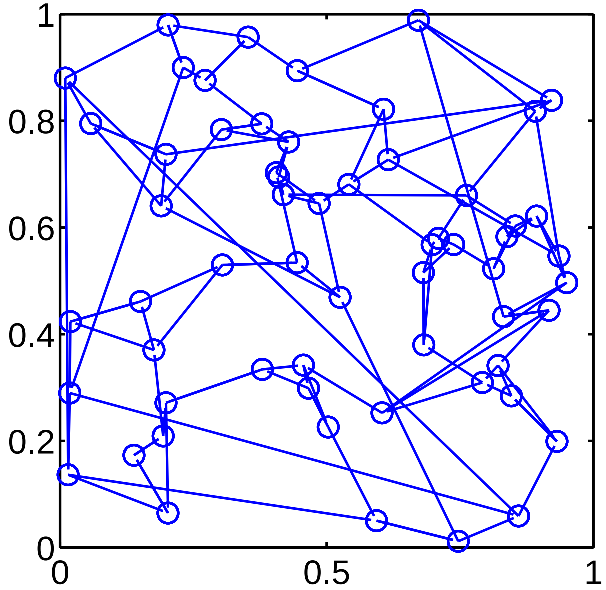
Hybrid control using approximate dynamic programming — approaching large problems

Anders Rantzer
LTH
Lund University, Sweden

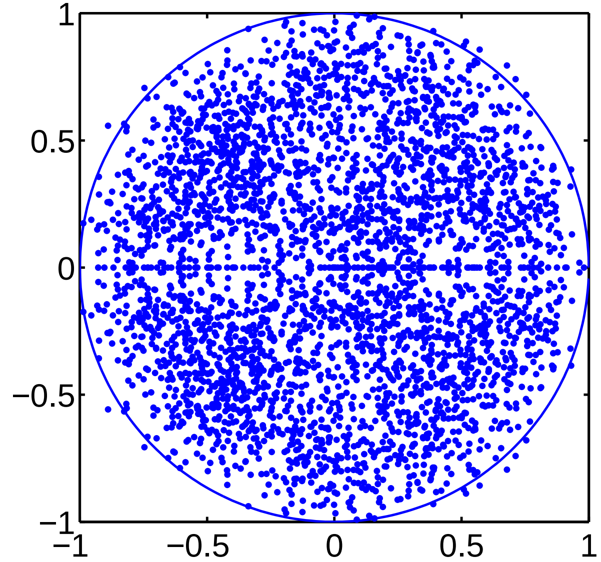
With contributions by
Sven Hedlund, Bo Lincoln, Stefan Solyom, Andreas Wernrud and Stephen Prajna

Optimal control: 60 discrete states, 30 continuous

120 edges



120×30 eigenvalues



Minimize

$$\sum_k z^T Q_{iu} z$$

Continuous dynamics:	$z(k+1) = A_{i(k)u(k)} z(k)$	$z(0) = z_0 \in \mathrm{R}^{30}$
Discrete jumps:	$i(k+1) = u(k)$	$i(0) = i_0$

Four steps of approximate value iteration

After four iterations we have one 30×30 matrix P^i for each node such that the following switch law is within a factor 3.81 from optimality:

$$\begin{cases} \text{Jump to node } n & \text{if } z^T[A_{in}^T P^n A_{in} + Q_{in}]z < z^T[A_{im}^T P^m A_{im} + Q_{im}]z \\ \text{Jump to node } m & \text{else} \end{cases}$$

Main messages of today

- There is a rich set of optimal control problems which have simple approximative solutions
- There are algorithms that find such solutions whenever they exist

Contents

- ○ Introduction
- • **Approximative dynamic programming**
- ○ If simple approximation exists, we will find one!
- ○ Duality and reachability
- ○ Conclusions

Who decides the price of a Volvo?

