
$x_{0|1}$
$x_{0|2}$
$x_{0|3}$
$x_{0|4}$
Add a soft terminal constraint
Soft constraint is satisfied ⟶ we can shift the hard constraint forward
Hard Constraint
Soft Constraint

UNIVERSITY OF TRENTO

Parallel-Constraint Model Predictive Control

Elias Fontanari
Gianni Lunardi
Matteo Saveriano
Andrea Del Prete



Fontanari, Lunardi, Saveriano, Del Prete (2025). Parallel-Constraint Model Predictive Control: Exploiting Parallel Computation for improving safety. IEEE ICRA.

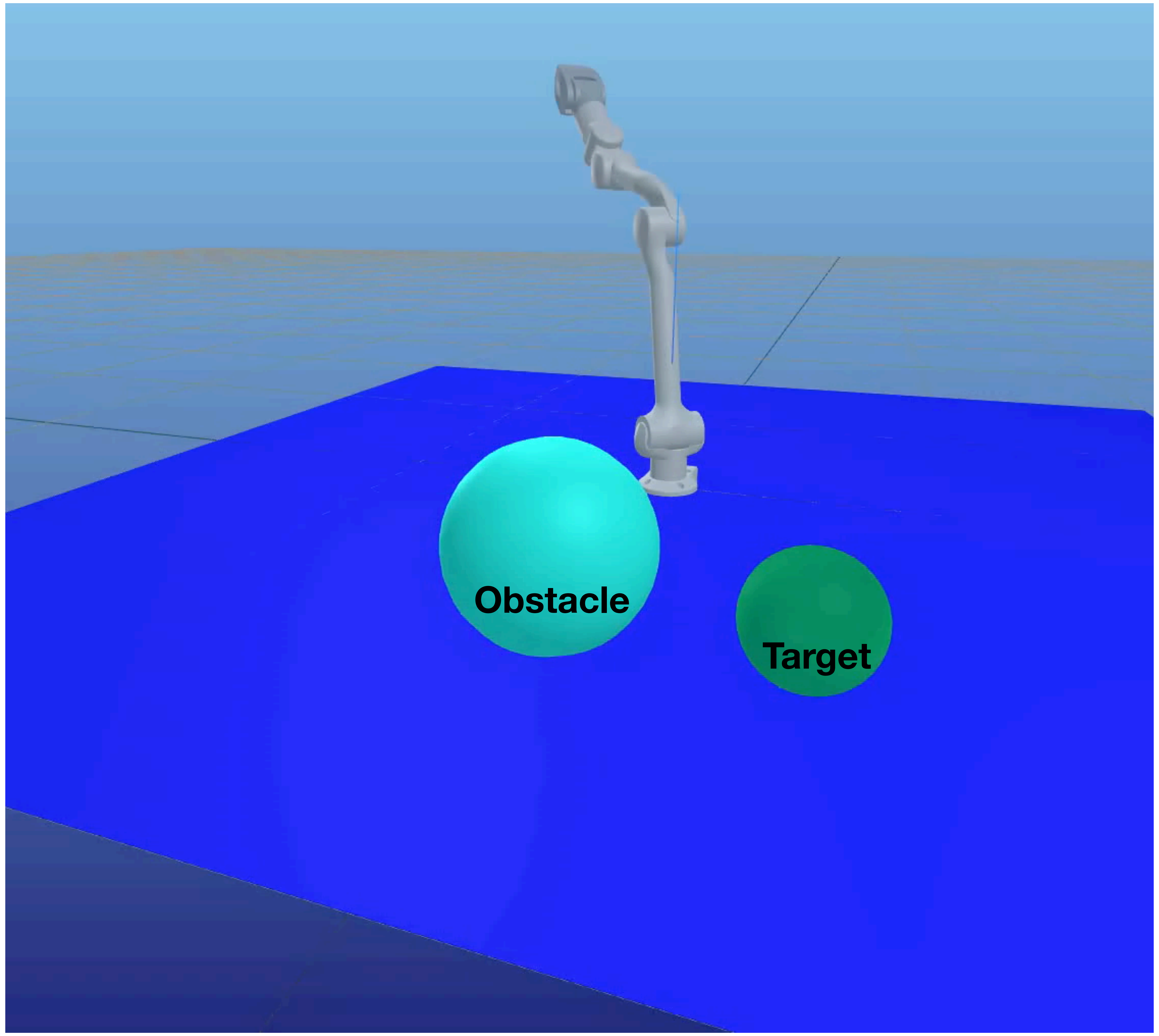
Parallel-Constraint MPC

- Solve in parallel N instances of this problem, one for each value of $p \in [1, N]$:

$$
\begin{aligned}
\underset{\{x_i\}_0^N, \{u_i\}_0^{N-1}}{\text{minimize}} \quad & \sum_{i=0}^{N-1} \ell_i(x_i, u_i) + \ell_N(x_N) \\
\text{subject to} \quad & x_0 = x_{init} \\
& x_{i+1} = f(x_i, u_i) \qquad i = 0 \dots N-1 \\
& x_i \in \mathcal{X}, u_i \in \mathcal{U} \qquad i = 0 \dots N-1 \\
& \boxed{x_p \in \hat{\mathcal{V}}}
\end{aligned}
$$

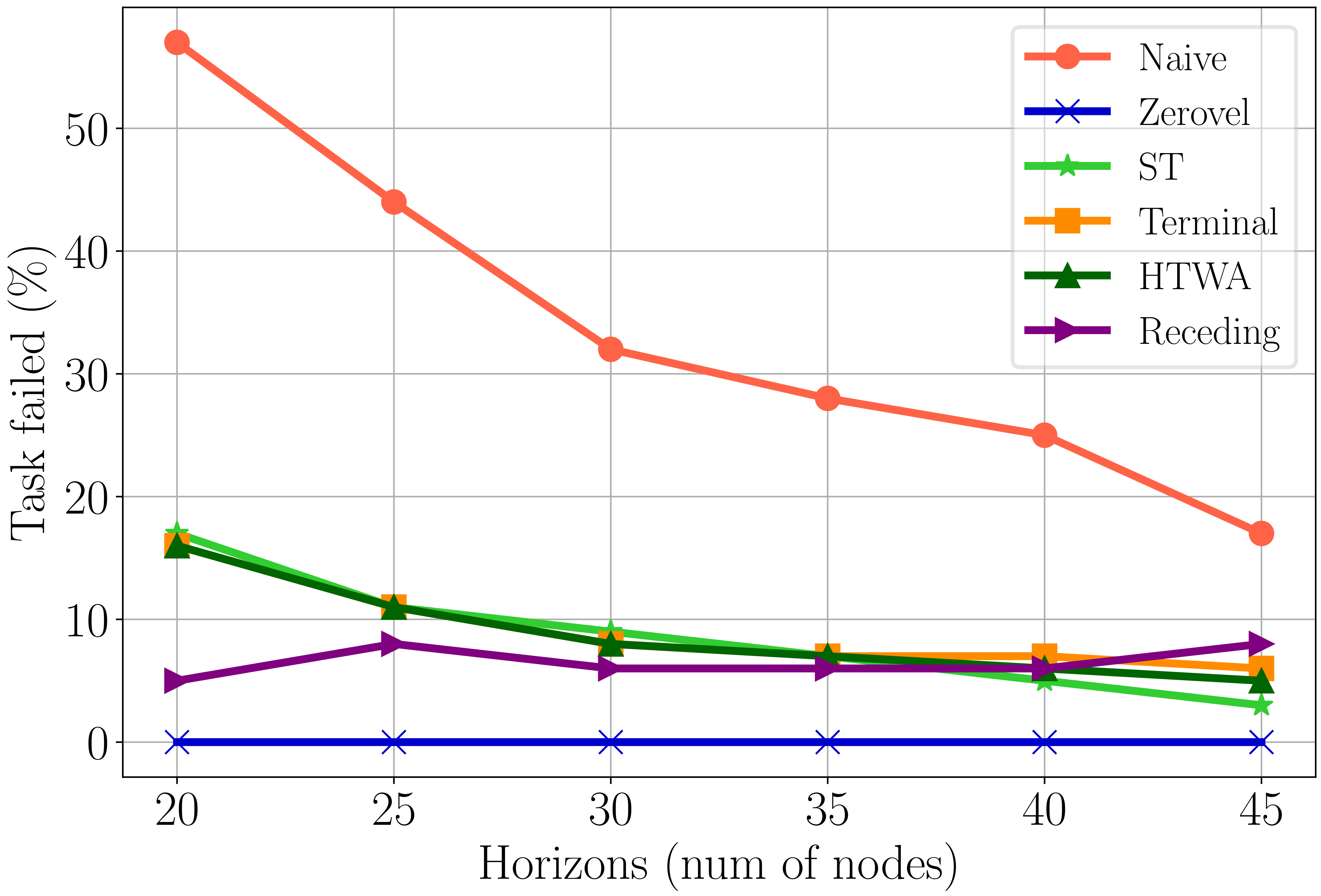
Simulation Results

- Comparing several MPC formulations
- 4 DoF Z1 robot manipulator
- Acados software library
- Safe set $\hat{\mathcal{V}}$ represented with neural network
- 500 simulations from random initial configurations
- Max horizon N=45 to ensure computation time < dt (5 ms)
- https://github.com/idra-lab/safe-mpc



