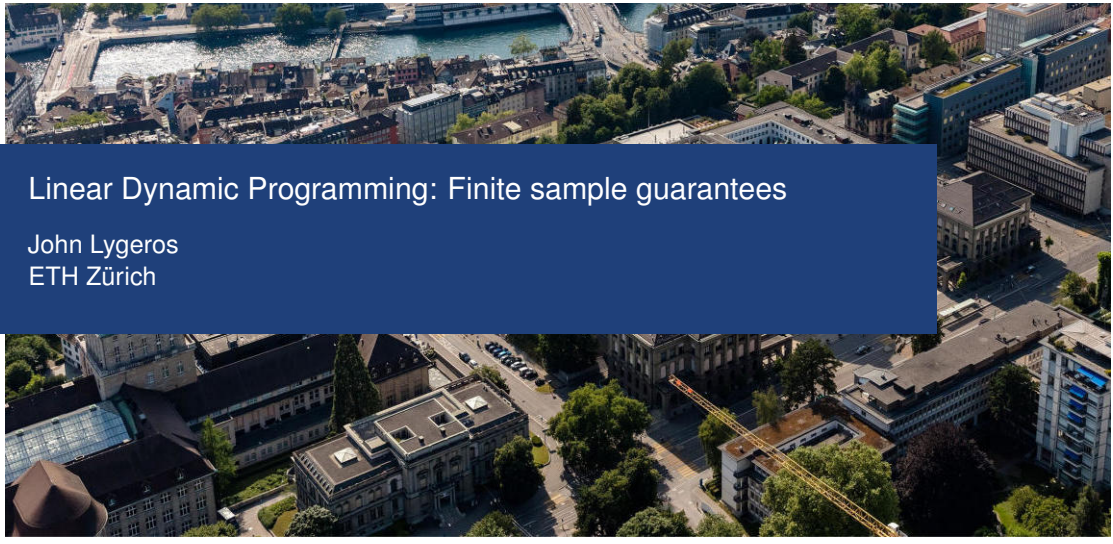
ETH zürich

AUTOMATIC CONTROL LABORATORY



# Linear Dynamic Programming: Finite sample guarantees

John Lygeros
ETH Zürich

Workshop on ”Trustworthy Data-Driven Control: From Finite Samples to Robust Guarantees”

# Acknowledgements

Thank you to

Andrea Martinelli

Lucia Falconi

Matilde Gargiani

And G. Banjac, P. Beuchat, A. Georghiou, P. Grontas, M. Kamgarpour, A. Kamoutsi, N. Kariotoglou, P. Mohajerin Esfahani, L. Pezzetti, N. Schmid, T. Summers, T. Sutter, ...

And the sponsors

NCCR Automation

erc

# Outline

# Dynamic Programming

**Value function characterisation**

- Fixed point of nonlinear operator

**Intractable in practice for state/action spaces with**

- High cardinality, or
- Infinite components

**Approximate dynamic programming**

- Restrict to subset defined by “features”, neural networks, . . .
- Provide guarantees on quality of approximation
- [Bertsekas, Tsitsiklis, van Roy, Powell, . . . ]

**Immediate connections to reinforcement learning**

# Linear programming approach

**Value function characterisation**

- Solution to a linear program
- Decision variables in space of functions
- Constraints parameterised by state-action pairs
- [Borkar, Hernandez-Lerma, Lasserre, . . . ]

**Elegant and theoretically powerful, but . . .**

- Large linear program for finite state/action spaces
- Infinite dimensional decision space and infinitely many constraints for infinite spaces

**Approximate dynamic programming**

- Restrict value function to space spanned by features and solve robust program
- e.g. Polynomial dynamics and function spaces $\rightarrow$ SDP [Lasserre]

# Data driven linear programming

**Solve a relaxed linear program based on data**

- Restrict value function to space spanned by features
- Approximate robust program by sampling constraints

**Some advantages**

- Completely data based
- Batch (instead of sequential) processing of data
- Standard linear program
- Approximation guarantees through the scenario programs

**and some drawbacks**

- Batch (instead of sequential) processing of data
- Theoretical guarantees not the end of the story . . .
- Novel persistency of excitation concept to avoid regularisation

# Outline

# Problem setup: stochastic optimal control

**Ingredients:**

- A discrete-time stochastic system $x^+ = f(x, u, \psi)$ with possibly infinite state & action spaces
- A stage-cost function $\ell : \mathbb{X} \times \mathbb{U} \to \mathbb{R}_+$

# Problem setup: stochastic optimal control

**Ingredients:**

- A discrete-time stochastic system $x^+ = f(x, u, \psi)$ with possibly infinite state & action spaces
- A stage-cost function $\ell : \mathbb{X} \times \mathbb{U} \to \mathbb{R}_+$

The $\gamma$-discounted $\infty$-horizon cost associated to a stationary feedback policy $\pi : \mathbb{X} \to \mathbb{U}$ is

$$v_\pi(x) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \ell(x_k, \pi(x_k)) \;\middle|\; x_0 = x\right]$$

