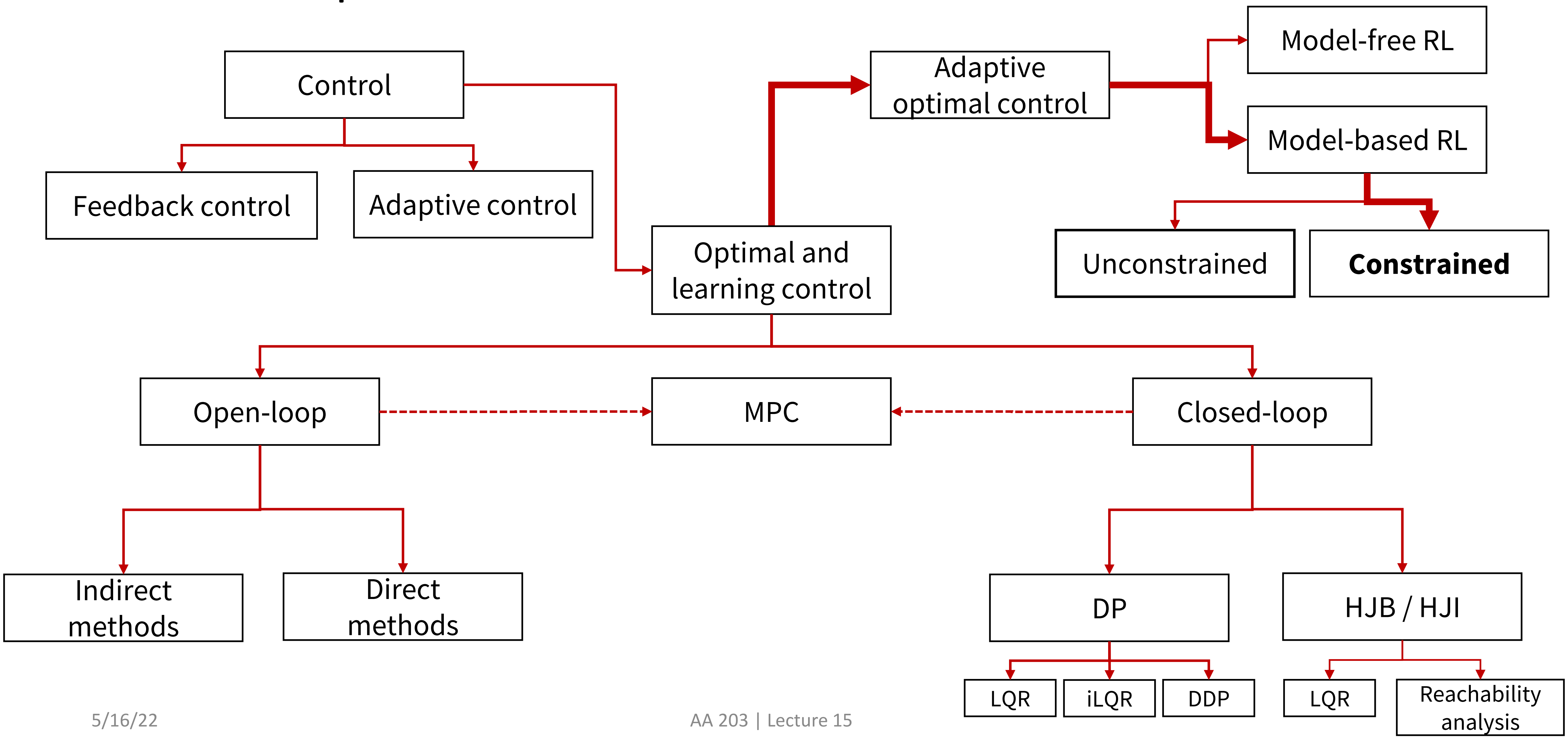
AA203
Optimal and Learning-based Control

Adaptive and learning MPC

Stanford University

Autonomous Systems Lab

Roadmap


Control
Feedback control
Adaptive control
Optimal and learning control
Adaptive optimal control
Model-free RL
Model-based RL
Unconstrained
Constrained
Open-loop
MPC
Closed-loop
Indirect methods
Direct methods
DP
HJB / HJI
LQR
iLQR
DDP
LQR
Reachability analysis