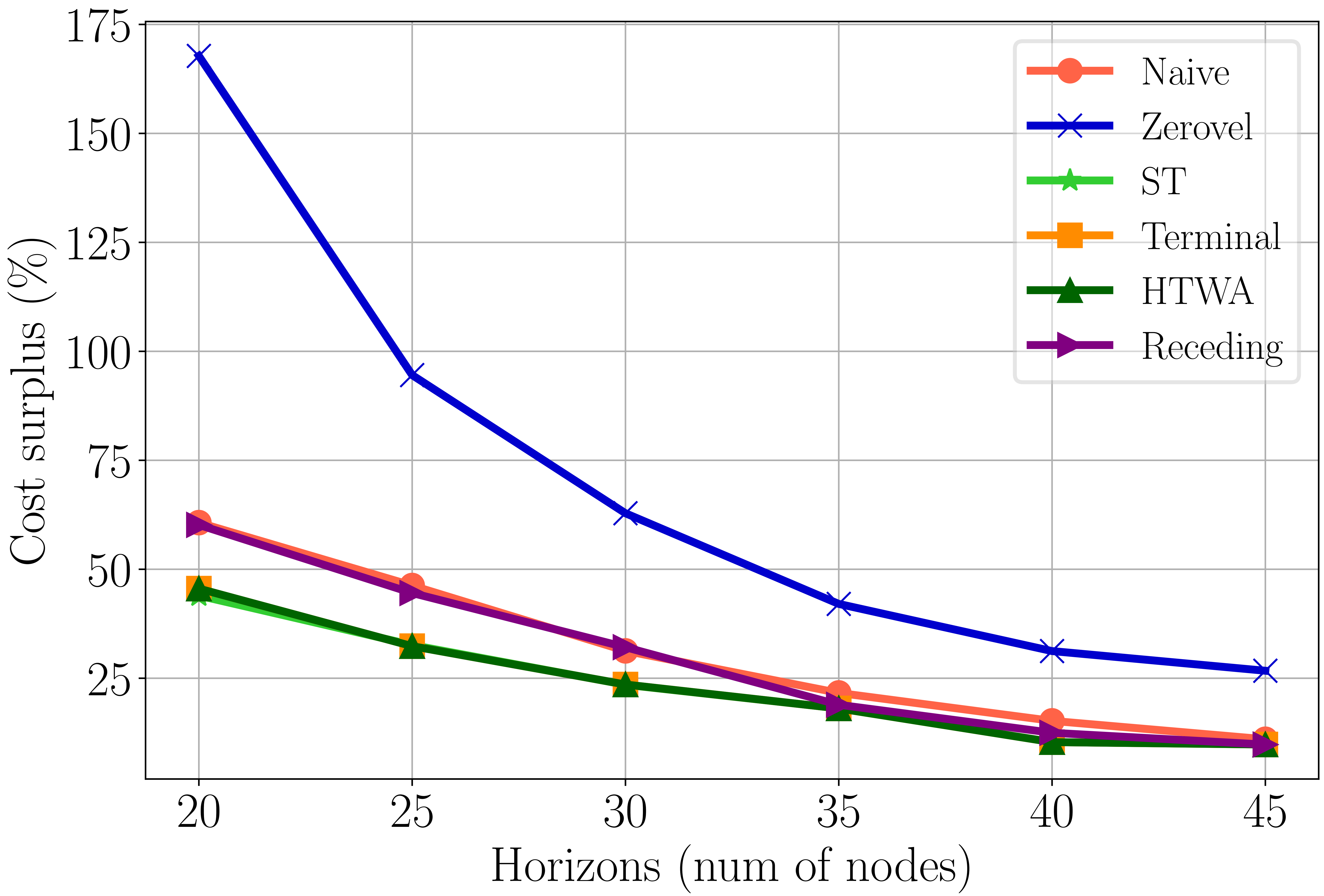
Simulation Results - Receding

- Naive: standard MPC formulation
- Zerovel: terminal constraint imposing zero velocity
- ST: soft terminal constraint $\hat{\mathscr{V}}$
- Terminal: hard terminal constraint $\hat{\mathscr{V}}$
- HTWA: hard terminal constraint $\hat{\mathscr{V}}$ with safe abort strategy



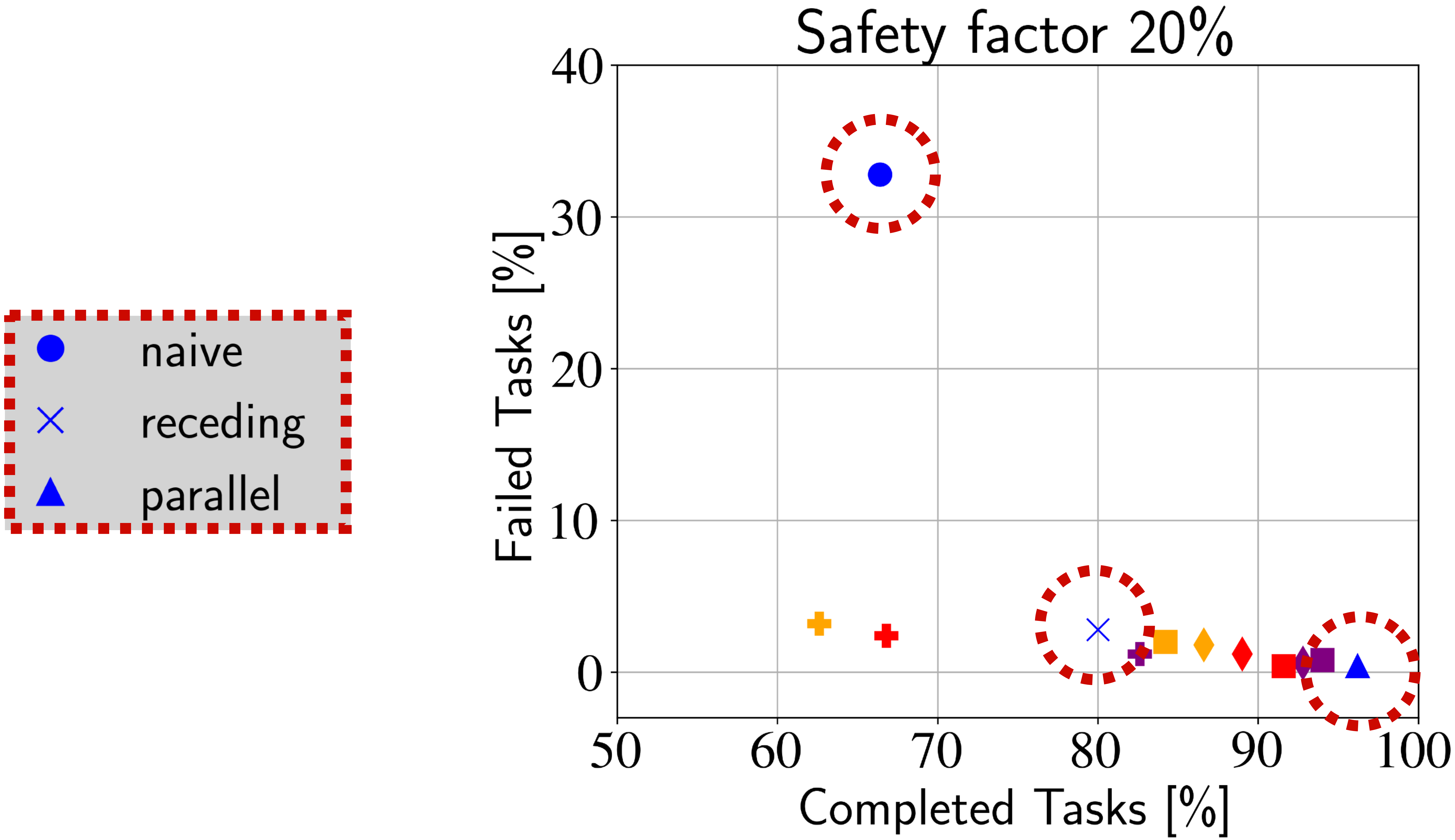
Simulation Results - Receding

- Naive: standard MPC formulation
- Zerovel: terminal constraint imposing zero velocity
- ST: soft terminal constraint $\hat{\mathscr{V}}$
- Terminal: hard terminal constraint $\hat{\mathscr{V}}$
- HTWA: hard terminal constraint $\hat{\mathscr{V}}$ with safe abort strategy



Simulation Results - Parallel

3-DoF manipulator



Computation Time

MPC Formulation	Computation Time (99-Percentile) [ms]
Naive	3.8
Soft Terminal	5.5
Soft Terminal with Abort	3.7
Hard Terminal with Abort	3.9
Receding Constraint	**3.9**

Conclusions

- Novel MPC formulations ensuring
  - Recursive feasibility under weaker conditions (N-Step CIS)
  - Safety under even weaker conditions (inner approx. of CIS)

On-going/future work

- Hardware implementation
- Computation/certification of N-Step CIS
- Handle dynamics uncertainties/obstacles
- Application as safety filter for RL policies

Safe and Efficient robot control

Combining learning and trajectory optimization

Andrea Del Prete

UNIVERSITY OF TRENTO

* UNIVERSITY OF TRENTO

** UNIVERSITY OF NOTRE DAME

*** ENS | PSL

CACTO: Continuous Actor-Critic with Trajectory Optimization

Gianluigi Grandesso*,
Elisa Alboni*,
Gastone Rosati Papini*,
Patrick Wensing,**
Justin Carpentier*,**
Andrea Del Prete*





[1] (2023) CACTO: Continuous Actor-Critic With Trajectory Optimization - Towards Global Optimality. IEEE RA-L
[2] (2024) CACTO-SL: Using Sobolev Learning to improve Continuous Actor-Critic with Trajectory Optimization. In L4DC

Reinforcement Learning ~~VS~~ Trajectory Optimization

WITH?

$$\min_{x(t),u(t)} \int_0^T l\left(x(t),u(t)\right)dt + l_f\left(x(T)\right)$$

$$\begin{aligned} \text{s.t.} \quad & \dot{x}(t) = f(x(t),u(t),t) \quad \forall t \in [0,T] \\ & x(0) = x_0 \\ & u_{min} < u(t) < u_{max} \quad \forall t \in [0,T] \end{aligned}$$