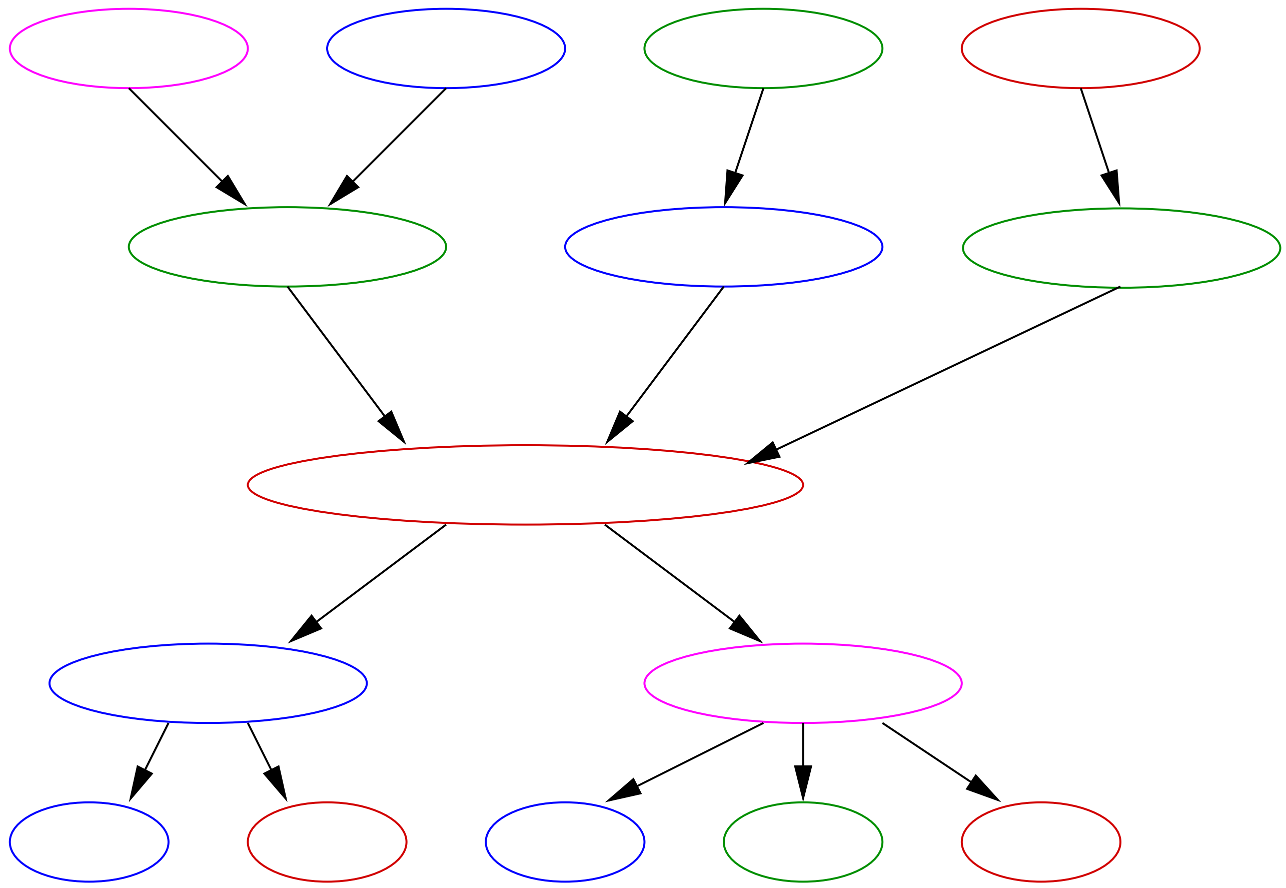
Subcontractor

Subcontractor

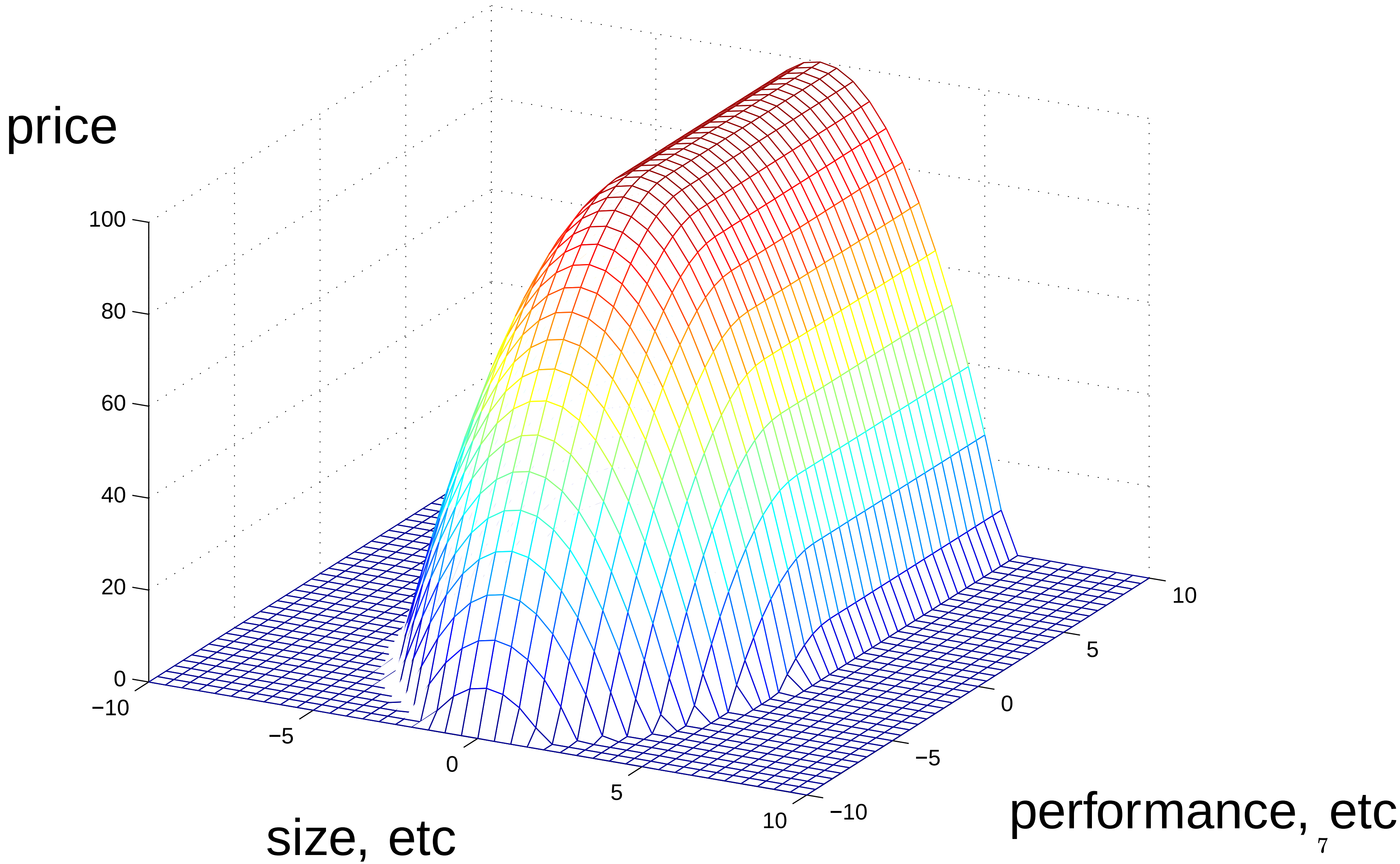
Car manufacturer

Car dealer

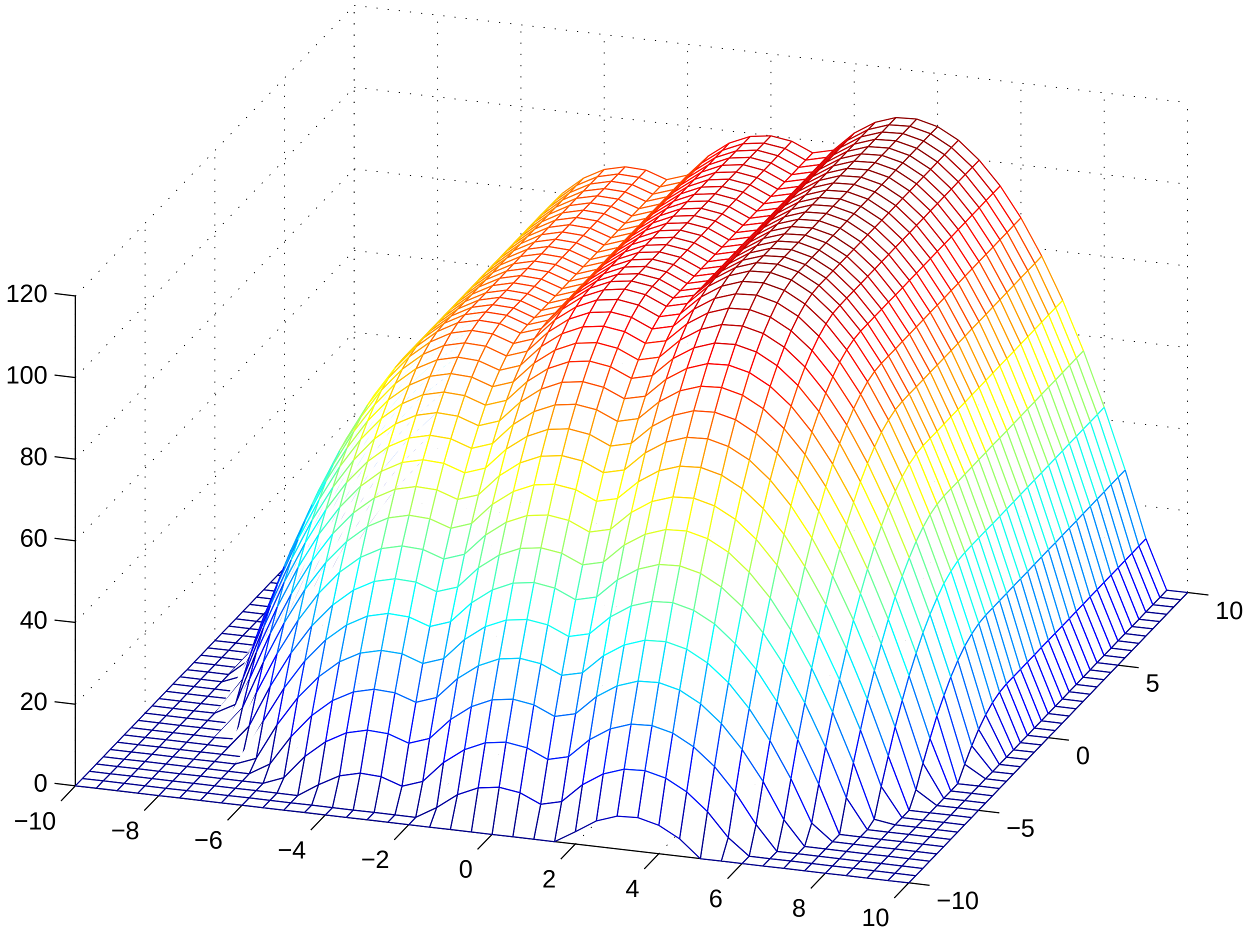
Customer



Valuation by the customer



Valuation by the car dealer



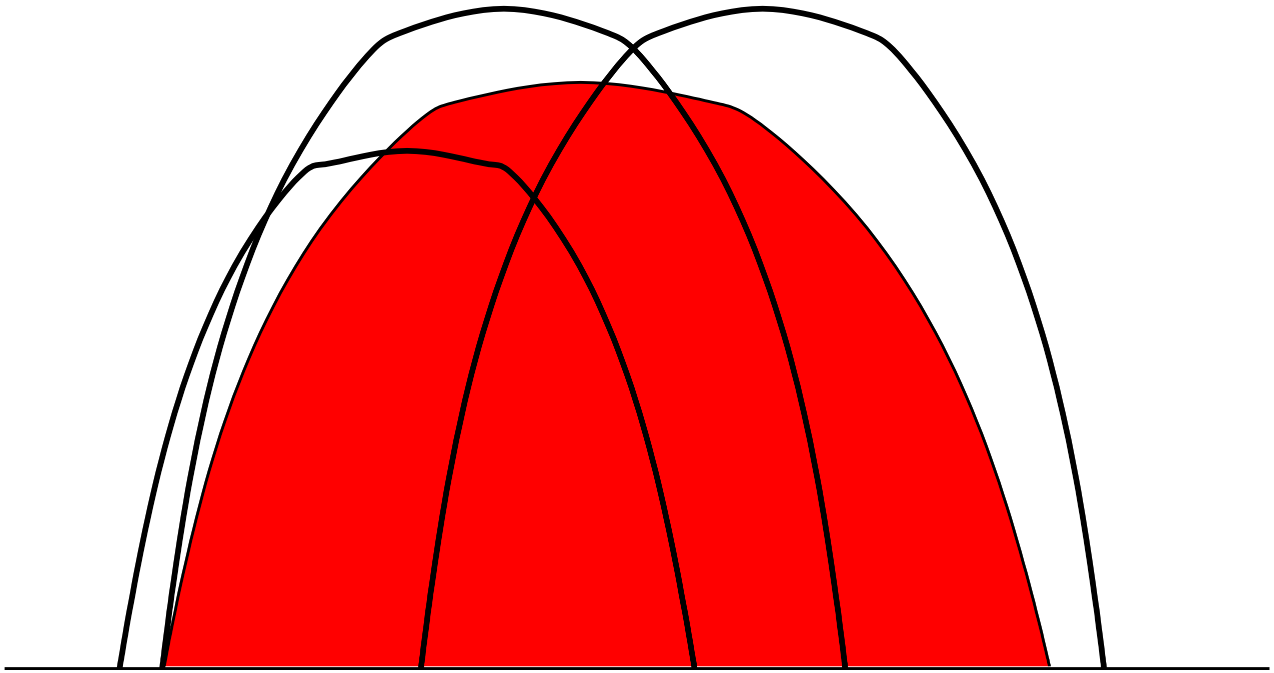
Customers: Andersson, Pettersson and Lundström

The key: Simplified valuation

Exact value-iteration gives absurd complexity.

Every subcontractor of Volvo would have to modify his prices when Andersson expands his garage.

Of course, pricing is not done like that.
Approximations are done in every step.



Dynamic programming in discrete time

Minimize $\displaystyle\sum_{k=0}^{\infty} l(x(k), u(k))$

subject to $x(k+1) = f(x(k), u(k)) \qquad k = 0, 1, 2, \ldots$

$x(0) = x_0$

Given x_0, let $V^*(x_0)$ denote the minimal value. The value function V^* satisfies the *Bellman equation*

$$V^*(x) = \min_u \left[V^*(f(x,u)) + l(x,u)\right]$$

In some cases V^* can be computed by recursive iteration:

$$V_{j+1}(x) = \min_u \left[V_j(f(x,u)) + l(x,u)\right]$$

Relax the optimality within given bounds

Replace the Bellman equation by an inequality:

$$\min_u \left[V(f(x,u)) + \underline{\alpha} l(x,u) \right] \leq V(x) \leq \min_u \left[V(f(x,u)) + l(x,u) \right]$$

where $\underline{\alpha} < 1$.