Adaptive and Learning MPC

- Learning MPC as an example of learning/adaptive constrained control
- Practical considerations
- Learning quantities other than dynamics

- Reading:
  - L. Hewing, K. P. Wabersich, M. Menner, M. N. Zeilinger. *Learning-Based Model Predictive Control: Toward Safe Learning in Control.* Annual Review of Control, Robotics, and Autonomous Systems, 2020.
  - U. Rosolia, X. Zhang, F. Borrelli. *Data-Driven Predictive Control for Autonomous Systems.* Annual Review of Control, Robotics, and Autonomous Systems, 2018.

Learning dynamics

- Approach:
  - Learn dynamics and maintain a measure of uncertainty
  - Incorporate uncertainty into controller to guarantee constraint satisfaction
    - Using, e.g., robust MPC
- Model learning types:
  - Robust/Set-membership models
    - Typically easier analysis, potentially sensitive to problem misspecification
  - Statistical models (e.g., least squares estimation)
    - More difficult analysis, able to account for more complicated interactions between uncertainties

Robust estimation models

- Setting: given operation data

$$X = [\mathbf{x}(0), \dots, \mathbf{x}(K+1)], \qquad U = [\mathbf{u}(0), \dots, \mathbf{u}(K)]$$

from system

$$\mathbf{x}(t+1) = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), \boldsymbol{\theta})$$
$$\mathbf{w}(t) \in W \ \ \forall t$$

- Approach: maintain feasible parameter set

$$\mathrm{T}_K = \{\boldsymbol{\theta}: \forall t = 0, \dots, K \ \exists \mathbf{w} \in W \ \text{ s.t. } \mathbf{x}(t+1) = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), \boldsymbol{\theta})\}$$

Set of *non-falsified* parameters

Robust estimation models

- Note that $T_{K+1} \subseteq T_K$: once a parameter value is falsified, it is removed from the feasible set forever.
- Frequently used consequence:
  - Let $U = [\mathbf{u}(0), \dots, \mathbf{u}(N)]$ denote a feasible open loop action sequence from state $\mathbf{x}(0)$ for all $\boldsymbol{\theta} \in T_K$. Then, U is feasible for all $\boldsymbol{\theta} \in T_{K+n}$ with $n \geq 0$ (from the same state $\mathbf{x}(0)$).

Additive linear example

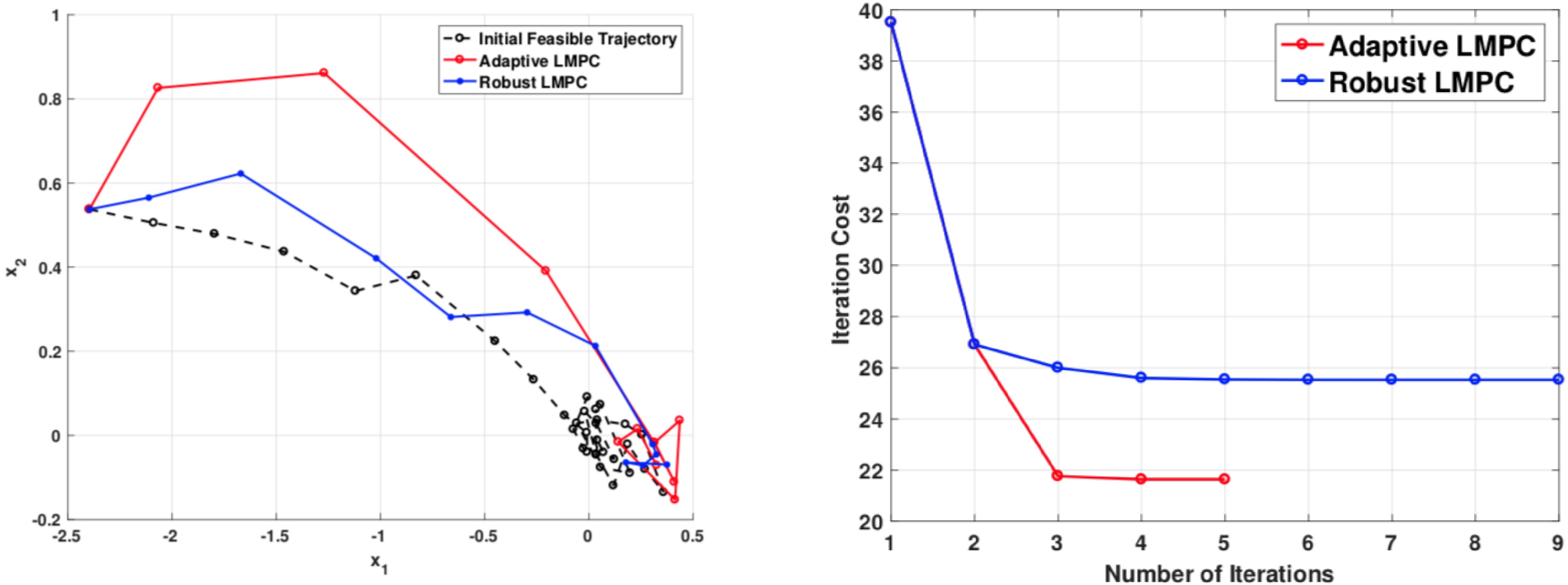
- Dynamics
$$\mathbf{x}(t+1) = A\mathbf{x}(t) + B\mathbf{u}(t) + \mathrm{E}\boldsymbol{\theta} + \mathbf{w}(t); \quad \mathbf{w}(t) \in W$$
  E known, $\boldsymbol{\theta}$ unknown.
- Assume initial polytopic parameter uncertainty set T_0.
- Polytopic constraints $F\mathbf{x} \leq \mathbf{f}, G\mathbf{u} \leq \mathbf{g}$.