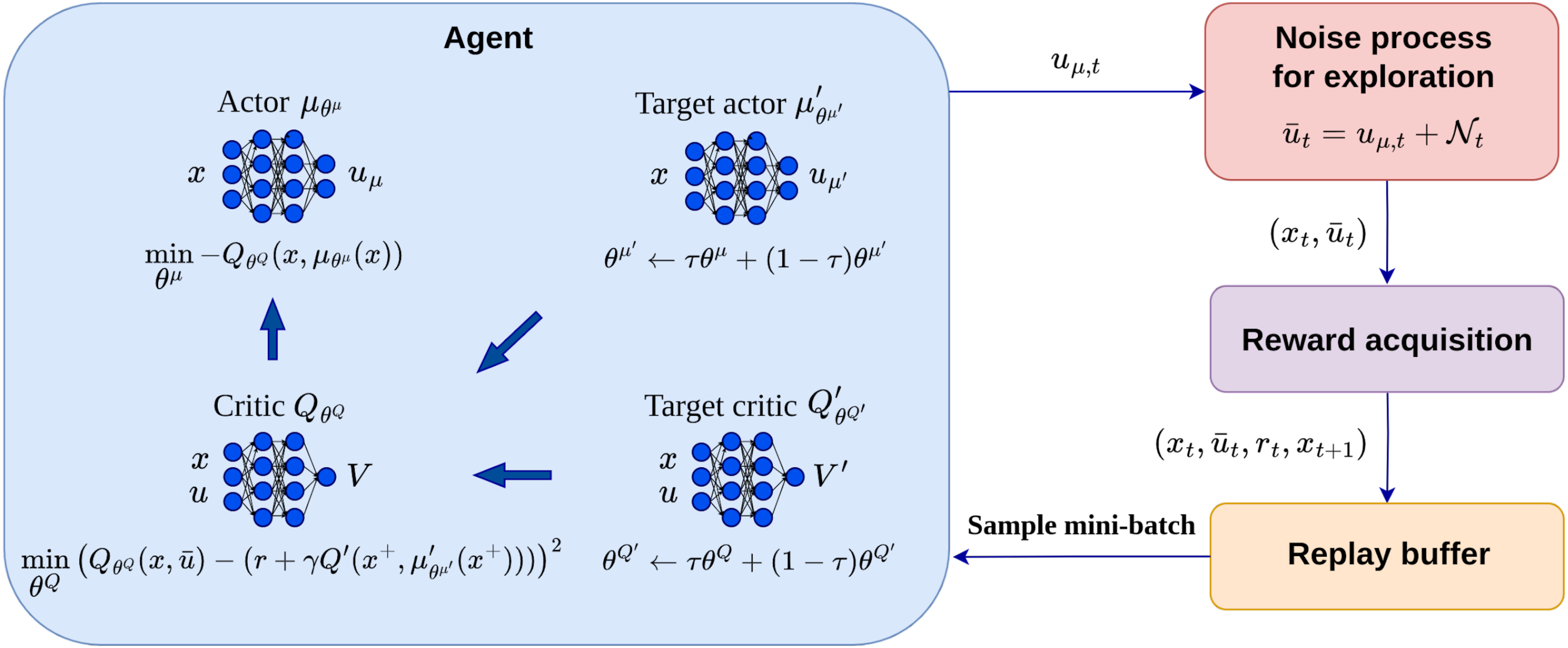
Reinforcement Learning

- \+ Less prone to poor local minima
- \+ Derivative free
- \+ Policy as output
- − Poor data efficiency (slow)

Trajectory Optimization

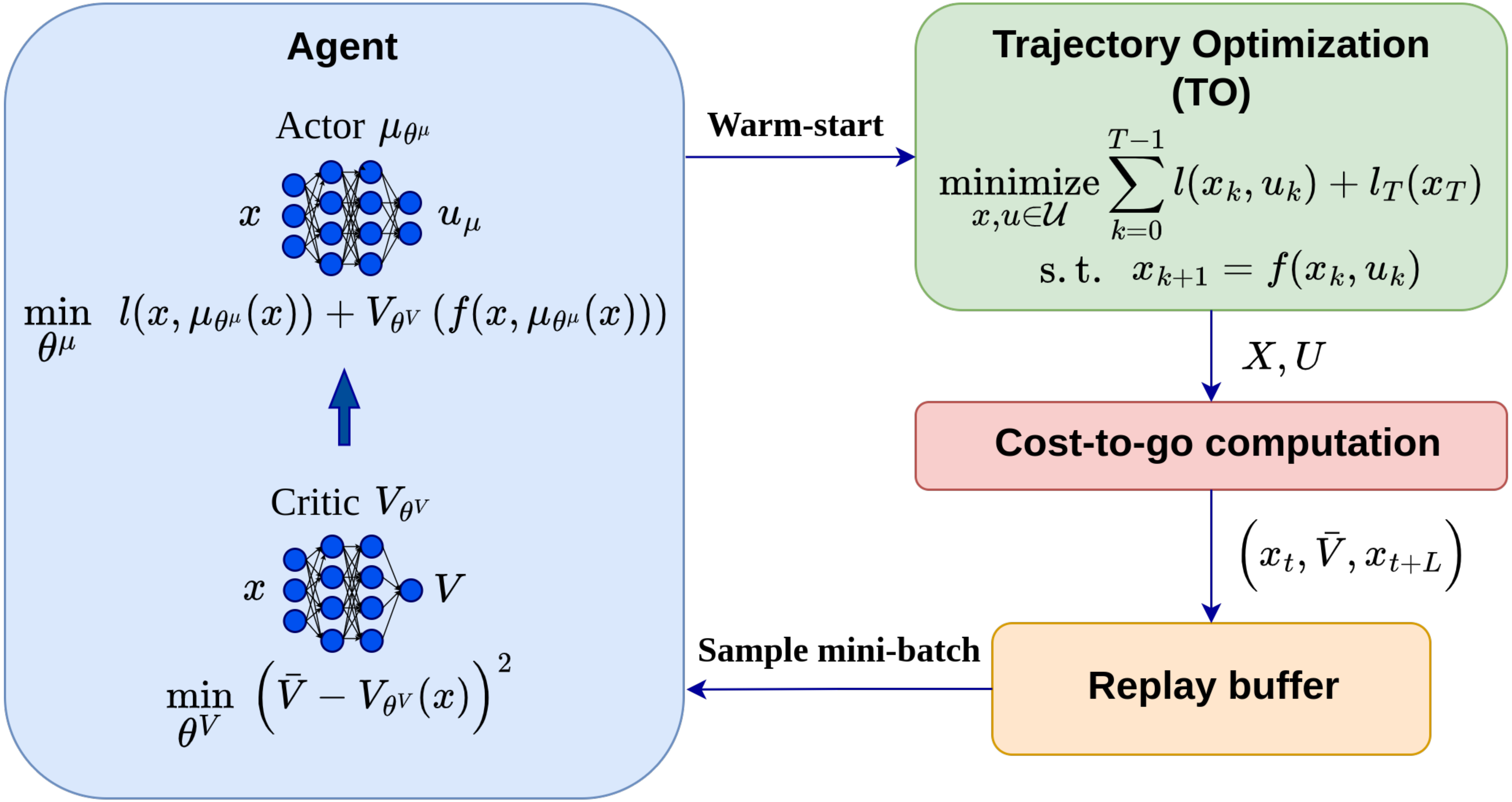
- \+ Data efficient (fast)
- \+ Exploits knowledge of dynamics derivatives
- − Can get stuck in poor local minima
- − Trajectory as output

Deep Deterministic Policy Gradient (DDPG)



Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., … Wierstra, D. (2015). Continuous control with deep reinforcement learning. In Foundations and Trends in Machine Learning

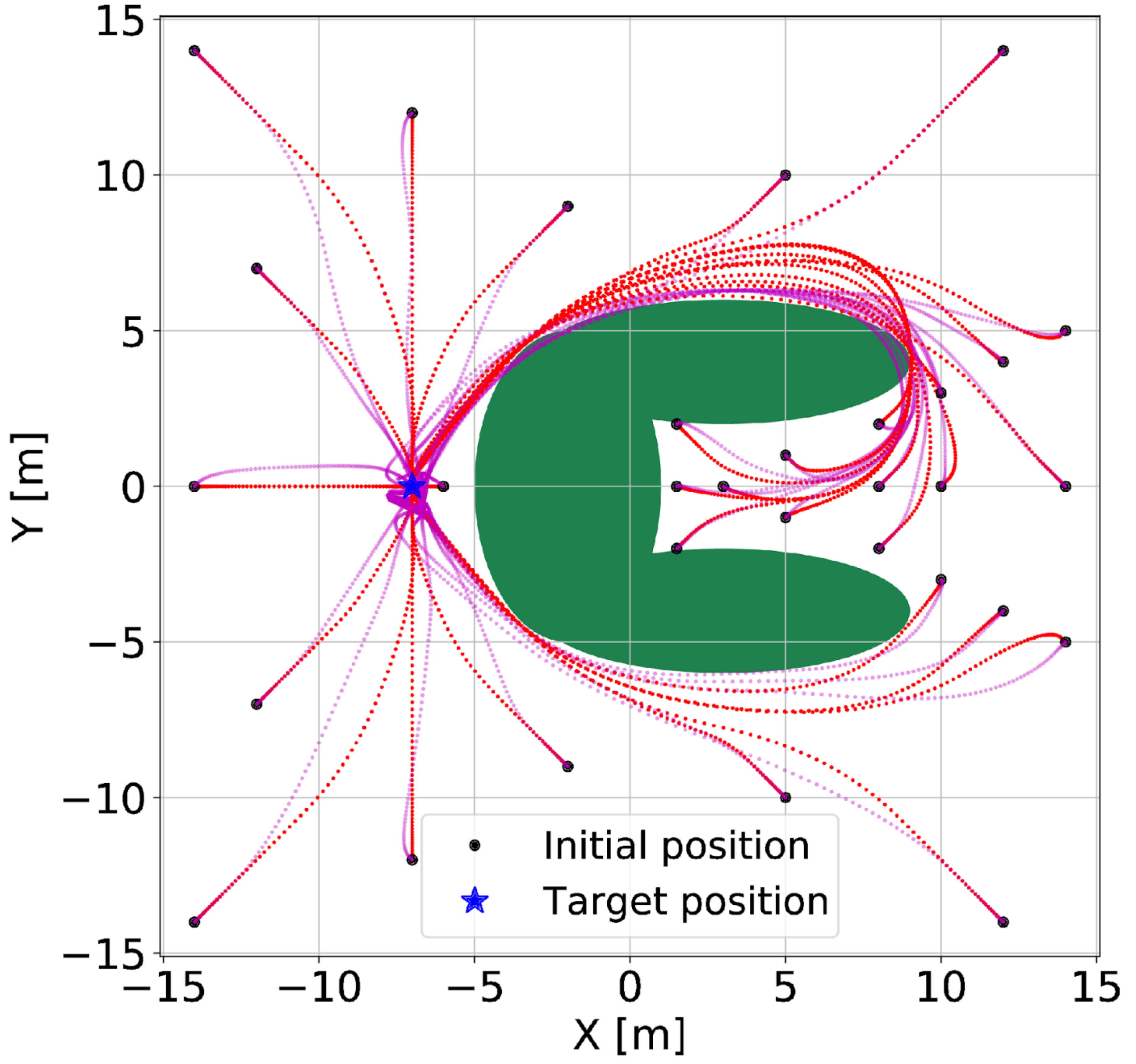
CACTO



[1] Grandesso, Alboni, Rosati Papini, Wensing, Del Prete (2023). CACTO: Continuous Actor-Critic With Trajectory Optimization - Towards Global Optimality. IEEE Robotics and Automation Letters