From the inequalities, it follows that

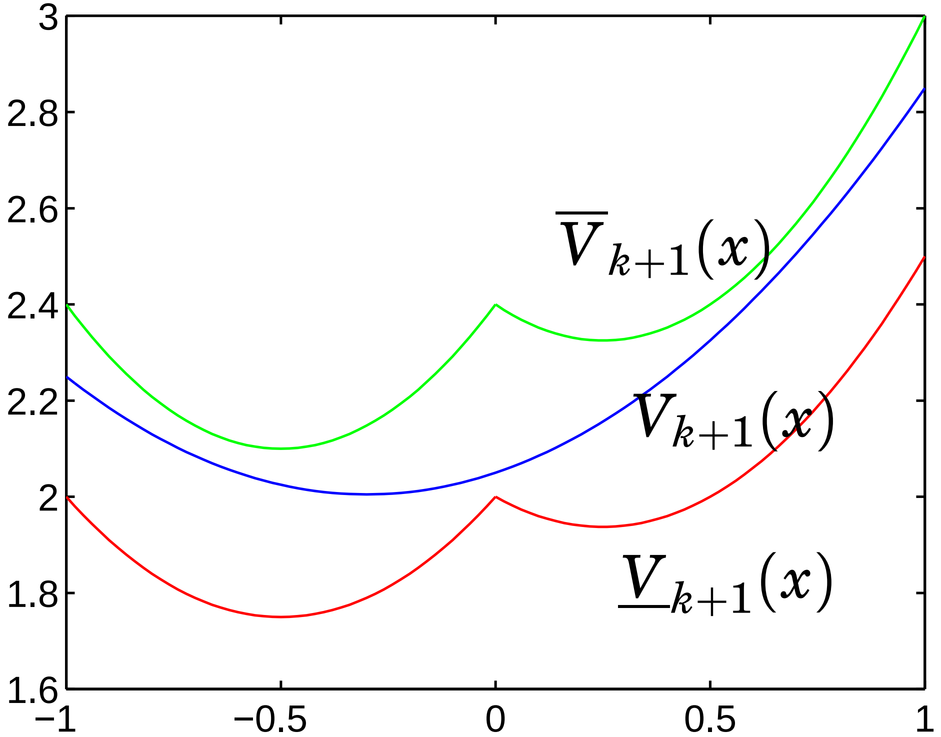
$$\underline{\alpha} V^*(x) \leq V(x) \leq V^*(x)$$

The recursive conditions become

$$\min_u \left[V_j(f(x,u)) + \underline{\alpha} l(x,u) \right] \quad \leq V_{j+1}(x) \leq \min_u \left[V_j(f(x,u)) + l(x,u) \right]$$

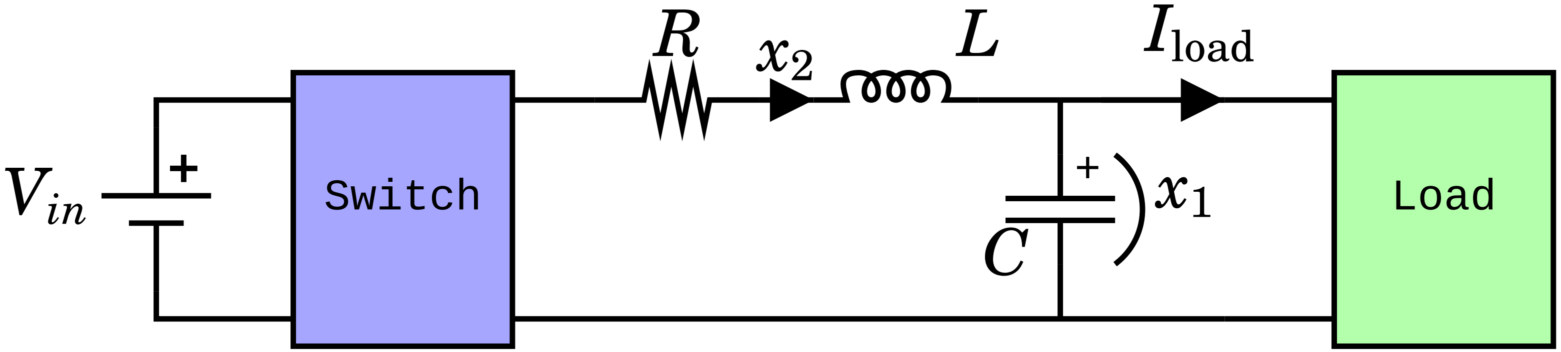
The interval for $V_{j+1}(x)$ makes it possible to work with a simplified parameterization of V_j.

Approximative dynamic programming



$$\underbrace{\min_u \left\{ V_k\left(f(x,u)\right) + \underline{\alpha} l(x,u) \right\}}_{\underline{V}_{k+1}(x)} \leq V_{k+1}(x) \leq \underbrace{\min_u \left\{ V_k\left(f(x,u)\right) + l(x,u) \right\}}_{\overline{V}_{k+1}(x)}$$