# Problem setup: stochastic optimal control

**Ingredients:**

- A discrete-time stochastic system $x^+ = f(x, u, \psi)$ with possibly infinite state & action spaces
- A stage-cost function $\ell : \mathbb{X} \times \mathbb{U} \to \mathbb{R}_+$

The $\gamma$-discounted $\infty$-horizon cost associated to a stationary feedback policy $\pi : \mathbb{X} \to \mathbb{U}$ is

$$v_\pi(x) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \ell(x_k, \pi(x_k)) \;\middle|\; x_0 = x\right]$$

**Objective:** find an optimal policy $\pi^*$ such that $v_{\pi^*}(x) = \inf_\pi v_\pi(x) = v^*(x)$

# Dynamic programming methods

- The value function admits a recursive definition – the **Bellman equation**

$$v_\pi(x) = \underbrace{\ell(x, \pi(x)) + \gamma \mathbb{E}\big[v_\pi(f(x, \pi(x), \psi))\big]}_{(\mathcal{T}_\pi v_\pi)(x)}$$

$$v^*(x) = \underbrace{\inf_{u \in \mathbb{U}} \Big\{ \ell(x, u) + \gamma \mathbb{E}\big[v^*(f(x, u, \psi))\big] \Big\}}_{(\mathcal{T} v^*)(x)}$$

# Dynamic programming methods

- The value function admits a recursive definition – the **Bellman equation**

$$v_\pi(x) = \underbrace{\ell(x, \pi(x)) + \gamma \mathbb{E}\big[v_\pi(f(x, \pi(x), \psi))\big]}_{(\mathcal{T}_\pi v_\pi)(x)}$$

$$v^*(x) = \underbrace{\inf_{u \in \mathbb{U}} \Big\{ \ell(x, u) + \gamma \mathbb{E}\big[v^*(f(x, u, \psi))\big] \Big\}}_{(\mathcal{T} v^*)(x)}$$

- $\mathcal{T}$, $\mathcal{T}_\pi$ are monotone and contractive

# Dynamic programming methods

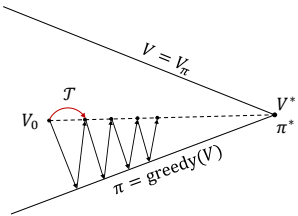
- The value function admits a recursive definition – the **Bellman equation**

$$v_\pi(x) = \underbrace{\ell(x, \pi(x)) + \gamma \mathbb{E}\big[v_\pi(f(x, \pi(x), \psi))\big]}_{(\mathcal{T}_\pi v_\pi)(x)}$$

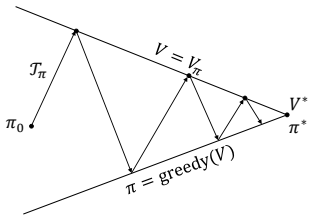
$$v^*(x) = \underbrace{\inf_{u \in \mathbb{U}} \Big\{ \ell(x, u) + \gamma \mathbb{E}\big[v^*(f(x, u, \psi))\big] \Big\}}_{(\mathcal{T} v^*)(x)}$$

- $\mathcal{T}$, $\mathcal{T}_\pi$ are monotone and contractive

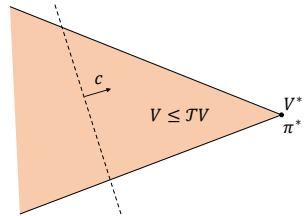
**Value Iteration**




**Policy Iteration**



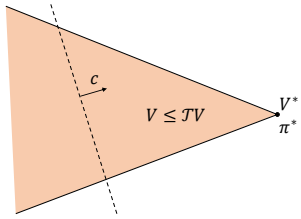

**Linear Programming**



# The linear programming formulation

One can find $v^*$ by solving the $\infty$-dimensional (nonlinear) program

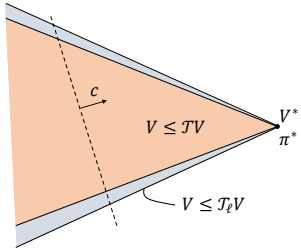
$$\sup_{v \in \mathbb{V}} \int_{\mathbb{X}} v(x) c(dx)$$
$$\text{s.t.} \quad v(x) \leq (\mathcal{T}v)(x) \quad \forall x,$$



# The linear programming formulation

One can find $v^*$ by solving the $\infty$-dimensional (nonlinear) program

$$\sup_{v \in \mathbb{V}} \int_{\mathbb{X}} v(x) c(dx)$$
$$\text{s.t.} \quad v(x) \leq (\mathcal{T}v)(x) \quad \forall x,$$




Can obtain linear program by substituting

$$v(x) \leq (\mathcal{T}v)(x) = \inf_{u \in \mathbb{U}} \Big\{ \ell(x,u) + \gamma \mathbb{E}\big[v(f(x,u,\psi))\big] \Big\} \quad \forall x$$

with

$$v(x) \leq (\mathcal{T}_\ell v)(x,u) = \ell(x,u) + \gamma \mathbb{E}\big[v(f(x,u,\psi))\big] \quad \forall (x,u)$$