Results

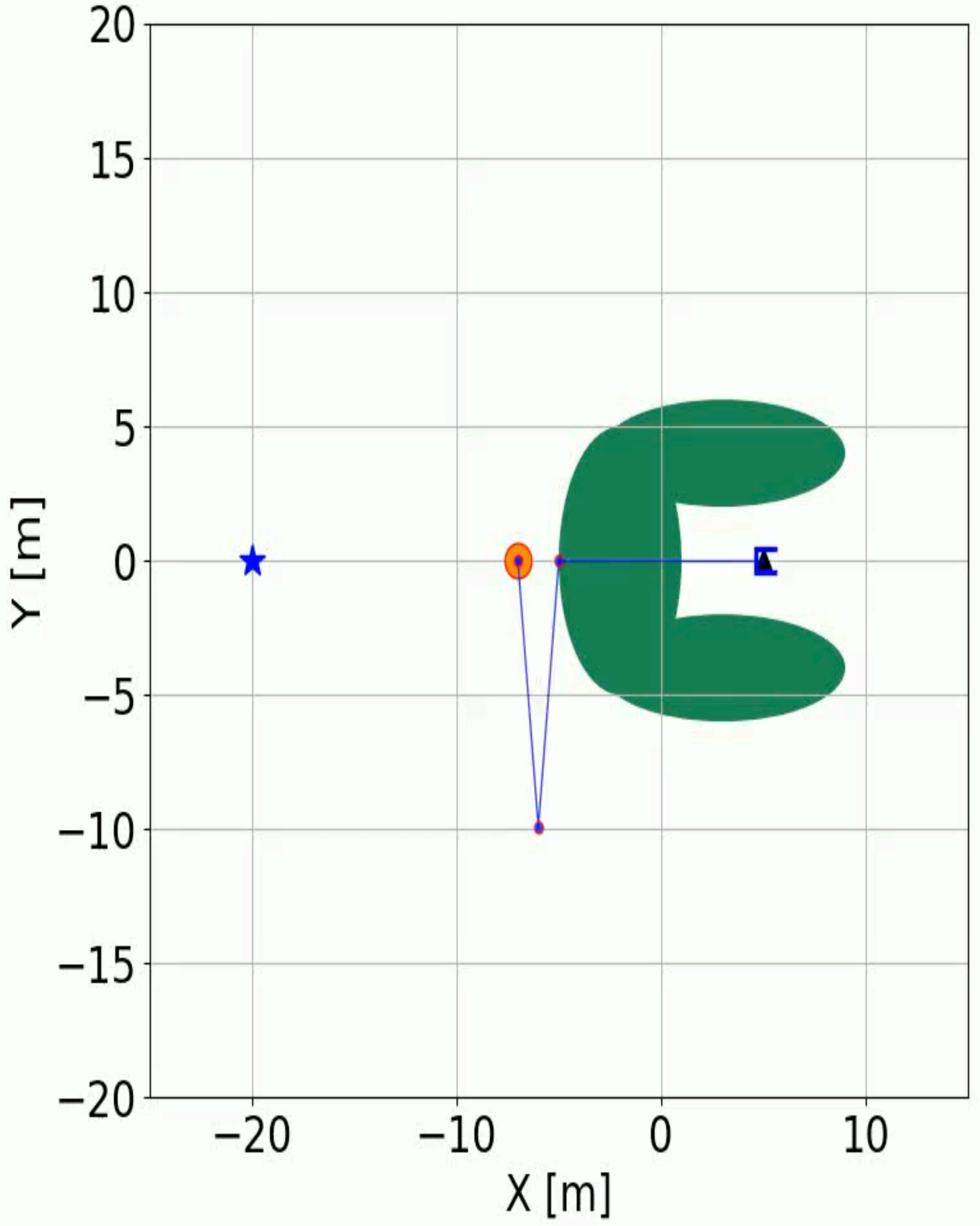
Task: find shortest path to target using low control effort and avoiding obstacles