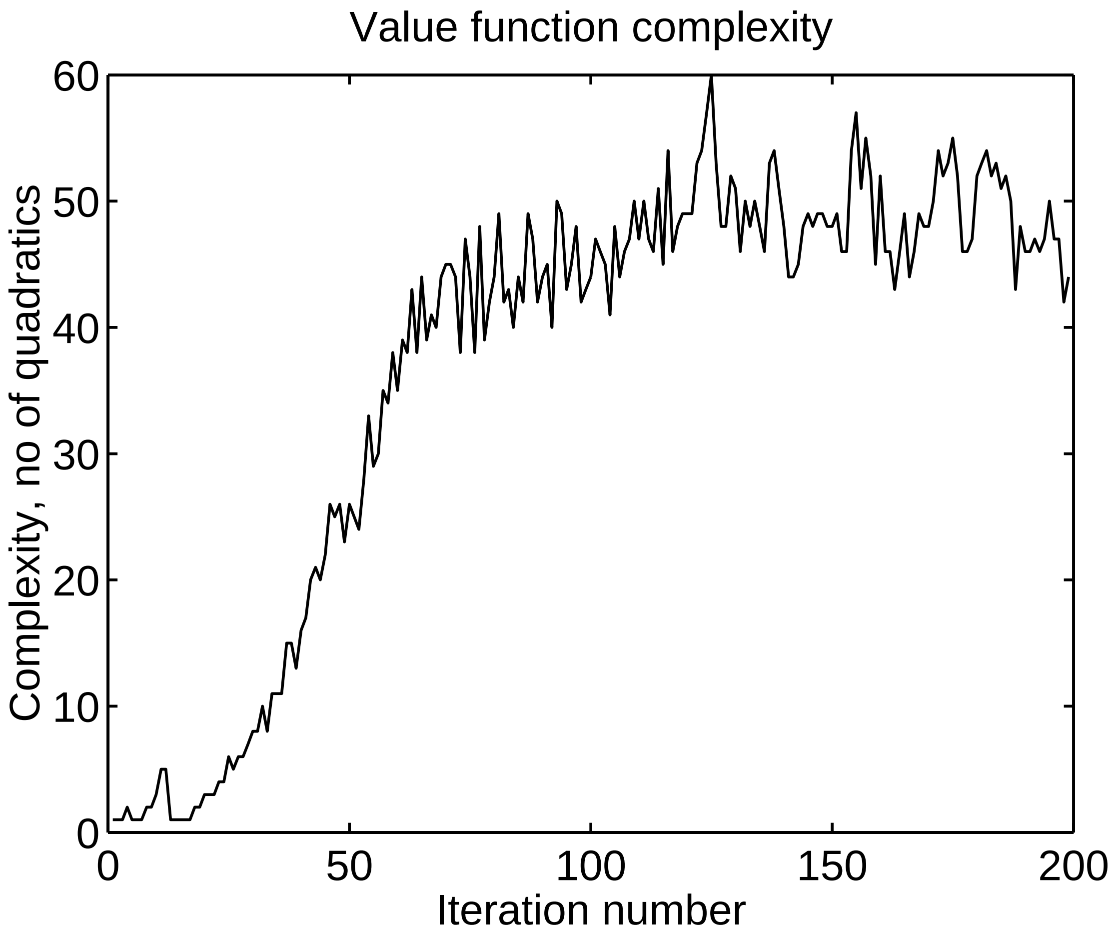
Example: Switched voltage converter



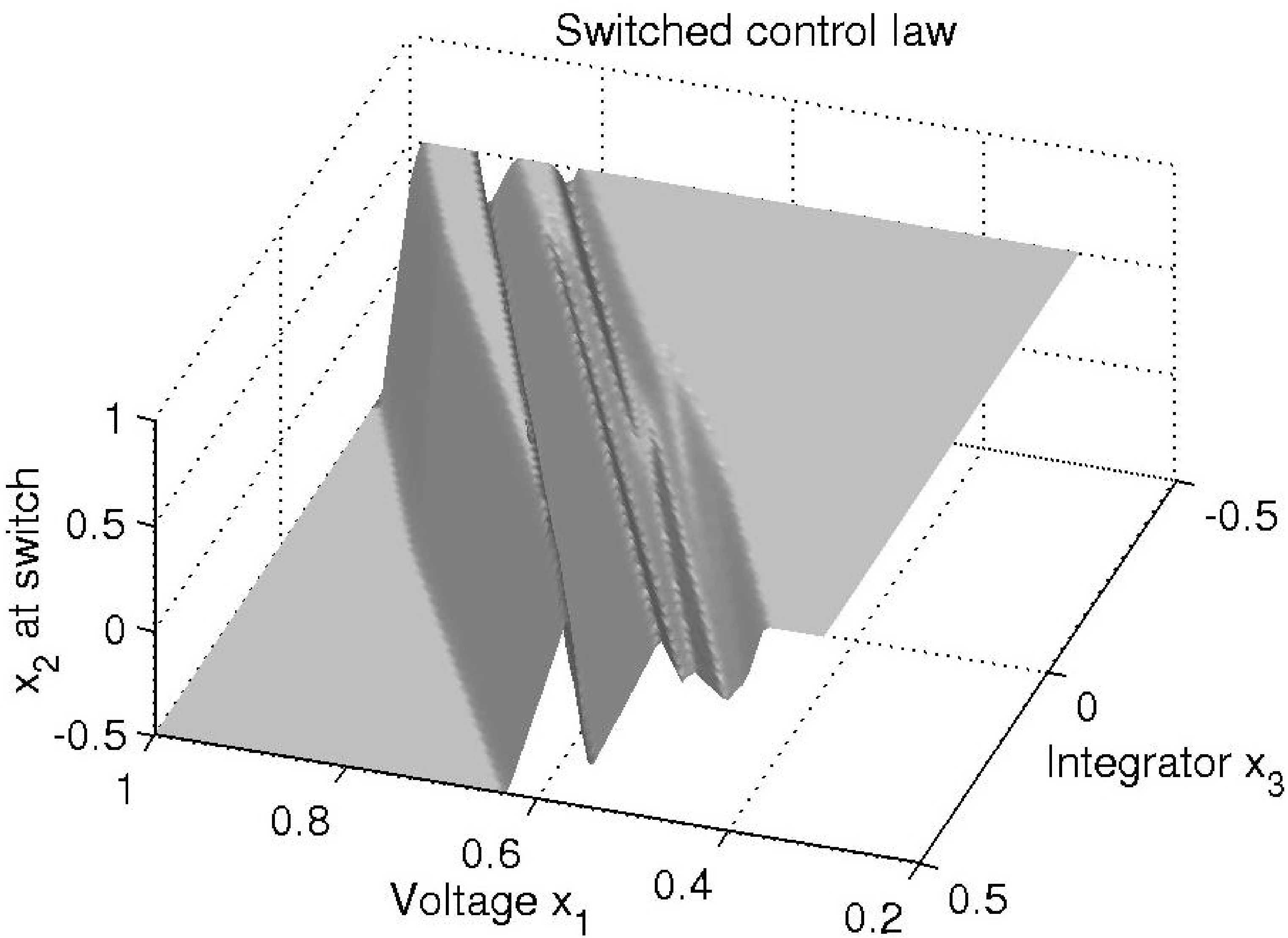
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{C}\left(x_2 - I_{\text{load}}\right) \\ -\frac{1}{L}x_1 - \frac{R}{L}x_2 + \frac{1}{L}s(t)V_{\text{in}} \\ V_{\text{ref}} - x_1 \end{bmatrix}$$

$$l(x) = q_P(x_1 - V_{\text{ref}})^2 + q_I x_3^2 + q_D(x_2 - I_{\text{load}})^2$$