# Observations

**The measure used in the cost does not matter much**

- Optimal value function obtained for any $c(\cdot)$
- Usually interpreted as initial condition distribution

# Observations

**The measure used in the cost does not matter much**

- Optimal value function obtained for any $c(\cdot)$
- Usually interpreted as initial condition distribution

**For finite state/action spaces**

- Standard finite dimensional linear program
- Decision variable $v(\cdot)$ vector containing value of each state
- Constraints parametrised by (finite) combinations of states and actions

# Observations

**The measure used in the cost does not matter much**

- Optimal value function obtained for any $c(\cdot)$
- Usually interpreted as initial condition distribution

**For finite state/action spaces**

- Standard finite dimensional linear program
- Decision variable $v(\cdot)$ vector containing value of each state
- Constraints parametrised by (finite) combinations of states and actions

**Linear programs convex but . . .**

- Can be more work to solve than policy/value iteration for large finite state-action spaces
- Infinite dimensional for continuous state/action spaces

# Observations

**The measure used in the cost does not matter much**

- Optimal value function obtained for any $c(\cdot)$
- Usually interpreted as initial condition distribution

**For finite state/action spaces**

- Standard finite dimensional linear program
- Decision variable $v(\cdot)$ vector containing value of each state
- Constraints parametrised by (finite) combinations of states and actions

**Linear programs convex but . . .**

- Can be more work to solve than policy/value iteration for large finite state-action spaces
- Infinite dimensional for continuous state/action spaces

**Model based**

- LP uses $l(x, u)$ and $f(x, u, \psi)$

# The $Q$-function

- If one is able to obtain $v^*$, then

$$\pi^*(x) \in \arg\min_{u \in \mathbb{U}} \left\{ \ell(x,u) + \gamma \mathbb{E}\left[ v^*(f(x,u,\psi)) \right] \right\}.$$

**Problem:** policy extraction is in general not possible if $f$ or $\ell$ are not known

# The $Q$-function

- If one is able to obtain $v^*$, then

$$\pi^*(x) \in \arg\min_{u \in \mathbb{U}} \left\{ \ell(x,u) + \gamma \mathbb{E}\left[ v^*(f(x,u,\psi)) \right] \right\}.$$

  **Problem:** policy extraction is in general not possible if $f$ or $\ell$ are not known

- $Q$-functions

$$q^*(x,u) = \underbrace{\ell(x,u) + \gamma \mathbb{E}\left[ \inf_{w \in \mathbb{U}} q^*(f(x,u,\psi),w) \right]}_{(\mathcal{F}q^*)(x,u)}$$

# The $Q$-function

- If one is able to obtain $v^*$, then

$$\pi^*(x) \in \arg\min_{u \in \mathbb{U}} \left\{ \ell(x,u) + \gamma \mathbb{E}\left[ v^*(f(x,u,\psi)) \right] \right\}.$$

  **Problem:** policy extraction is in general not possible if $f$ or $\ell$ are not known

- $Q$-functions

$$q^*(x,u) = \underbrace{\ell(x,u) + \gamma \mathbb{E}\left[ \inf_{w \in \mathbb{U}} q^*(f(x,u,\psi),w) \right]}_{(\mathcal{F}q^*)(x,u)}$$

- Because $v^*(x) = \min_{u \in \mathbb{U}} q^*(x,u)$ model-free policy extraction:

$$\pi^*(x) \in \arg\min_{u \in \mathbb{U}} q^*(x,u).$$

# LP formulation for $Q$-functions

- $\mathcal{F}$ is monotone contraction mapping, hence

$$\sup_{q \in \mathbb{Q}} \int_{\mathbb{X} \times \mathbb{U}} q(x,u) c(dx,du)$$

$$\text{s.t. } q(x,u) \leq (\mathcal{F}q)(x,u) = \ell(x,u) + \gamma \mathbb{E}\left[ \inf_{w \in \mathbb{U}} q(f(x,u,\psi),w) \right] \quad \forall (x,u)$$

**Problem:** We can not relax the constraints due to nesting of $\mathbb{E}$ and $\inf$

# LP formulation for $Q$-functions

- $\mathcal{F}$ is monotone contraction mapping, hence

$$\sup_{q \in \mathbb{Q}} \int_{\mathbb{X} \times \mathbb{U}} q(x,u) c(dx, du)$$
$$\text{s.t. } q(x,u) \leq (\mathcal{F}q)(x,u) = \ell(x,u) + \gamma \mathbb{E}\left[ \inf_{w \in \mathbb{U}} q(f(x,u,\psi), w) \right] \quad \forall (x,u)$$