Systems: 2D single/double integrator, 6D car model, 3-joint manipulator

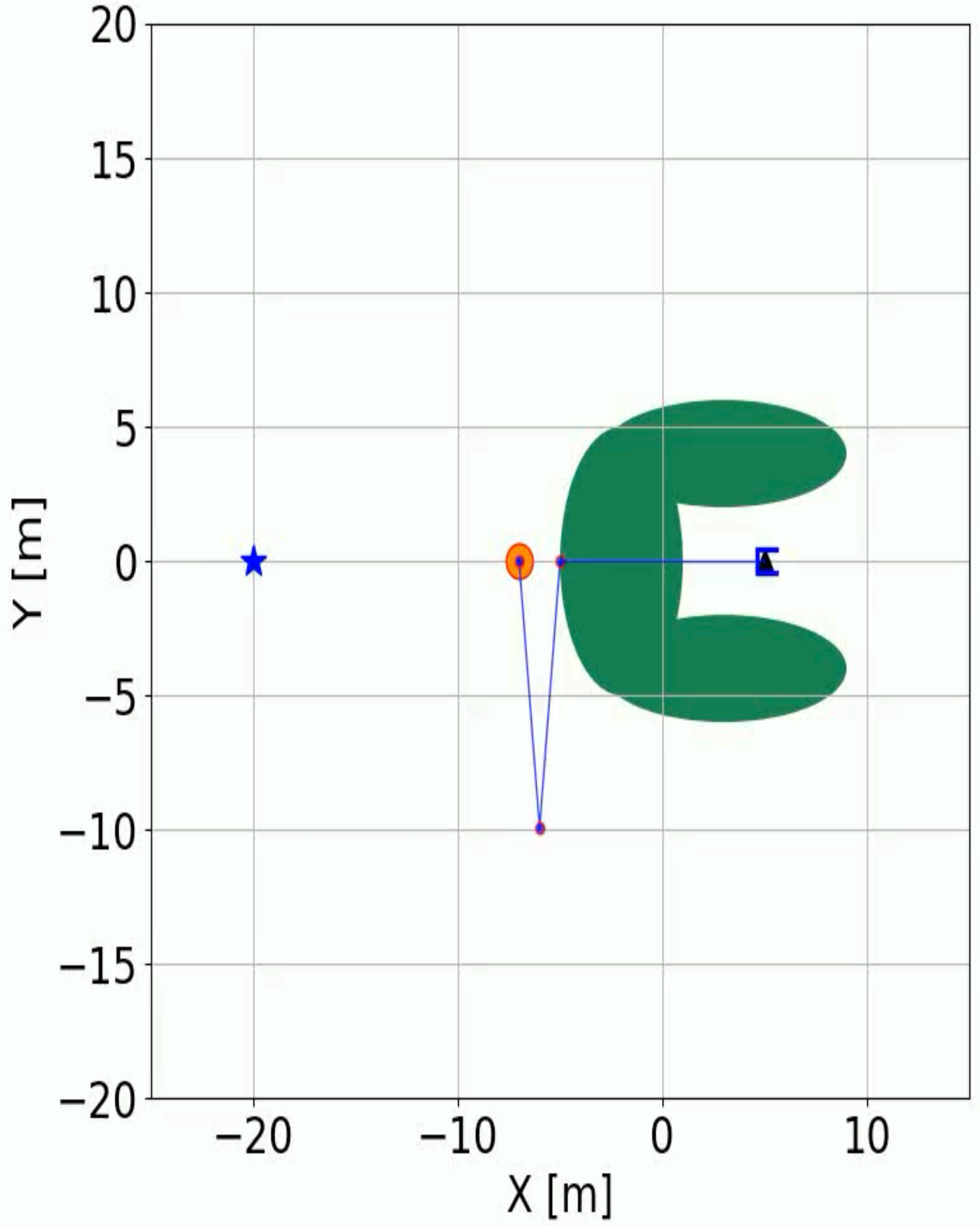
Results: 3-DoF Manipulator

Initial Conditions
warm-start



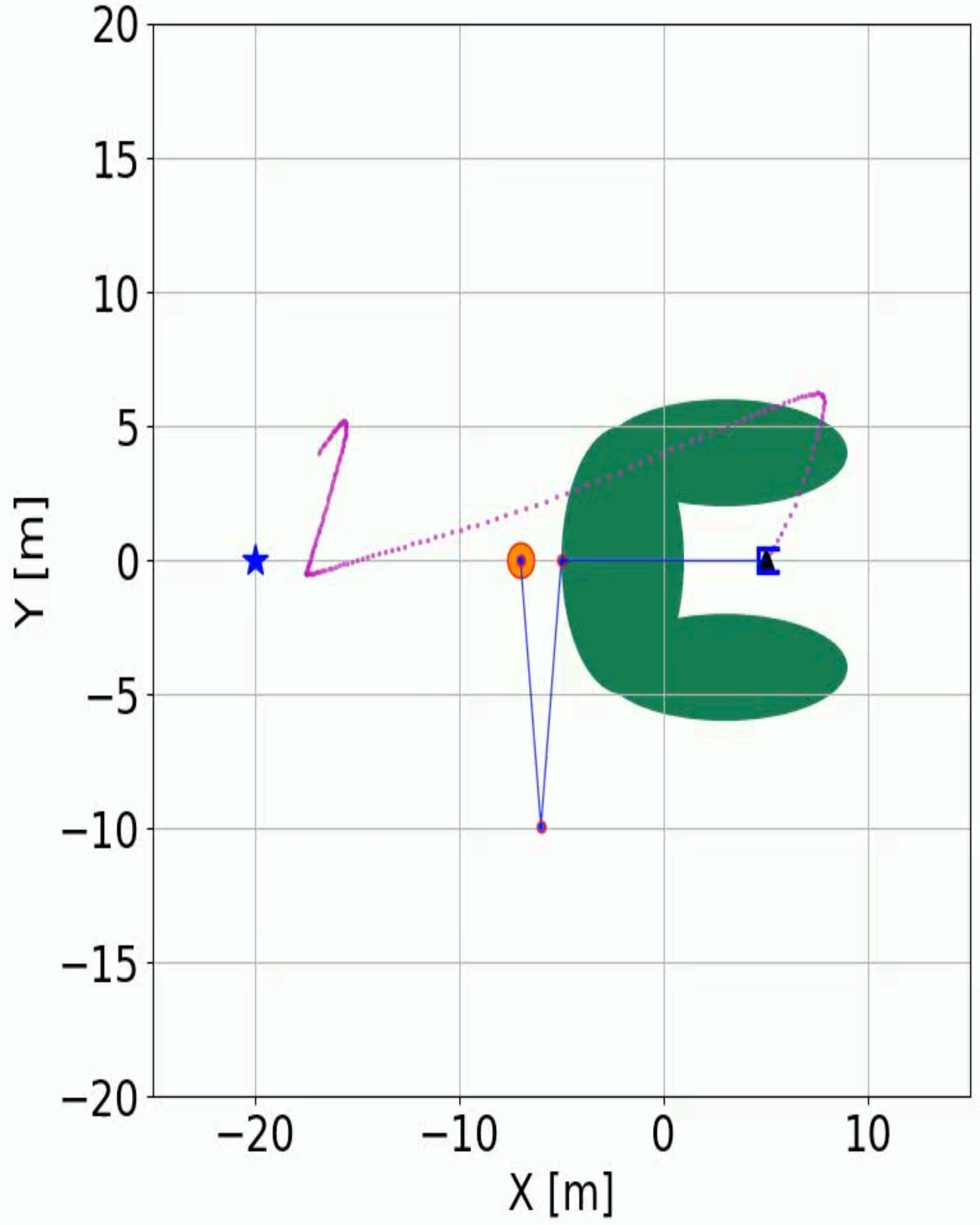
Cost = 70800

Random
warm-start



Cost = 88647

CACTO
warm-start



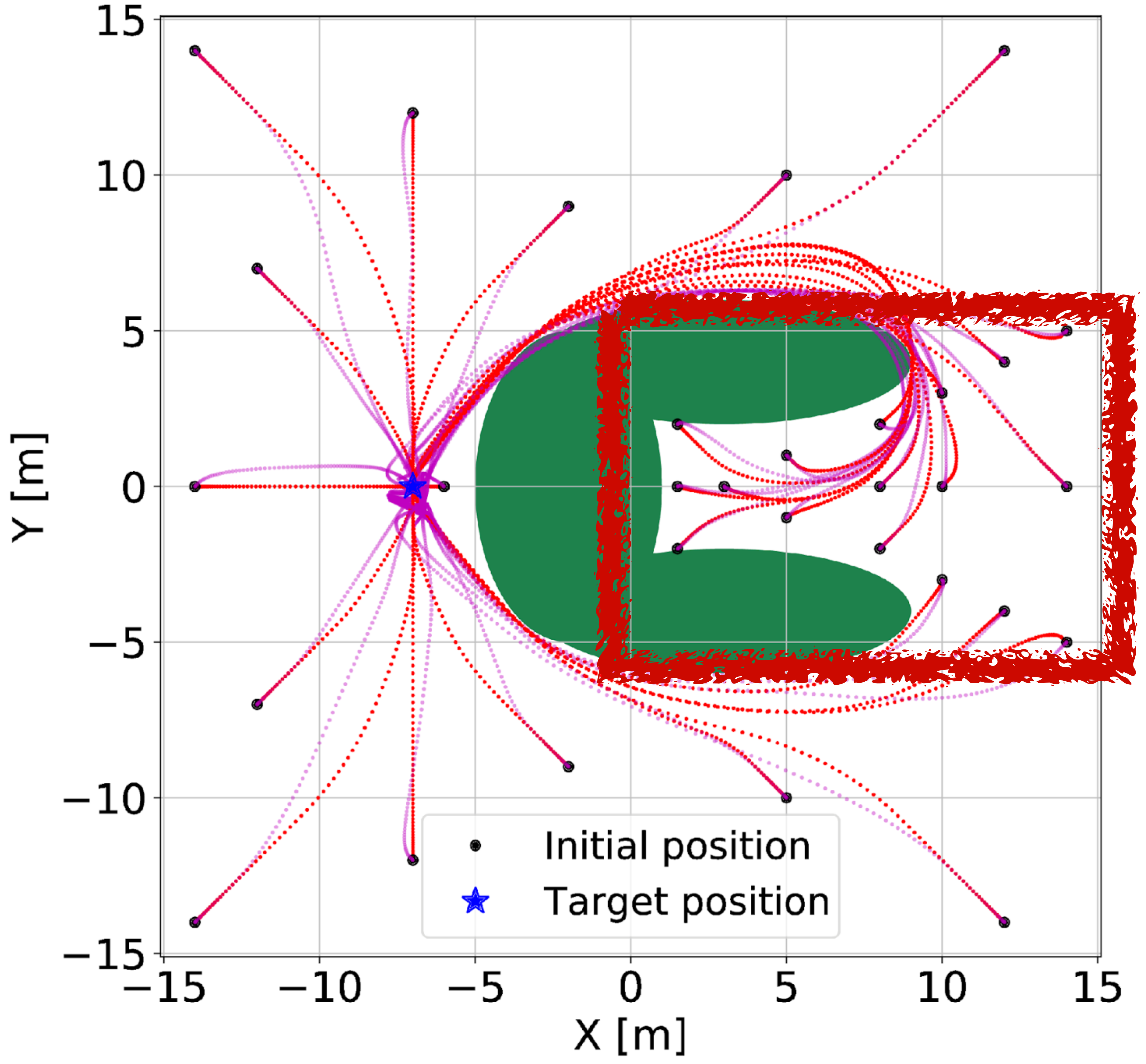
Cost = -145875

Comparison: CACTO vs TO

% of times TO finds better solution if warm-started with CACTO rather than:

- Random values
- Initial conditions (ICS) for states, zero for other variables

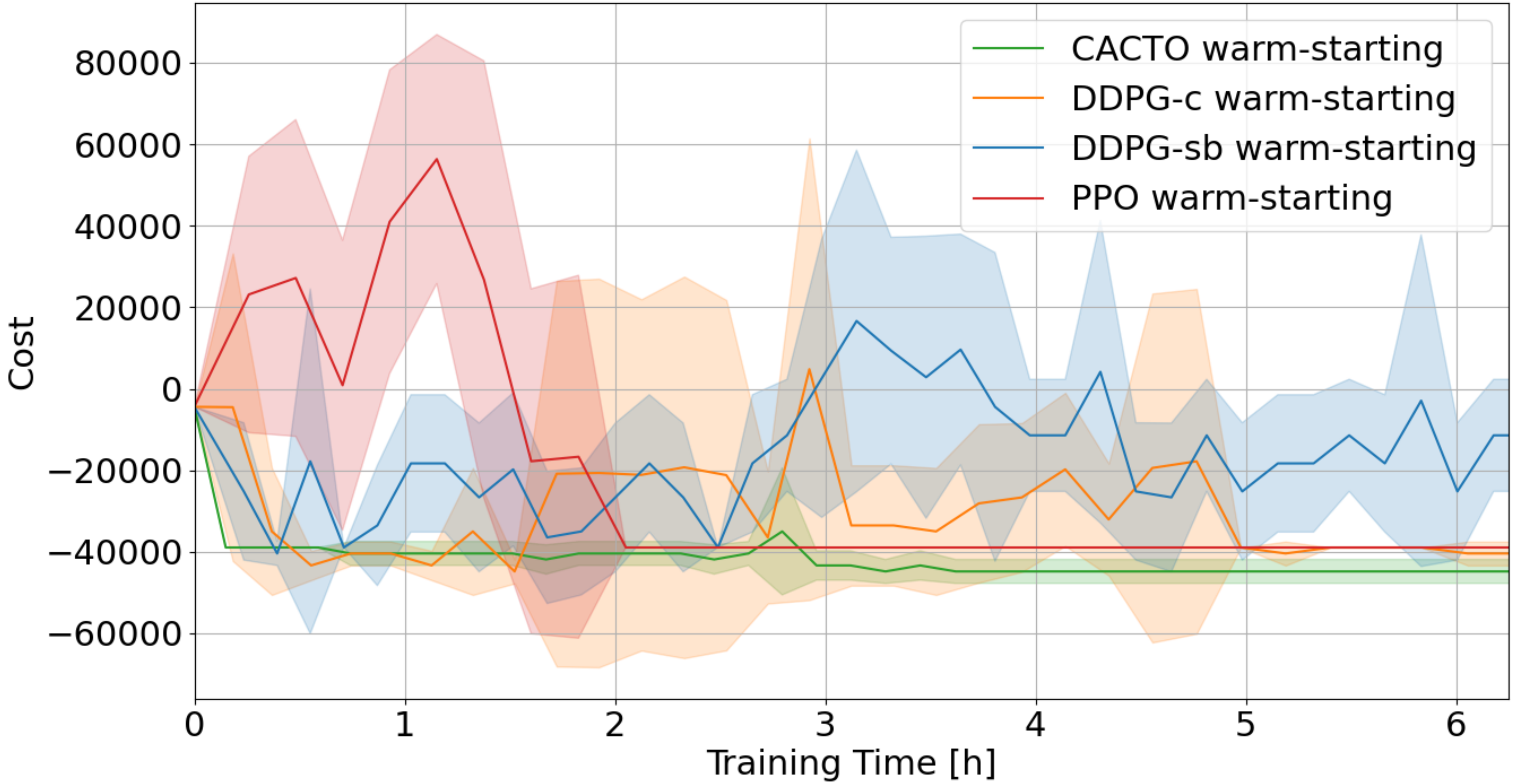
System	Hard Region	
	CACTO < (≤) Random	CACTO < (≤) ICS
2D Single Integrator	**99.1%** (99.1%)	**92%** (99.1%)
2D Double Integrator	**99.9%** (99.9%)	**92%** (99.1%)
Car	**100%** (100%)	**92.9%** (100%)
Manipulator	**87.5%** (87.5%)	**100%** (100%)



