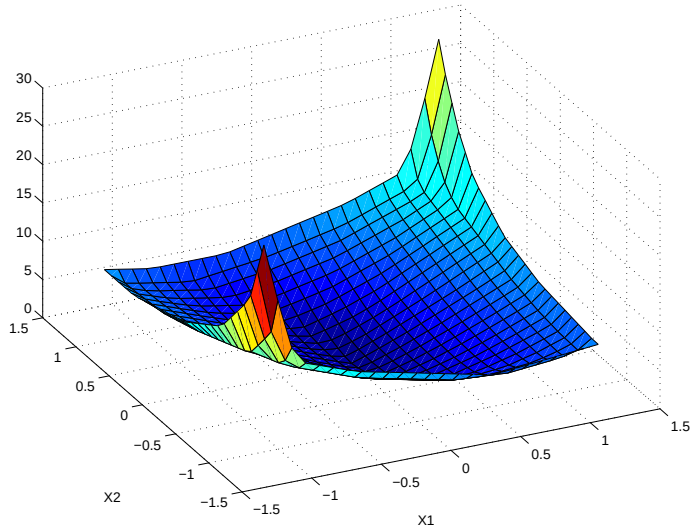
Relaxing Dynamic Programming

Bo Lincoln and Anders Rantzer
Dept. of Automatic Control
Lund Institute of Technology, Sweden
lincoln @ control.lth.se

Outline of the talk

1. Motivation: Dynamic programming, why not used more often?
2. Relaxing dynamic programming (general)
3. Application: Optimal control with piecewise linear cost
4. Application: Switched linear system
5. Conclusions