**Problem:** We can not relax the constraints due to nesting of $\mathbb{E}$ and inf

- **Solution 1**: Double the decision variables by re-introducing the value function

$$\sup_{q \in \mathbb{Q}, v \in \mathbb{V}} \int_{\mathbb{X} \times \mathbb{U}} q(x,u) c(dx, du)$$
$$\text{s.t. } q(x,u) \leq \ell(x,u) + \gamma \mathbb{E}\left[ v(f(x,u,\psi)) \right] \quad \forall (x,u)$$
$$v(x) \leq q(x,u) \quad \forall (x,u)$$

- [Cogill,Rotkowitz, van Roy, Lall], [Beuchat et.al. 2020]

# Solution 2: Relaxed Bellman operator for $Q$-functions

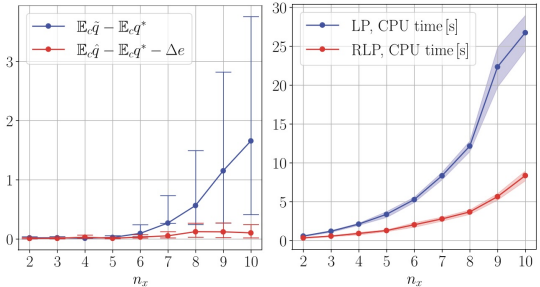
- Interchange $\mathbb{E}$ and inf

$$(\hat{\mathcal{F}}q)(x,u) = \ell(x,u) + \gamma \inf_{w\in\mathbb{U}} \mathbb{E}\big[q(f(x,u,\psi),w)\big]$$

- Leading to relaxed linear program

$$\sup_{q\in\mathbb{Q}} \int_{\mathbb{X}\times\mathbb{U}} q(x,u)c(dx,du)$$

$$\text{s.t.} \quad q(x,u) \le \ell(x,u) + \gamma\mathbb{E}\big[q(f(x,u,\psi),w)\big] \quad \forall (x,u,w)$$



# Relaxed Bellman operator properties

**Proposition ([Martinelli et.al. 2022])**

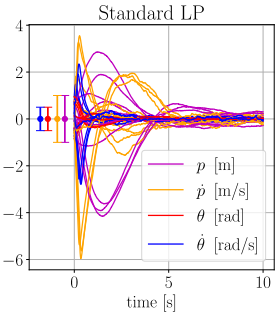
(1) *$\hat{\mathcal{F}}$ is a monotone contraction mapping with a unique fixed point $\hat{q}(x, u)$*

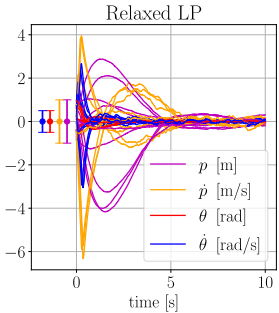
(2) *The fixed point of $\hat{\mathcal{F}}$ is a point-wise upper bound to the fixed point of $\mathcal{F}$*

(3) *For deterministic systems fixed points coincide*

(4) *For stochastic affine dynamics and quadratic cost optimal policies coincide*

- For nonlinear systems, empirical evidence is encouraging



# Outline

# Solving the LP

Focus on relaxed LP formulation

$$\sup_{q \in \mathbb{Q}} \int_{\mathbb{X} \times \mathbb{U}} q(x,u) c(dx,du)$$
$$\text{s.t.} \quad q(x,u) \leq \ell(x,u) + \gamma \mathbb{E}\big[q(f(x,u,\psi),w)\big] \quad \forall (x,u,w)$$

Convex optimisation problems can be solved efficiently in principle, but ...

For continuous states/actions

- Decision variable $q$ takes values in $\infty$-dimensional space
- Number of constraints is infinite (parameterized by states $\times$ actions $\times$ actions)
- Expectation difficult to compute

Model based, uses $l(x,u)$ and $f(x,u,\psi)$

# From infinite to semi-infinite

Restrict attention to finitely parameterised set of functions

# From infinite to semi-infinite

Restrict attention to finitely parameterised set of functions

To maintain linearity consider linear subspace spanned by family of features

$$\mathbb{Q}^{\Phi} = \left\{ q \in \mathbb{Q} \mid q(x,u) = \sum_{i=1}^{r} \alpha_i \phi_i(x,u) = \Phi\alpha, \text{ with } \alpha \in \mathbb{R}^r \right\}$$

# From infinite to semi-infinite

Restrict attention to finitely parameterised set of functions

To maintain linearity consider linear subspace spanned by family of features

$$\mathbb{Q}^{\Phi} = \left\{ q \in \mathbb{Q} \mid q(x,u) = \sum_{i=1}^{r} \alpha_i \phi_i(x,u) = \Phi\alpha, \text{ with } \alpha \in \mathbb{R}^r \right\}$$

Resulting linear program has finite number of decision variables

$$\begin{aligned} &\sup_{\alpha \in \mathbb{R}^r} \left[ \int_{\mathbb{X}\times\mathbb{U}} \Phi(x,u) c(dx,du) \right] \alpha \\ &\text{s.t.} \quad \Phi(x,u)\alpha \leq \ell(x,u) + \gamma \mathbb{E}\left[\Phi(f(x,u,\psi),w)\right]\alpha \quad \forall (x,u,w) \end{aligned}$$