2D Double Integrator - CACTO warm-start

Comparison: CACTO, DDPG, PPO

Mean cost + std. dev. (across 5 runs) found by TO warm-started with different policies

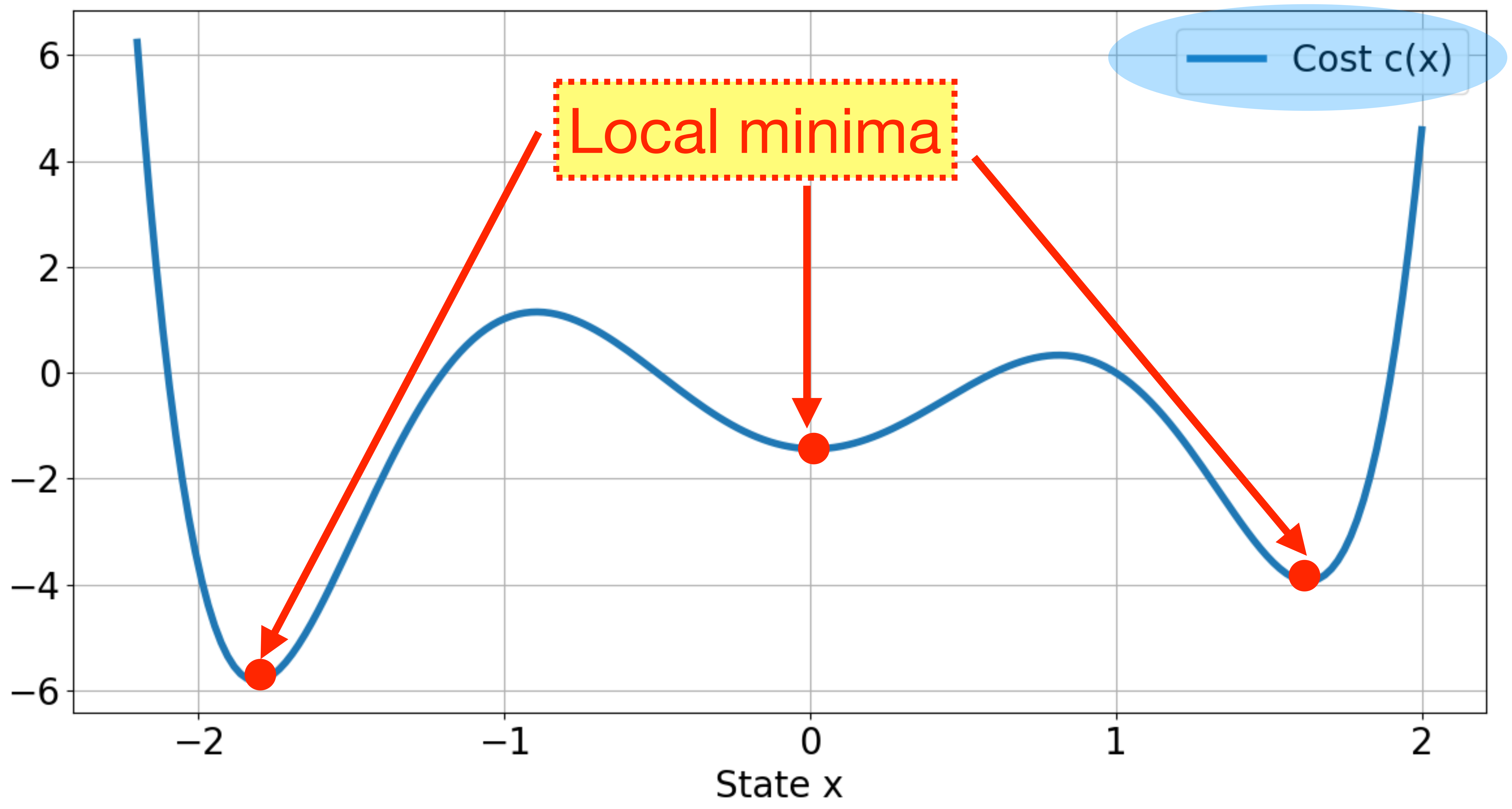


1D Example

$$\underset{X,U}{\text{minimize}} \quad \sum_{k=0}^{T-1} \left[c(x_k) + w_u ||u_k||^2 \right] + c(x_T)$$

$$\text{subject to} \quad x_{k+1} = x_k + \Delta t\, u_k \quad \forall\, k = 0, \ldots, T-1$$