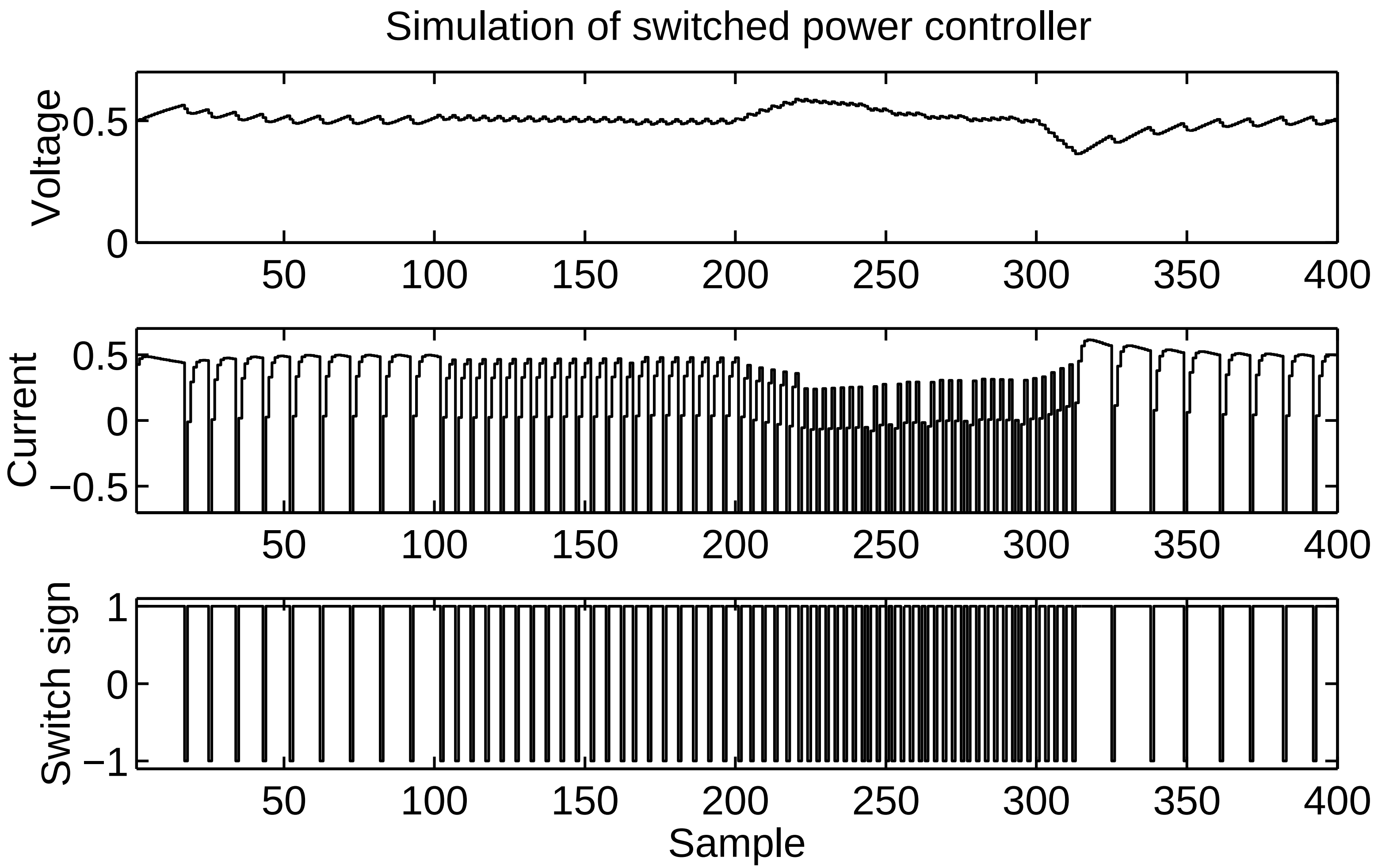
Example: Switched voltage converter



Example: Switched voltage converter



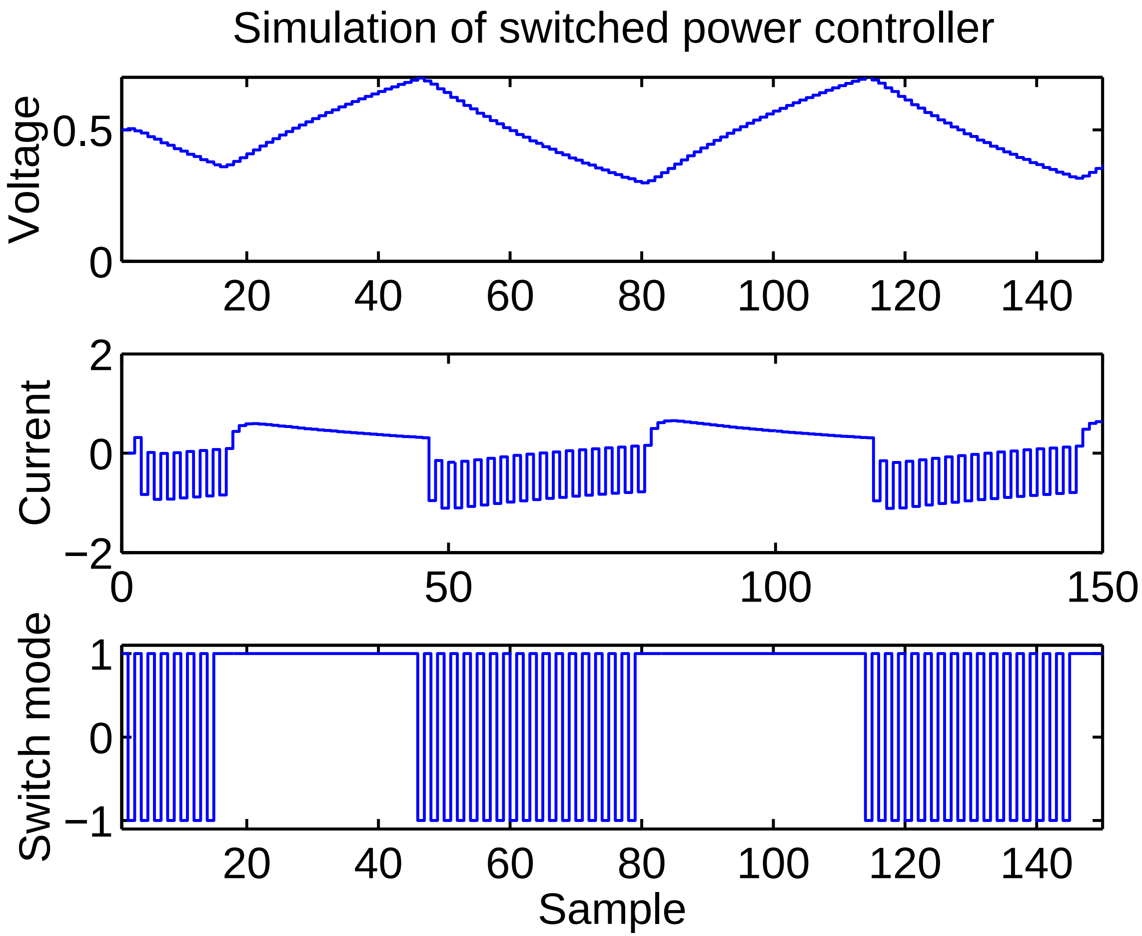
Example: Switched voltage converter



$$I_{\text{load}} = \{0.3A \qquad 0.1A \qquad -0.2A \qquad 0.3A\}$$

Example: Switched voltage converter

Frequency weights in the cost function can be used to suppress undesired harmonics. This increases state dimension, but has no significant effect on computational complexity.