# From infinite to semi-infinite

Restrict attention to finitely parameterised set of functions

To maintain linearity consider linear subspace spanned by family of features

$$\mathbb{Q}^{\Phi} = \left\{ q \in \mathbb{Q} \mid q(x,u) = \sum_{i=1}^{r} \alpha_i \phi_i(x,u) = \Phi\alpha, \text{ with } \alpha \in \mathbb{R}^r \right\}$$

Resulting linear program has finite number of decision variables

$$\begin{aligned} &\sup_{\alpha \in \mathbb{R}^r} \left[ \int_{\mathbb{X}\times\mathbb{U}} \Phi(x,u) c(dx,du) \right] \alpha \\ &\text{s.t.} \quad \Phi(x,u)\alpha \leq \ell(x,u) + \gamma \mathbb{E}\left[\Phi(f(x,u,\psi),w)\right]\alpha \quad \forall (x,u,w) \end{aligned}$$

Notes

- Robust linear program
- Cost involves moments of features with respect to $c(\cdot)$
- Optimal solution now depends on $c(\cdot) \longrightarrow$ importance weights, design choice

# From semi-infinite to finite

In model based setting can be tackled with robust LP methods, dualisation, ...

Or simply by sampling a finite number of constraints

Also works with just data on system trajectory snippets

$$\mathcal{D} = \left\{(x_i, u_i, l_i, x_i^+)\right\}_{i=1}^{N}, \text{ plus arbitrary } w_i, \ i = 1, \ldots, N$$

with $l_i = l(x_i, u_i)$ and $x_i^+ = f(x_i, u_i, \psi_i)$ for some noise realisation $\psi_i$

# From semi-infinite to finite

In model based setting can be tackled with robust LP methods, dualisation, ...

Or simply by sampling a finite number of constraints

Also works with just data on system trajectory snippets

$$\mathcal{D} = \left\{(x_i, u_i, l_i, x_i^+)\right\}_{i=1}^N, \text{ plus arbitrary } w_i, \ i = 1, \ldots, N$$

$$\text{with } l_i = l(x_i, u_i) \text{ and } x_i^+ = f(x_i, u_i, \psi_i) \text{ for some noise realisation } \psi_i$$

For deterministic systems simply [Banjac, JL 2019]

$$\begin{aligned} &\sup_{\alpha \in \mathbb{R}^r} \left[\int_{\mathbb{X}\times\mathbb{U}} \Phi(x,u) c(dx, du)\right] \alpha \\ &\text{s.t.} \quad \Phi(x_i, u_i)\alpha \leq l_i + \gamma \Phi(f(x_i, u_i), w_i)\alpha \quad \forall i = 1 \ldots, N \end{aligned}$$

# From semi-infinite to finite

In model based setting can be tackled with robust LP methods, dualisation, ...

Or simply by sampling a finite number of constraints

Also works with just data on system trajectory snippets

$$\mathcal{D} = \left\{(x_i, u_i, l_i, x_i^+)\right\}_{i=1}^{N}, \text{ plus arbitrary } w_i, \ i = 1, \ldots, N$$

$$\text{with } l_i = l(x_i, u_i) \text{ and } x_i^+ = f(x_i, u_i, \psi_i) \text{ for some noise realisation } \psi_i$$

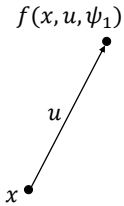
For deterministic systems simply [Banjac, JL 2019]

$$\begin{aligned} &\sup_{\alpha \in \mathbb{R}^r} \left[\int_{\mathbb{X}\times\mathbb{U}} \Phi(x,u) c(dx, du)\right] \alpha \\ &\text{s.t.} \quad \Phi(x_i, u_i)\alpha \leq l_i + \gamma \Phi(f(x_i, u_i), w_i)\alpha \quad \forall i = 1 \ldots, N \end{aligned}$$

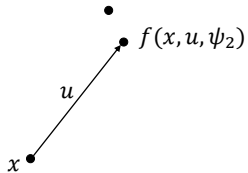
For stochastic systems also need to deal with expectation

# Approximating expectation for stochastic systems

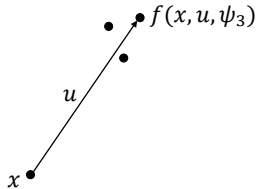
# Approximating expectation for stochastic systems



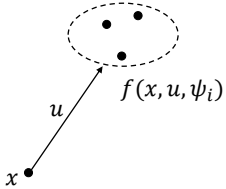
# Approximating expectation for stochastic systems



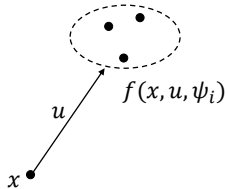
# Approximating expectation for stochastic systems



# Approximating expectation for stochastic systems



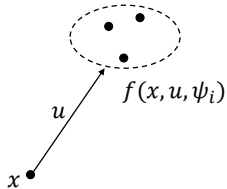
# Approximating expectation for stochastic systems