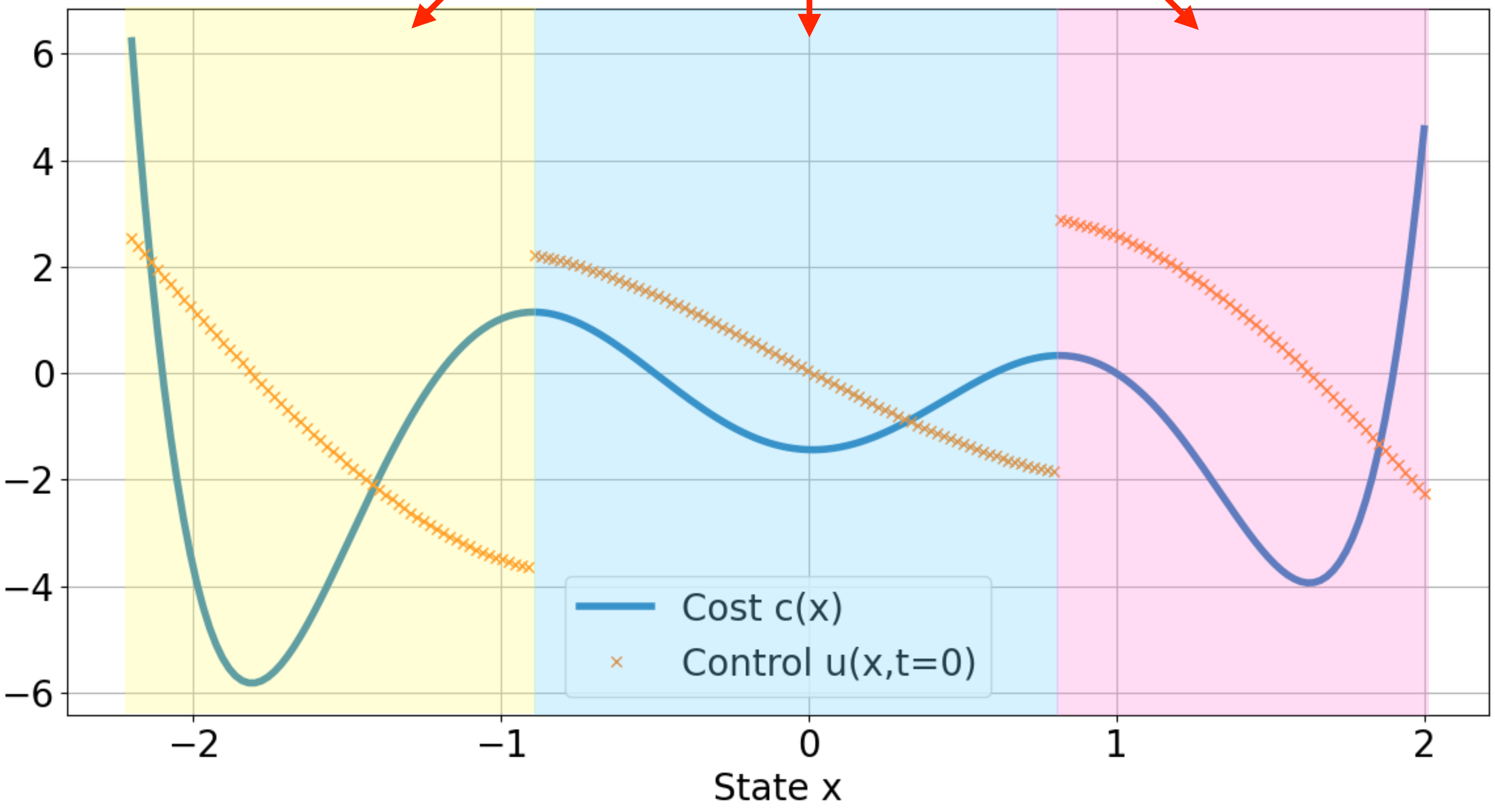
$$x_0 = x_{init}$$



Trajectory Optimization

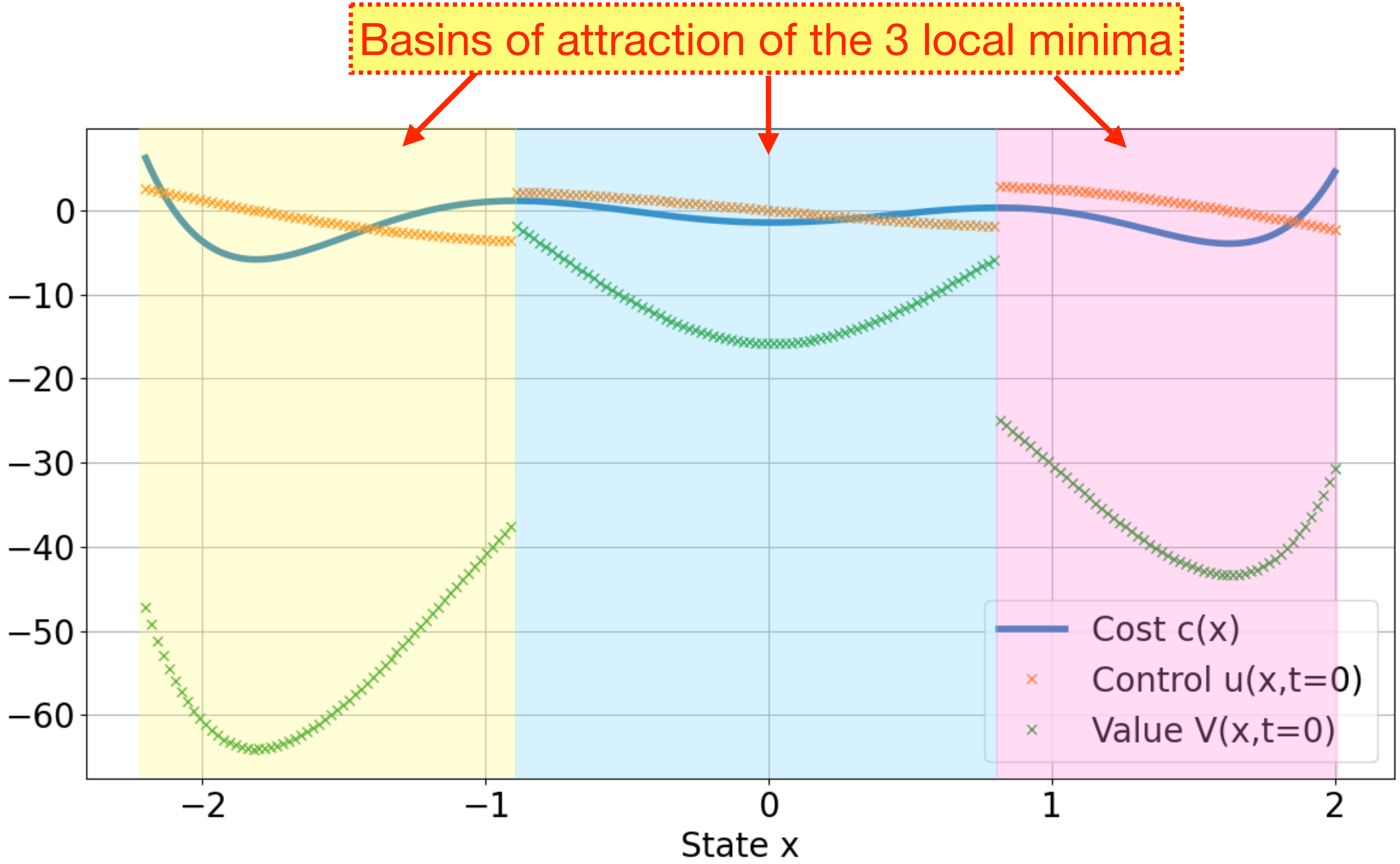
With naive initial guess

Basins of attraction of the 3 local minima



Trajectory Optimization

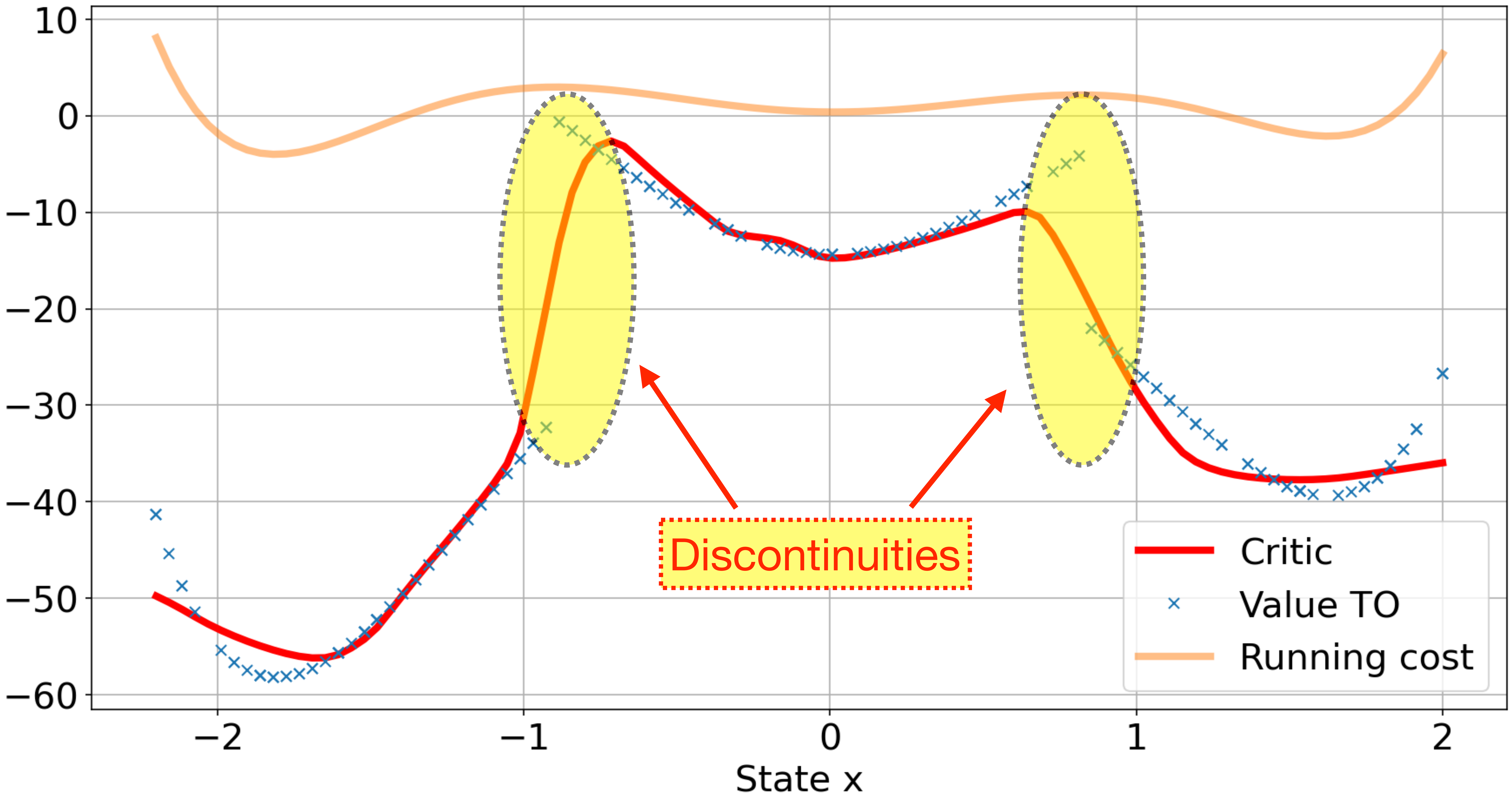
With naive initial guess



First Iteration

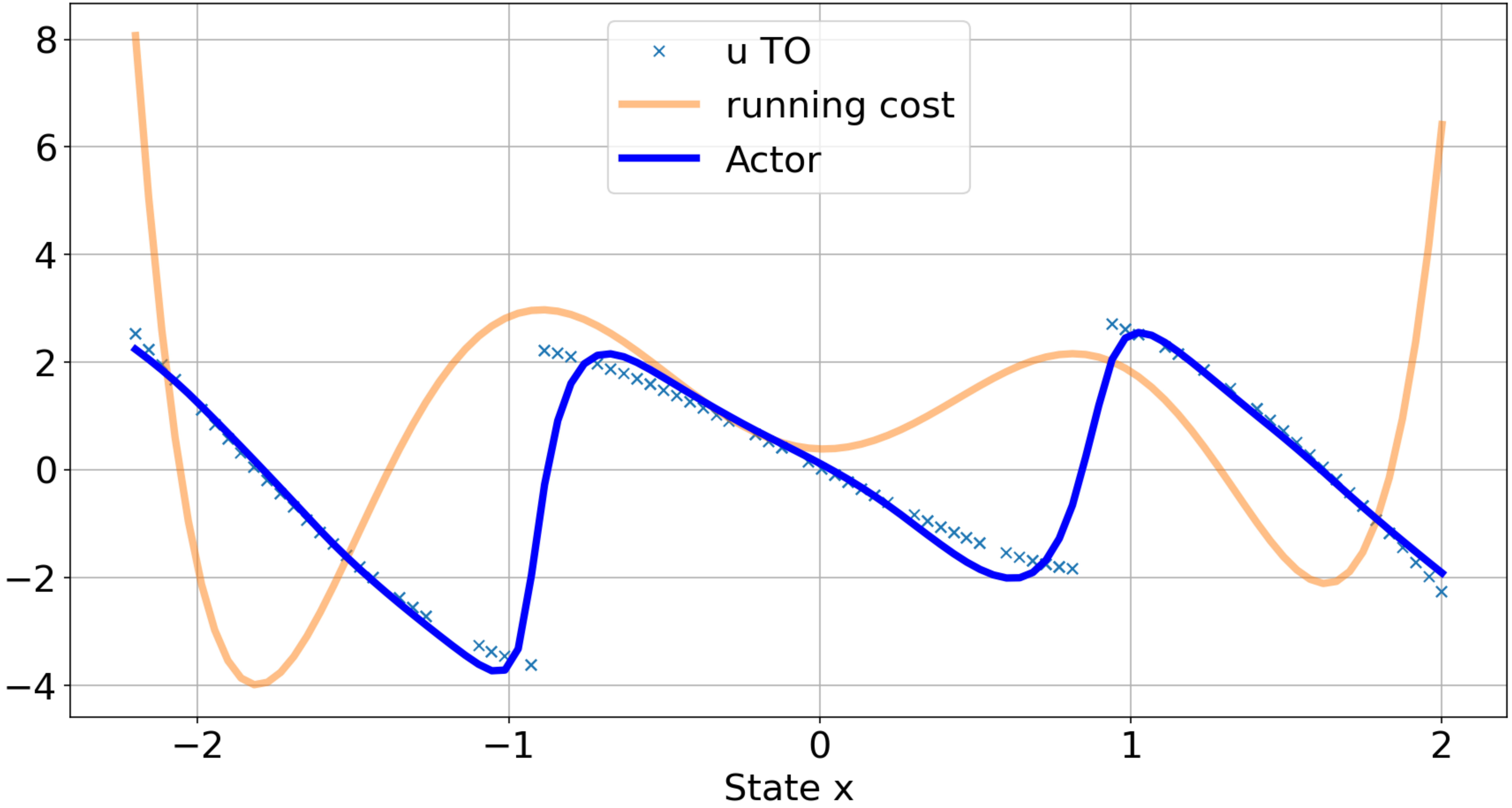
Learning the critic

The Value function is discontinuous so the network approximates it.