Additive linear example

Bujarbaruah, Zhang, Rosolia, Borrelli. *Adaptive MPC for Iterative Tasks*, CDC 2019.

- Let X_f denote terminal invariant associated with dynamics and T_0.
- Then, X_f also invariant for $T_n, \mathrm{n} \geq 0$.
- Approach: At timestep n, consider combined disturbance
$$\mathbf{d}(t) = E\boldsymbol{\theta} + \mathbf{w}(t), \qquad \boldsymbol{\theta} \in T_n$$
  Use robust/tube MPC to solve.
- Can also adapt terminal invariant, will see later.

Additive linear example



Bujarbaruah, Zhang, Rosolia, Borrelli. *Adaptive MPC for Iterative Tasks*, CDC 2019.

Robust Adaptive MPC

- Many similar approaches for
  - Multiplicative uncertainty
$$\mathbf{x}(t+1) = \theta_A \mathbf{x}(t) + \theta_B \mathbf{u}(t) + \mathbf{w}(t)$$
  - Nonlinear (but linearly parameterized) uncertainty
$$\mathbf{x}(t+1) = A\mathbf{x}(t) + B\mathbf{u}(t) + \Phi\big(\mathbf{x}(t), \mathbf{u}(t)\big)\boldsymbol{\theta}$$
- For nonlinear dependence on $\boldsymbol{\theta}$, there also exist robust *non-parametric* methods

Stochastic estimation models

System

$$\mathbf{x}(t+1) = f(\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), \boldsymbol{\theta})$$

with $\mathbf{w}(t) \sim p(\mathbf{w})$ i.i.d. (independent and identically distributed)
Common assumption: noise appears linearly

$$\mathbf{x}(t+1) = f(\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\theta}) + \mathbf{w}(t)$$

Approach:

- Use tools from *probabilistic* estimation (e.g. max likelihood, Bayesian inference, etc.)
- Construct confidence intervals or credible regions to *probabilistically* guarantee safety

Confidence sets

- In set-membership identification, we constructed sets that contained the parameters with probability 1
- In this section, we will consider sets of the form $T_k(\delta)$ such that
$$p(\,\boldsymbol{\theta} \in T_k(\delta) \mid \mathbf{x}_0, \dots, \mathbf{x}_k, \mathbf{u}_0, \dots, \mathbf{u}_k\,) \geq 1 - \delta$$
- Similarly, can no longer reason about constraints being satisfied with probability 1, must work with *chance constraints*