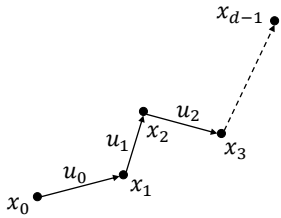



Re-initialise system, collect *M* samples
Monte-Carlo estimate

- Unbiased estimator
- Concentration bounds
- [Sutter et.al. 2017]

# Approximating expectation for stochastic systems

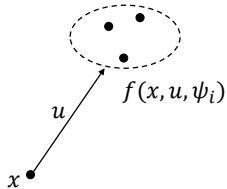


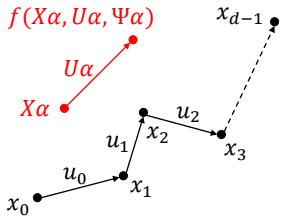




Re-initialise system, collect *M* samples
Monte-Carlo estimate

- Unbiased estimator
- Concentration bounds
- [Sutter et.al. 2017]

# Approximating expectation for stochastic systems




Re-initialise system, collect $M$ samples
Monte-Carlo estimate

- Unbiased estimator
- Concentration bounds
- [Sutter et.al. 2017]

$$\bar{\theta}_{X\alpha,U\alpha} = q(f(X\alpha, U\alpha, \Psi\alpha), W\alpha)$$

$$X = \begin{bmatrix} x_0 & \cdots & x_{d-1} \end{bmatrix}, U = \begin{bmatrix} u_0 & \cdots & u_{d-1} \end{bmatrix},$$
$$\Psi = \begin{bmatrix} \psi_0 & \cdots & \psi_{d-1} \end{bmatrix}, W = \begin{bmatrix} w_0 & \cdots & w_{d-1} \end{bmatrix}$$

$\bar{\theta}_{X\alpha,U\alpha}$ is biased

We do not know $f(X\alpha, U\alpha, \Psi\alpha)$

$X^+\alpha \neq f(X\alpha, U\alpha, \Psi\alpha)$ for $X^+ = \begin{bmatrix} x_0^+ & \cdots & x_{d-1}^+ \end{bmatrix}$

# Stochastic linear-quadratic systems

For linear or affine dynamics and generalised quadratic cost

- Linear combinations of transitions respect dynamics
- (Under an additional constraint in the case of affine systems)
- With adapted noise statistics

Moreover . . .

**Proposition ([Martinelli et.al. 2022])**

(1) *The LP with biased constraints is policy-preserving*

(2) *If the data matrix* $\begin{bmatrix} X \\ U \\ \mathbf{1}^\top \end{bmatrix}$ *is full row-rank then we can construct constraints for all* $(x, u)$

(3) *For deterministic affine systems (2) equivalent to a PE condition on the input*

# Putting it all together

## For deterministic systems

- Select features $\phi_i(x, u),\ i = 1 \ldots, r$
- Sample data $\mathcal{D} = \left\{(x_i, u_i, l_i, x_i^+)\right\}_{i=1}^N$, plus arbitrary $w_i,\ i = 1, \ldots, N$
- Solve

$$\sup_{\alpha \in \mathbb{R}^r} \left[\int_{\mathbb{X}\times\mathbb{U}} \Phi(x, u) c(dx, du)\right] \alpha$$
$$\text{s.t.} \quad \Phi(x_i, u_i)\alpha \leq l_i + \gamma \Phi(x_i^+, w_i)\alpha \quad i = 1 \ldots, N$$

- Extract $\pi(x) \in \arg\min_{u\in\mathbb{U}} \Phi(x, u)\alpha^*$
- Implement on the system and hope for the best!

## For stochastic systems

- Additionally approximate the expectation in the constraints

# What can we prove?

"Regularised" deterministic problem

$$\sup_{\|\alpha\| \leq \theta} \left[ \int_{\mathbb{X} \times \mathbb{U}} \Phi(x,u) c(dx,du) \right] \alpha$$

$$\text{s.t.} \quad \Phi(x_i,u_i)\alpha \leq l_i + \gamma \Phi(x_i^+, w_i)\alpha \quad i = 1 \ldots, N$$

**Theorem ([Mohajerin Esfahani et.al. 2018])**

*(Under some technical conditions, e.g. compactness of input and action spaces)*

$$\left| \int_{\mathbb{X} \times \mathbb{U}} (q^*(x,u) - \Phi(x,u)\alpha^*)\, c(dx,du) \right| \leq h(r,\theta,N)$$

*with* $h(r,\theta,N) \to 0$ *as* $r,\theta,N \to \infty$

# Observations

**Theorem in [Mohajerin Esfahani et.al. 2018]**

- Value functions of stochastic systems with exact expectation
- Approximation of expectation in [Sutter et.al. 2017], $h(r, \theta, N, M)$

# Observations

**Theorem in [Mohajerin Esfahani et.al. 2018]**

- Value functions of stochastic systems with exact expectation
- Approximation of expectation in [Sutter et.al. 2017], $h(r, \theta, N, M)$

**Combination of**

- Theorem on approximation of infinite dimensional LP
- Value bound for scenario programs from [Mohajerin Esfahani et.al. 2015]
- Important details and caveats omitted: Choice of norms, functional form of bound, ...