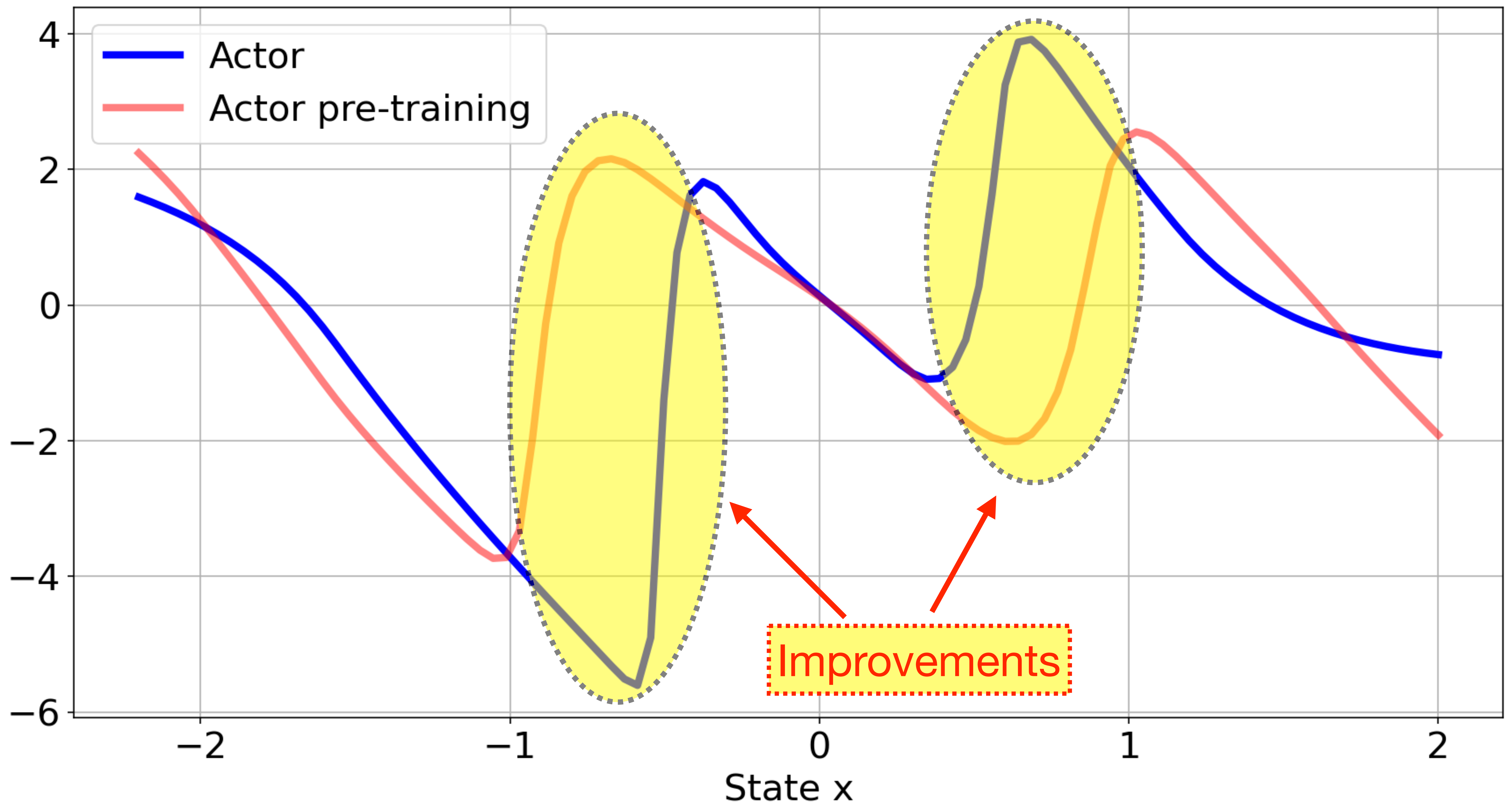
Supervised Learning of the actor

At the first iteration we pre-train the actor to imitate the control inputs of TO.



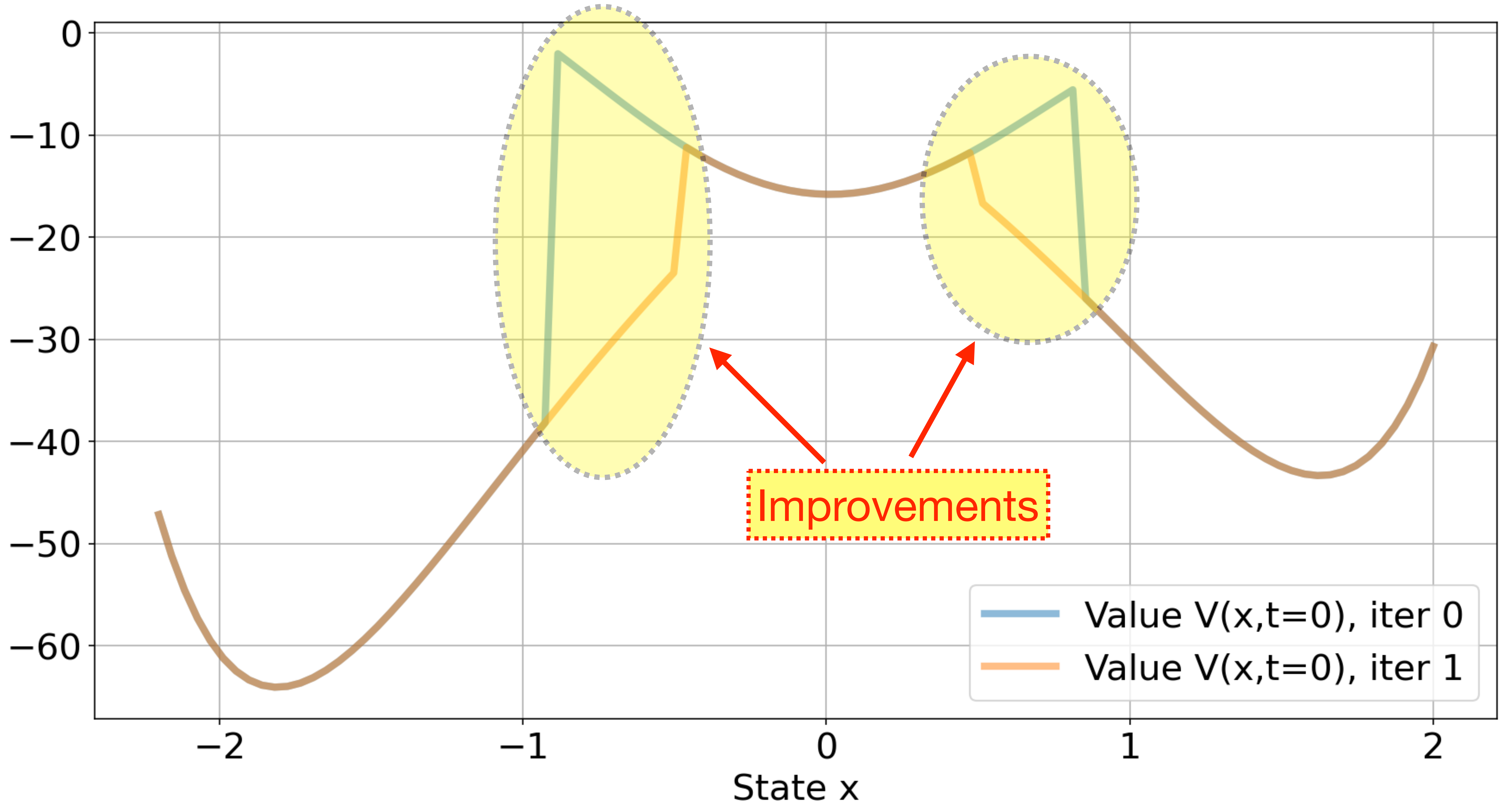
Learning the actor minimizing Q

We improve the actor by minimizing the Q function



Using the actor to warm-start TO

TO improves thanks to the initial guess of the actor



Conclusions

- TO guides the RL exploration making it sample efficient
- Global convergence proof for discrete-space version of CACTO

Recent extension

- Improve data efficiency leveraging derivative of Value function [2]

Future work

- Bias initial episode state to improve data efficiency
- Parallelize on GPUs
- Handle non-differentiable dynamics

[2] Alboni, Grandesso, Rosati Papini, Carpentier, Del Prete (2024). CACTO-SL: Using Sobolev Learning to improve Continuous Actor-Critic with Trajectory Optimization. In Learning for Dynamics and Control Conference (L4DC)

Take-Home Message

Globally Optimal and Safe Robot Control

- Using ideas from TO we can make RL efficient and safe
  - Use dynamics derivatives to guide RL exploration (CACTO)
  - Use novel safe sets to make control (RL) safe

Current challenges

- algorithms to compute $\hat{\mathscr{V}}$ do not scale and cannot certify set properties (e.g. N-Step Control Invariance)
- dynamics derivatives are ill-defined in contact-rich tasks

Safe and Efficient Robot Control

Combining learning and trajectory optimization

Andrea Del Prete

UNIVERSITY OF TRENTO