Computing confidence sets

- Most common approach: assume noise is Gaussian, take Bayesian approach (i.e., compute a posterior distribution from which confidence sets can be computed)
  - Model: linearly parameterized or Gaussian process
- Frequentist approaches:
  - Statistical bootstrapping
  - If noise model sub-Gaussian, can use concentration inequalities (effectively yields same result as Gaussian confidence intervals)

Chance-constrained optimal control problem

$$J_0^*(\mathbf{x}_0) = \min_{\mathbf{u}_0,\dots,\mathbf{u}_{T-1}} p(\mathbf{x}_N) + \sum_{k=0}^{N-1} c(\mathbf{x}_k, \mathbf{u}_k)$$

subject to $\quad \mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + \mathbf{w}_k, \qquad k = 0, \dots, N-1$

$$\mathbf{w}_k \sim p(\mathbf{w}) \text{ i.i.d.}, k = 0, \dots, N-1$$

$$p(\mathbf{x}_k \in X\ \forall k) \geq 1 - \delta_x$$

$$p(\mathbf{u}_k \in U\ \forall k) \geq 1 - \delta_u$$

Chance-constrained optimal control problem

$$J_0^*(\mathbf{x}_0) = \min_{\mathbf{u}_0,\ldots,\mathbf{u}_{T-1}} p(\mathbf{x}_N) + \sum_{k=0}^{N-1} c(\mathbf{x}_k, \mathbf{u}_k)$$

subject to $\quad \mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k + \mathbf{w}_k, \qquad k = 0, \ldots, N-1$

$$\mathbf{w}_k \sim p(\mathbf{w}) \text{ i.i.d.}, k = 0, \ldots, N-1$$

$$\boxed{p(\mathbf{x}_k \in X\ \forall k) \geq 1 - \delta_x}$$

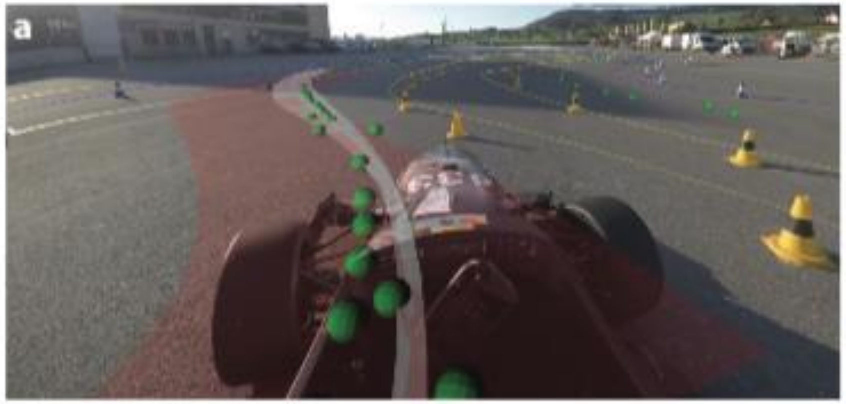
$$p(\mathbf{u}_k \in U\ \forall k) \geq 1 - \delta_u$$

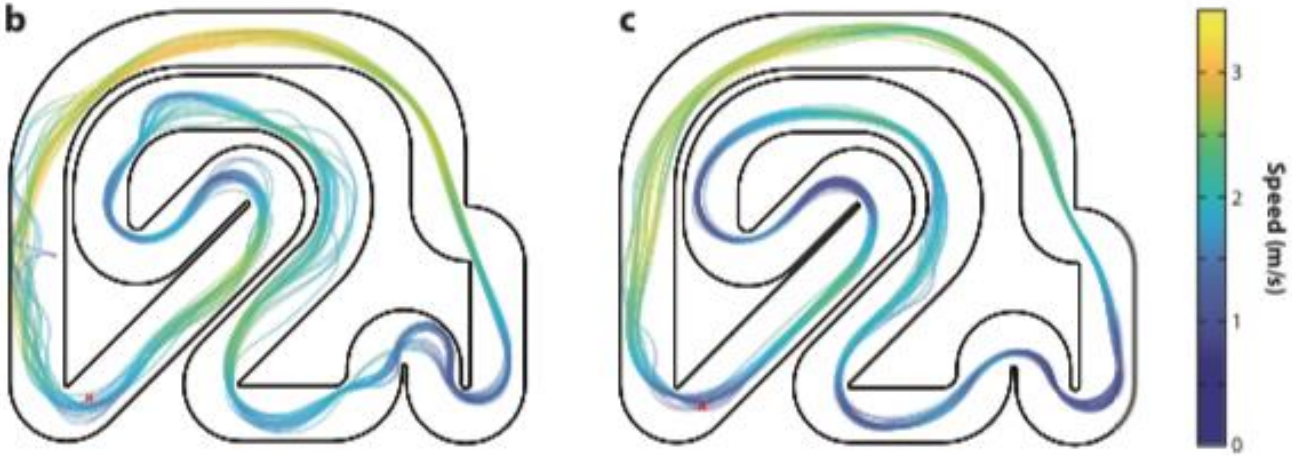
Much difficulty in chance-constrained trajectory optimization stems from even evaluating this “trajectory-wise” probability