# Observations

**Theorem in [Mohajerin Esfahani et.al. 2018]**

- Value functions of stochastic systems with exact expectation
- Approximation of expectation in [Sutter et.al. 2017], $h(r, \theta, N, M)$

**Combination of**

- Theorem on approximation of infinite dimensional LP
- Value bound for scenario programs from [Mohajerin Esfahani et.al. 2015]
- Important details and caveats omitted: Choice of norms, functional form of bound, ...

**What goes into the bound?**

- Richness of function approximation $(r, \theta)$
- Number of constraints $N$
- Bound sub-optimality (due to $(r, \theta)$) and super-optimality (due to $N$)

# Observations

**The bound shows that**

- As the function approximation gets richer $(r, \theta) \to \infty \ldots$
- and the number of sampled constraints increases $N \to \infty \ldots$
- (and the estimate of the expectation gets better $M \to \infty \ldots$)
- our estimate of the optimal cost converges to the real optimal cost

# Observations

**The bound shows that**

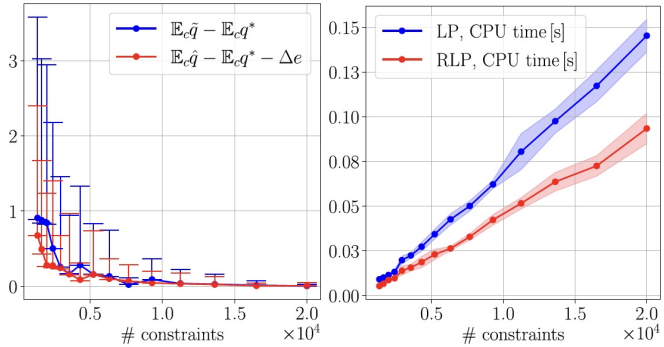
- As the function approximation gets richer $(r, \theta) \to \infty \ldots$
- and the number of sampled constraints increases $N \to \infty \ldots$
- (and the estimate of the expectation gets better $M \to \infty \ldots$)
- our estimate of the optimal cost converges to the real optimal cost

**The bound says nothing about**

- The performance of the greedy policy $\pi(x) \in \arg\min_{u \in \mathbb{U}} \Phi(x, u)\alpha^*$
- Computational efficiency
- Curse of dimensionality in scenario bound of [Mohajerin Esfahani et.al. 2015]

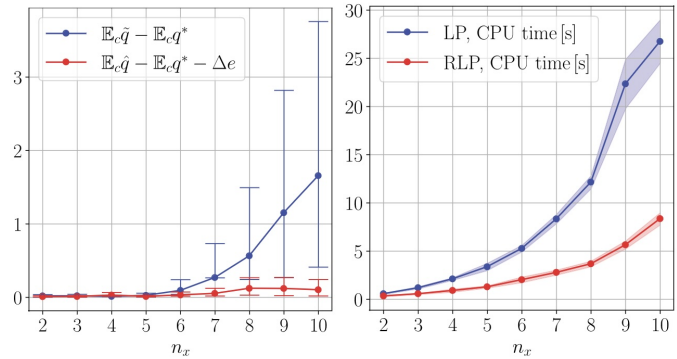
# In action

2-state, 1-input stochastic linear system, Monte-Carlo expectation



# In action

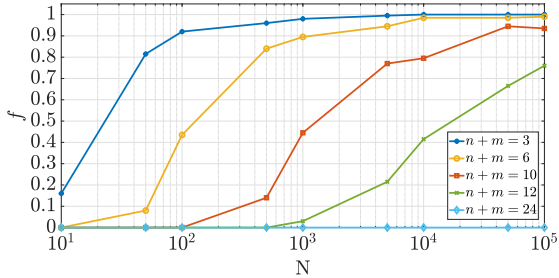
2-input stochastic linear systems, increasing state dimension, Monte Carlo expectation



# But . . .

Regulariser can introduce bias

Removing regulariser can lead to unbounded programs



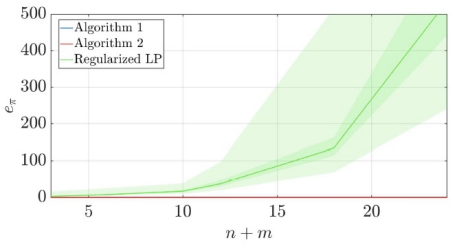

Asymptotically: Boundedness by classical PE conditions [Lu & Meyn 2023]

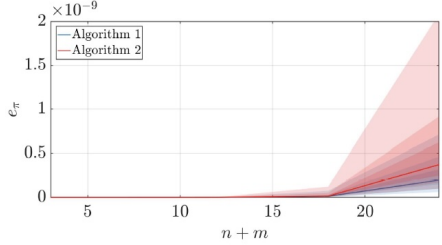
Can we say something with finite samples?