An example control problem

$$x(n+1) = Ax(n) + Bu(n)$$

$$-1 \leq u(n) \leq 1$$

Minimize cost

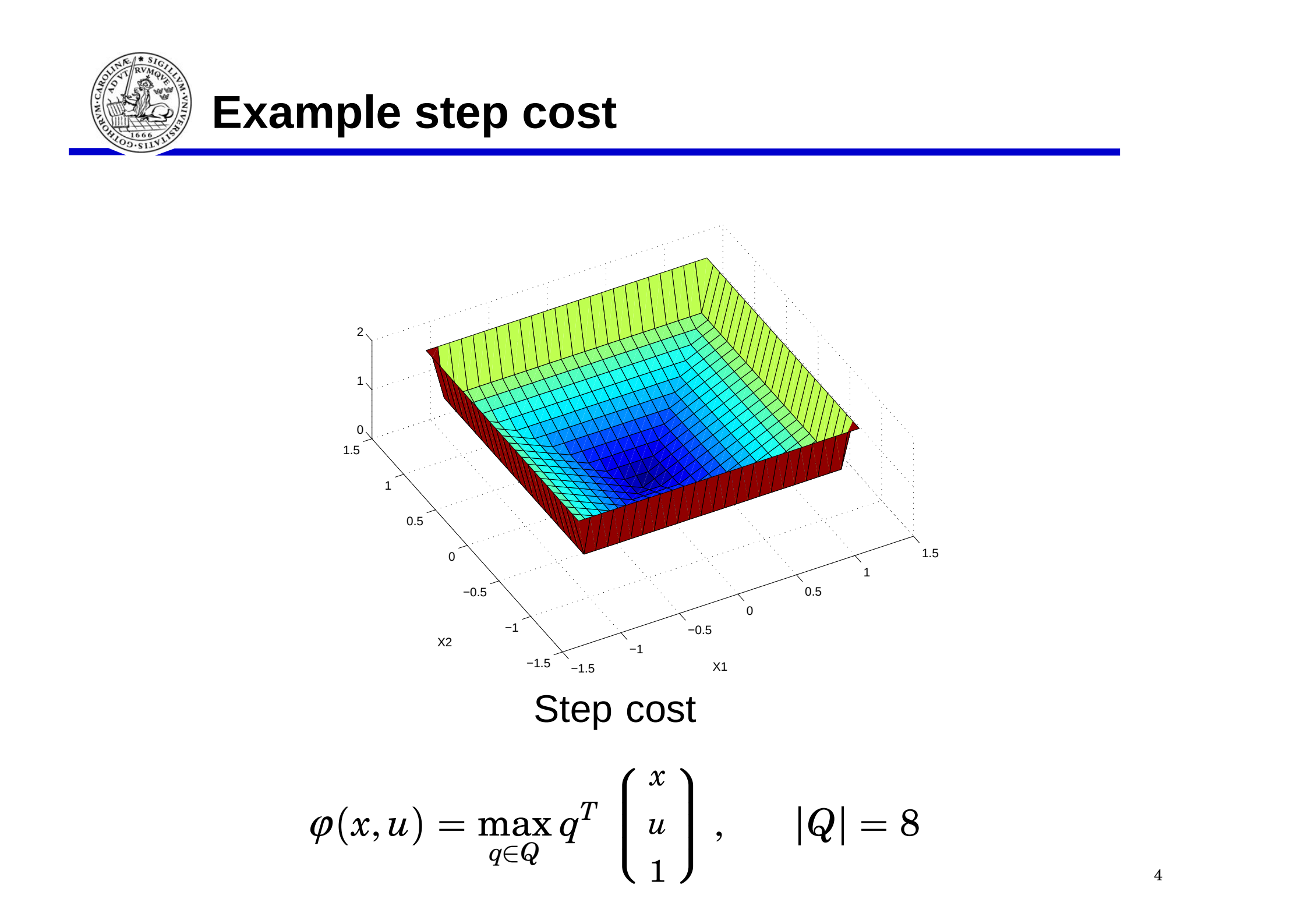
$$\sum_{i=0}^{N} \varphi(x(i), u(i))$$

Control law

$$u(n) = \mu(x(n))$$

A value function $V(x,n)$ is the expected cost starting from $x(n)$ at time n to time N.

$V^*(x,n)$ is the optimal (smallest possible) value function.



Example step cost

Step cost

$$\varphi(x,u) = \max_{q \in Q} q^T \begin{pmatrix} x \\ u \\ 1 \end{pmatrix}, \qquad |Q| = 8$$