A robust approach to stochastic control

- Simple set-theoretic computations of robust MPC are convenient
- Common approach: divide “risk” equally over timesteps, so at each time constraints must be satisfied with probability $1 - \frac{\delta}{2N}$
- Then guarantee that all parameters in confidence set $T_K\left(\frac{\delta}{2}\right)$ satisfy per-timestep chance constraints; better chance constraint satisfaction typically relies on Monte Carlo methods
- Typically over-conservative in practice
- Recursive feasibility arguments difficult

Application





As more and more laps are driven, the racecar is able to go faster as the dynamics are identified with high confidence (allowing for more aggressive control)

Hewing, Wabersich, Menner, Zeilinger, "Learning-Based Model Predictive Control: Toward Safe Learning in Control," 2019.

Learning the terminal constraint

- Line of work from Rosolia and Borrelli over multiple papers (2017-2020)
- Assume we have access to terminal control invariant X_f
- Know that including backward reachable set of X_f (i.e., $\mathrm{Pre}(X_f) \cup X_f$) is also invariant
- Therefore, given trajectory $\{\mathbf{x}(0), \dots, \mathbf{x}(N+1)\}$ such that $x(N+1) \in X_f$, know:

$$X_f \cup \{\mathbf{x}(0), \dots, \mathbf{x}(N)\}$$

  is control invariant.

Learning the terminal constraint

- Algorithm: assume access to a demonstration trajectory or stabilizing controller
- Initialize $X_f = \{0\}$ (assuming 0 is an equilibrium)
- Iterate over episodes k = 1,…
  - Each episode k yields data

$$D_k = \{\mathbf{x}_k(0), \dots, \mathbf{x}_k(N)\}, \;\; C_k = \{c(\mathbf{x}_k(0)), \dots, c(\mathbf{x}_k(N))\}$$