# Safely removing the regulariser

## Exploit degree of freedom afforded by $c(\cdot)$

- Select $c(\cdot)$ as a function of data
- Possibly add targeted [Falconi et.al. 2025] or auxiliary [Martinelli et.al. 2026?] samples
- Ensure LP is bounded without regularisation

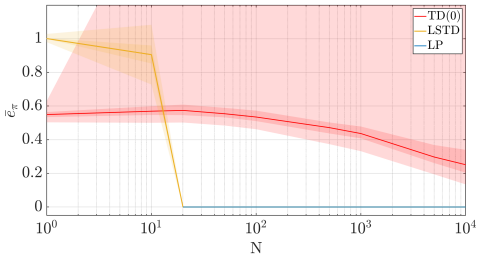


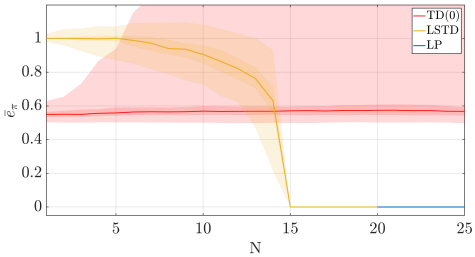




* Q-function associated with specific policy [Falconi et.al. 2025]

# Competitive with RL for linear systems

3 state, 2 input random linear systems

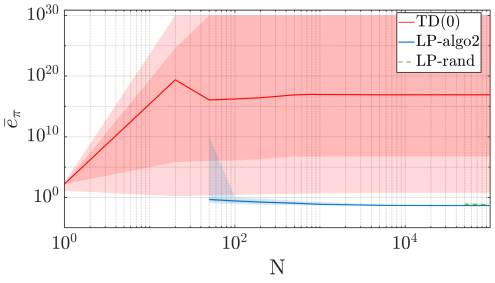


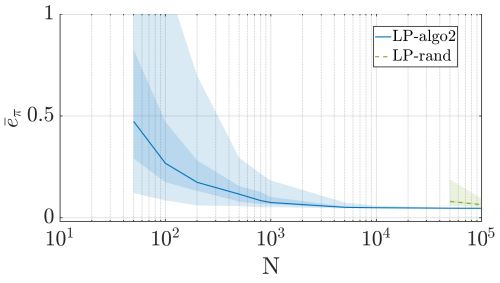




* Q-function associated with specific policy [Falconi et.al. 2025]

# Competitive with RL for nonlinear systems

## Cart-pole system




* Q-function associated with specific policy [Falconi et.al. 2025]

# Outline

# Potential benefits

## Versatile

- One-shot LP for optimal policy
- Policy evaluation LP for PI
- Regularisation, additional constraints, choice of $c(\cdot)$, auxiliary input $w_i, \ldots$

## Processes data in bulk

- As opposed to sequentially
- Double-edged sword: No natural data compression

## Basis for iterative schemes

- Maintain support constraints
- E.g. form previous PI iterates

# Potential pitfalls

Sensitive to constraint sampling

- Care needed to obtain even a bounded solution
- A non-obvious PE condition?

Regularisation

- Needed for theoretical results
- Can bias solution and obscure problems

At the end of the day, we still have to solve a large LP

- Curse of dimensionality for general systems

Performance of resulting greedy policy requires further analysis

# My thanks to

Andrea Martinelli

Lucia Falconi

Matilde Gargiani

And G. Banjac, P. Beuchat, A. Georghiou, P. Grontas, M. Kamgarpour, A. Kamoutsi, N. Kariotoglou, P. Mohajerin Esfahani, L. Pezzetti, N. Schmid, T. Summers, T. Sutter, ...

And the sponsors

NCCR Automation

erc

Thank you!

# References


- Falconi, Martinelli, Lygeros, “Data-driven optimal control via linear programming: boundedness guarantees”, *IEEE TAC*, 2025
- Martinelli, Gargiani, Draskovic, Lygeros, “Data- driven optimal control of affine systems: A linear programming perspective”, *IEEE L-CSS*, 2022
- Martinelli, Gargiani, Lygeros, “Data-driven optimal control with a relaxed linear program”, *Automatica*, 2022
- P. Beuchat, A. Georghiou, and J. Lygeros, “Performance guarantees for model-based approximate dynamic programming in continuous spaces,” *IEEE TAC*, 2020.
- G. Banjac and J. Lygeros, “A data-driven policy iteration scheme based on linear programming,” in *IEEE CDC*, 2019.
- Mohajerin Esfahani, Sutter, Kuhn, and Lygeros, “From infinite to finite programs: Explicit error bounds with applications to approximate dynamic programming,” *SIAM JoO*, 2018.
- Sutter, Kamoutsi, Mohajerin Esfahani, and Lygeros, “Data-driven approximate dynamic programming: A linear programming approach,” in *IEEE CDC*, 2017.