Contents

- Introduction
- Approximative dynamic programming
- **If simple approximation exists, we will find one!**
- Duality and reachability
- Conclusions

Theorem

Suppose $V^*(f(x,u)) \leq \gamma l(x,u)$ uniformly and there is a polynomial U of degree n with $(1-\varepsilon)V^*(x) \leq U(x) \leq V^*(x)$. Then, with $V_0 \equiv 0$ and $\underline{\alpha} = 1 - \varepsilon(1+\gamma)^2$, the inequalities

$$\min_u \left[V_j(f(x,u)) + \underline{\alpha} l(x,u)\right] \quad \leq V_{j+1}(x) \leq \min_u \left[V_{j+1}(f(x,u)) + l(x,u)\right]$$

have solutions of degree n polynomials $V_0, V_1, V_2 \ldots$ and

$$\underline{\alpha}_k V^*(x) \leq V_k(x) \leq V^*(x)$$

where $\underline{\alpha}_k = \left[1 + \gamma(1+\gamma^{-1})^{1-k}\right]^{-1} \underline{\alpha}$.

If $\mu_k(x) = \arg\min_u \left[V_k(f(x,u)) + \underline{\alpha}_k l(x,u)\right]$, then $\underline{\alpha}_k V_{\mu_k} \leq V^*$.

If simple approximation exists, we will find one!

Asume V^{S} is "simple" and satisfies

$$\min_u \left[V^*(f(x,u)) + \underline{\alpha} l(x,u)\right] \quad \leq V^{\mathsf{S}}(x) \leq \min_u \left[V^{\mathsf{S}}(f(x,u)) + l(x,u)\right]$$

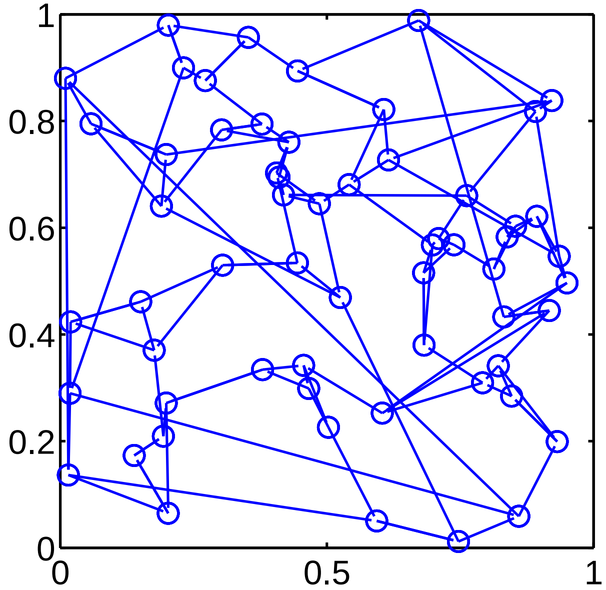
Then $\underline{\alpha} V^* < V^{\mathsf{S}} < V^*$ and the following relaxed value iteration is feasible in every step:

$$\min_u \left[V_j(f(x,u)) + \underline{\alpha} l(x,u)\right] \quad \leq V_{j+1}(x) \leq \min_u \left[V_{j+1}(f(x,u)) + l(x,u)\right]$$

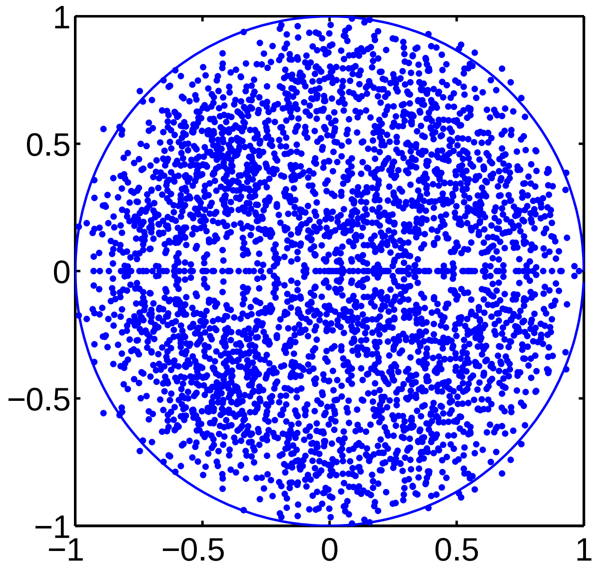
with $V_0 = 0$.

Optimal control: 60 discrete states, 30 continuous

120 edges



120×30 eigenvalues



Minimize

$$\sum_k z^T Q_{iu} z$$

Continuous dynamics: $z(k+1) = A_{i(k)u(k)} z(k)$ $z(0) = z_0 \in \mathrm{R}^{30}$

Discrete jumps: $i(k+1) = u(k)$ $i(0) = i_0$

Four steps of approximate value iteration

After four iterations we have one 30×30 matrix P^i for each node such that the following switch law is within a factor 3.81 from optimality:

$$\begin{cases} \text{Jump to node } n & \text{if } z^T[A_{in}^T P^n A_{in} + Q_{in}]z < z^T[A_{im}^T P^m A_{im} + Q_{im}]z \\ \text{Jump to node } m & \text{else} \end{cases}$$