  - Expand terminal constraint via

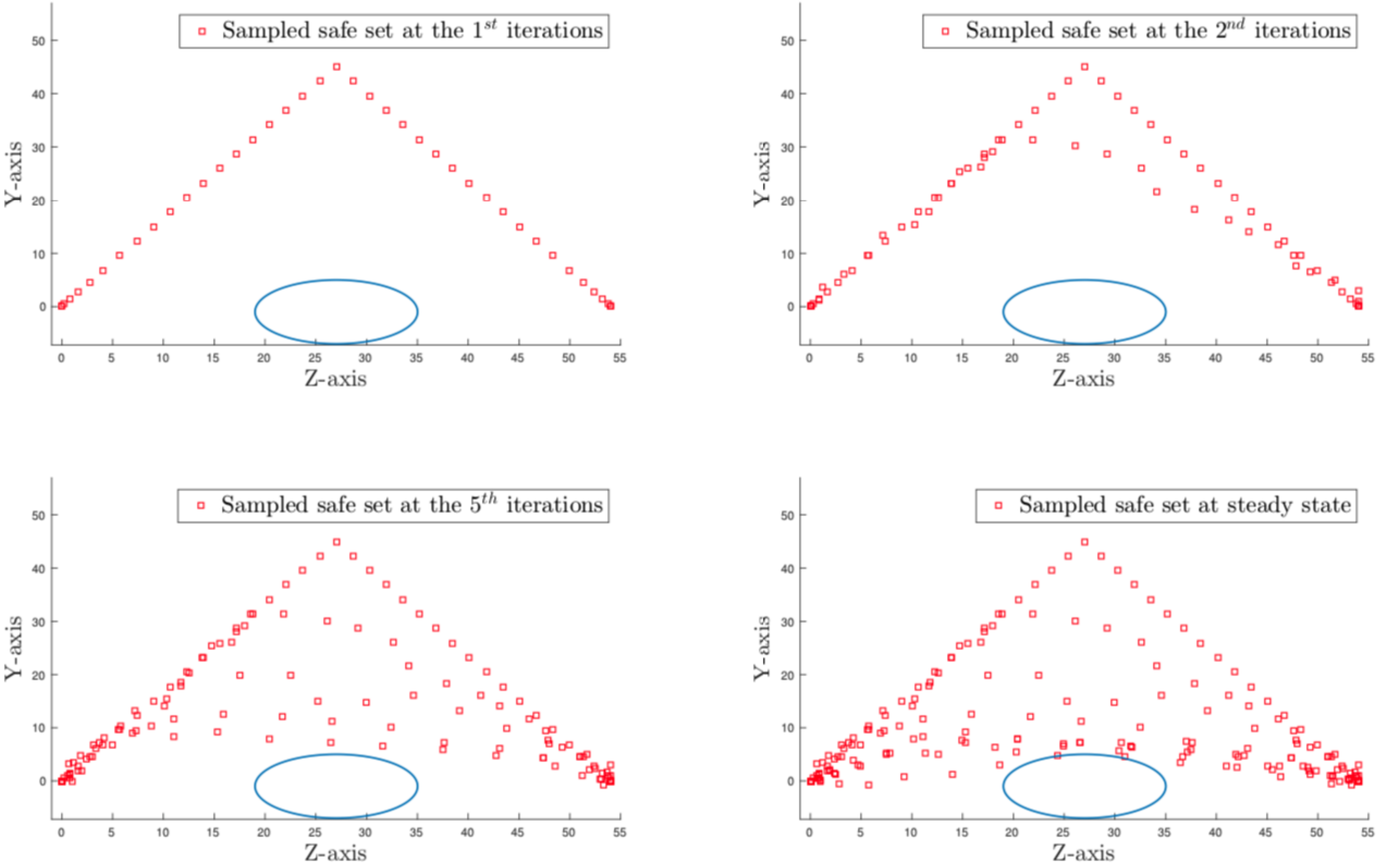
$$X_f \leftarrow X_f \cup D_k$$

  - Terminal cost $p(\mathbf{x})$ is the sum of all future costs from the last time that state was visited
  - Solve MPC problem with terminal constraint X_f and terminal cost $p(\mathbf{x})$

Learning the terminal constraint

- Can show that for systems without disturbances, this results in monotonic performance improvement.
- In practice, to make optimization problem tractable, use convex hull of sampled set and weighted sum of tail costs.
- Blanchini & Pellegrino (2005) showed that the convex hull of the sampled set is also control invariant for LTI systems!

Performance



Iteration	Iteration Cost
$j = 0$	65.000000000000000
$j = 1$	33.634529488066327
$j = 2$	24.216166714512450
$j = 3$	19.625000000001727
$j = 4$	19.625000000000004
$j = 5$	17.625000000022546
$j = 6$	17.625000000000000
$j = 7$	16.625000000000000
$j = 8$	16.625000000000000

Rosolia, Borelli, "Learning Model Predictive Control for Iterative Tasks. A Data-Driven Control Framework," TAC 2017.

Learning the terminal cost

- Important to also learn the terminal cost.
- Simple approach: use the tail cost from the previous visit to a given state

What else could we learn?

- Learn terminal cost: use, e.g., similar ideas to Q-learning
- Learn controller hyperparameters (e.g., planning horizon)
- Learn constraints (based on e.g., binary signals of constraint violation)
- Learning from demonstrations (behavioral cloning, imitation learning–not covered in this class but practically very useful)

Next time

- Unconstrained model-based methods in the tabular and nonlinear setting