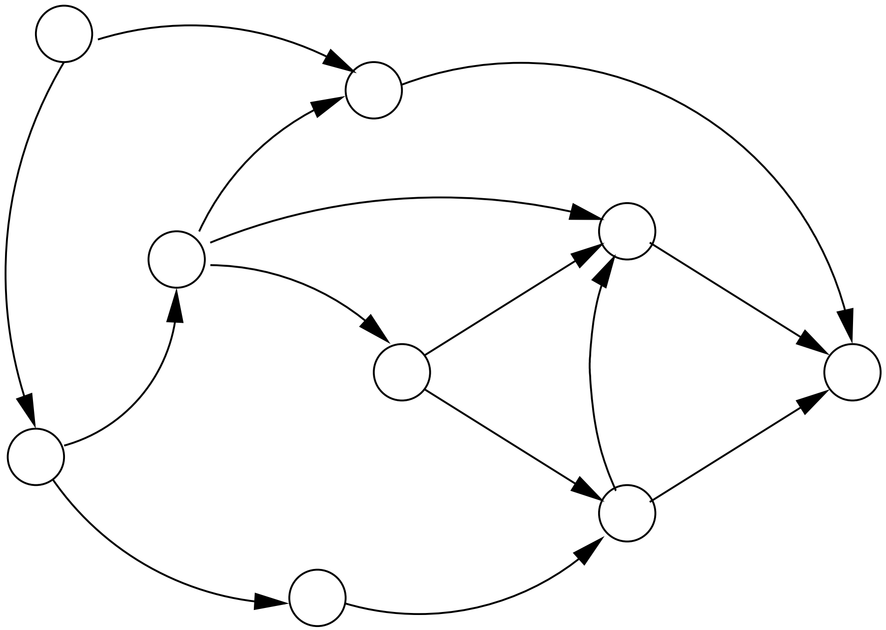
Main messages of today

- There is a rich set of optimal control problems which have simple approximative solutions
- There are algorithms that find such solutions whenever they exist

Ideas from the discrete setting are extended to continuous and hybrid setting using semi-definite and sum-of-squares programming

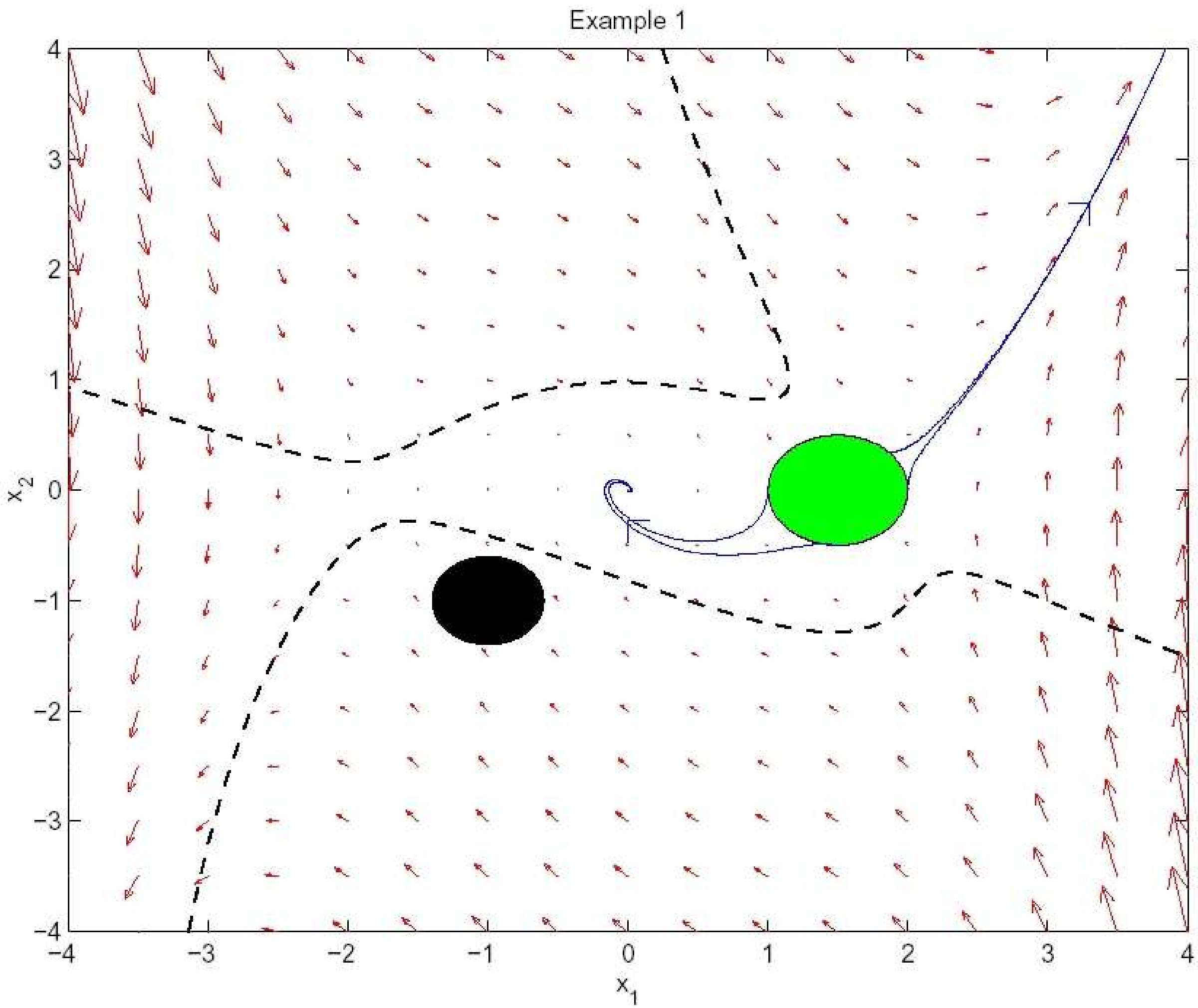
Contents

What do we learn from discrete optimization?



- Value iteration, policy iteration
- Decentralized computations
- Two dual view-points
  - Flow optimization gives an explicit control law
  - Cost optimization bounds the reachability

Example — Safety verification



$$\dot{x}_1 = x_2 \qquad \dot{x}_2 = -x_1 + x_1^3/3 - x_2$$

[Khalil,Prajna,Jadbabaie]

Duality in safety verification

Let $f \in C^1(\mathbf{R}^n, \mathbf{R}^n)$ and let $\Gamma \subset X \subset \mathbf{R}^n$ be open and bounded. Assume existence of $V \in C^1(\mathbf{R}^n)$ such that $\nabla V(x)f(x) > 0$ for $x \in X \setminus \Gamma$. Then the following two conditions are equivalent:

There exist $V \in C^1(\mathbf{R}^n)$ such that

$$V(x_0) - V(x_f) > 0 \qquad \text{and} \qquad \nabla V(x)f(x) > 0 \qquad \forall x \in X \setminus \Gamma$$

There exists no trajectory of the system $\dot{x} = f(x)$ such that

$$\begin{aligned} x(0) &= x_0 && \\ x(T) &= x_f && T > 0 \\ x(t) &\in X \setminus \Gamma && t \in [0, T] \end{aligned}$$

Long term impact of the CC project?

We are starting to learn how to combine the following two:

- Concepts (e.g. iteration methods, duality) from the literature on discrete networks and automata.
- Performance measures and computational methods (LMIs, sum-of-squares) from control in continuous state space.

The main impact is cross-fertilization of ideas.