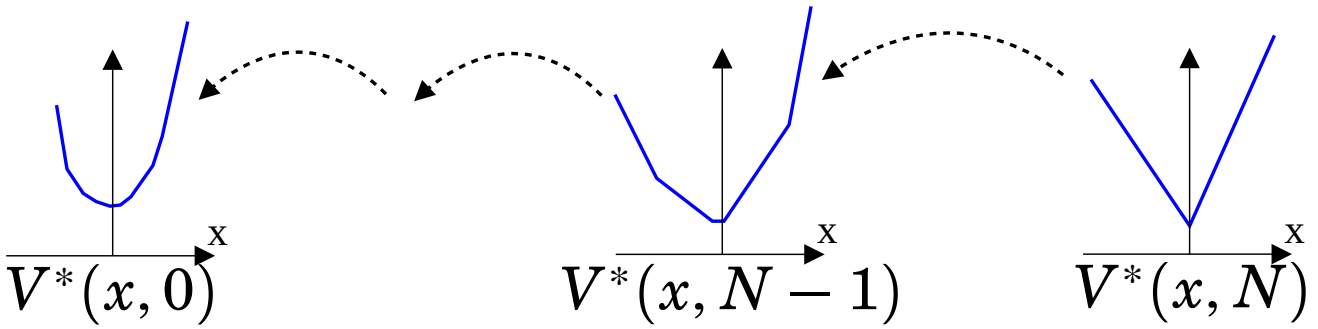
Dynamic Programming

We let the final cost

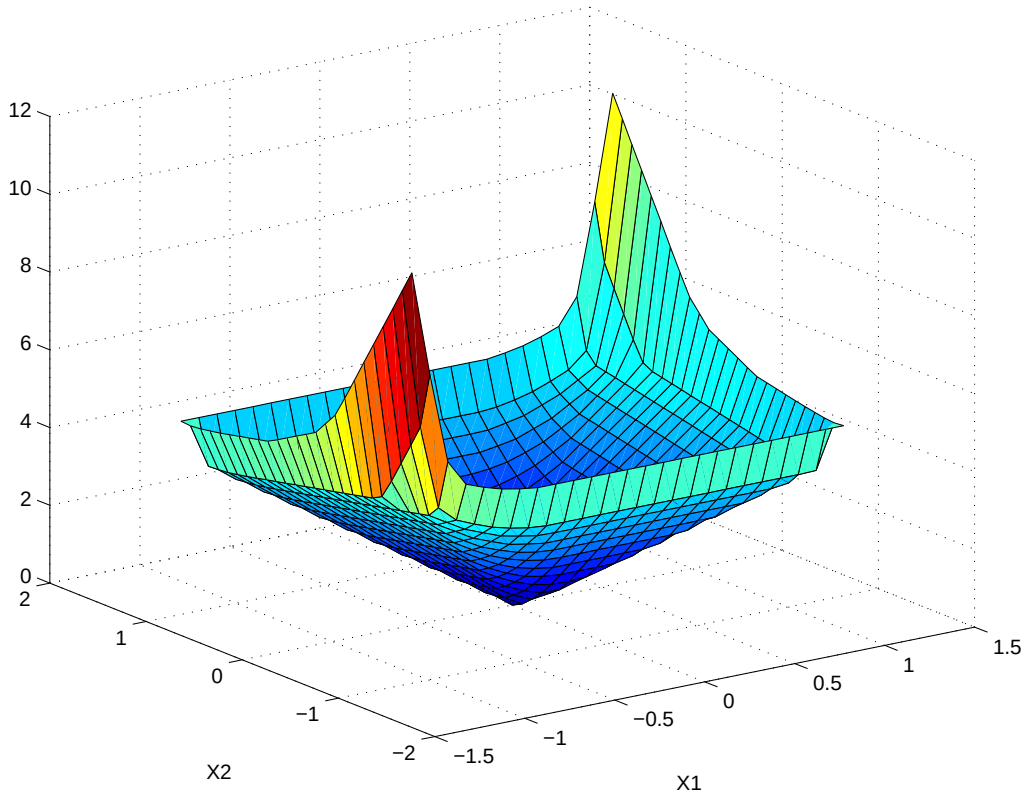
$$V^*(x,N) = \max_{p \in P} p^T \begin{pmatrix} x \\ 1 \end{pmatrix}$$

Optimal value $V^*(x,n)$ at time n from $V^*(x,n+1)$:

$$V^*(x,n) = \min_u \left\{ V^*\left(Ax + Bu, n+1\right) + \varphi(x,u) \right\}$$



Dynamic Programming



$N = 3,\ |P| = 157$

Problem: For each time step, V^* gets harder to represent.

Dynamic Programming

In general, we assume

$$x(n+1) = f(x,u), \quad u \in U$$

and a value function $V(x,n)$ that can be described by

$$V(x,n) = \underset{p \in P(n)}{\text{select}}\, p(x)$$

where $|P(n)|$ typically grows exponentially with time.

select could be e.g. max or min .

Relaxing Dynamic Programming

If $V^*(x,n)$ is hard to find, search for simpler $V^\alpha(x,n)$ s.t

$$V^*(x,n) \geq V^\alpha(x,n) \geq V^{\alpha *}(x,n)$$

Typically,

$$V^{\alpha *}(x,n) = \alpha V^*(x,n), \quad \alpha < 1 \quad (*)$$

or

$$V^{\alpha *}(x,n) = V^*(x,n) - (N-n)\alpha, \quad \alpha \geq 0 \quad (**)$$

Relaxed step cost

These correspond to the relaxed step cost φ^α

$$\varphi^\alpha(x,u) = \begin{cases} \alpha\varphi(x,u) & (*) \text{ or} \\ \varphi(x,u) - \alpha & (**) \end{cases}$$

Give bounds on V^* from V^α:

$$\frac{1}{\alpha}V^\alpha(x,n) \geq V^*(x,n) \geq V^\alpha(x,n) \qquad (*)$$

$$V^\alpha(x,n) + (N-n)\alpha \geq V^*(x,n) \geq V^\alpha(x,n) \qquad (**)$$

Calculating V^α

Assume we have

$$V^\alpha(x, n+1) = \underset{p \in P(n+1)}{\text{select}}\ p(x)$$

One step backwards:

Calculate

$$\overline{V}(x,n) = \min_u \left\{ V^\alpha\big(f(x,u), n+1\big) + \varphi(x,u) \right\}$$

$$\underline{V}(x,n) = \min_u \left\{ V^\alpha\big(f(x,u), n+1\big) + \varphi^\alpha(x,u) \right\}$$

Calculating V^{α}

Then, again

$$\overline{V}(x,n) = \underset{p \in \overline{P}(n)}{\text{select}}\, p(x)$$

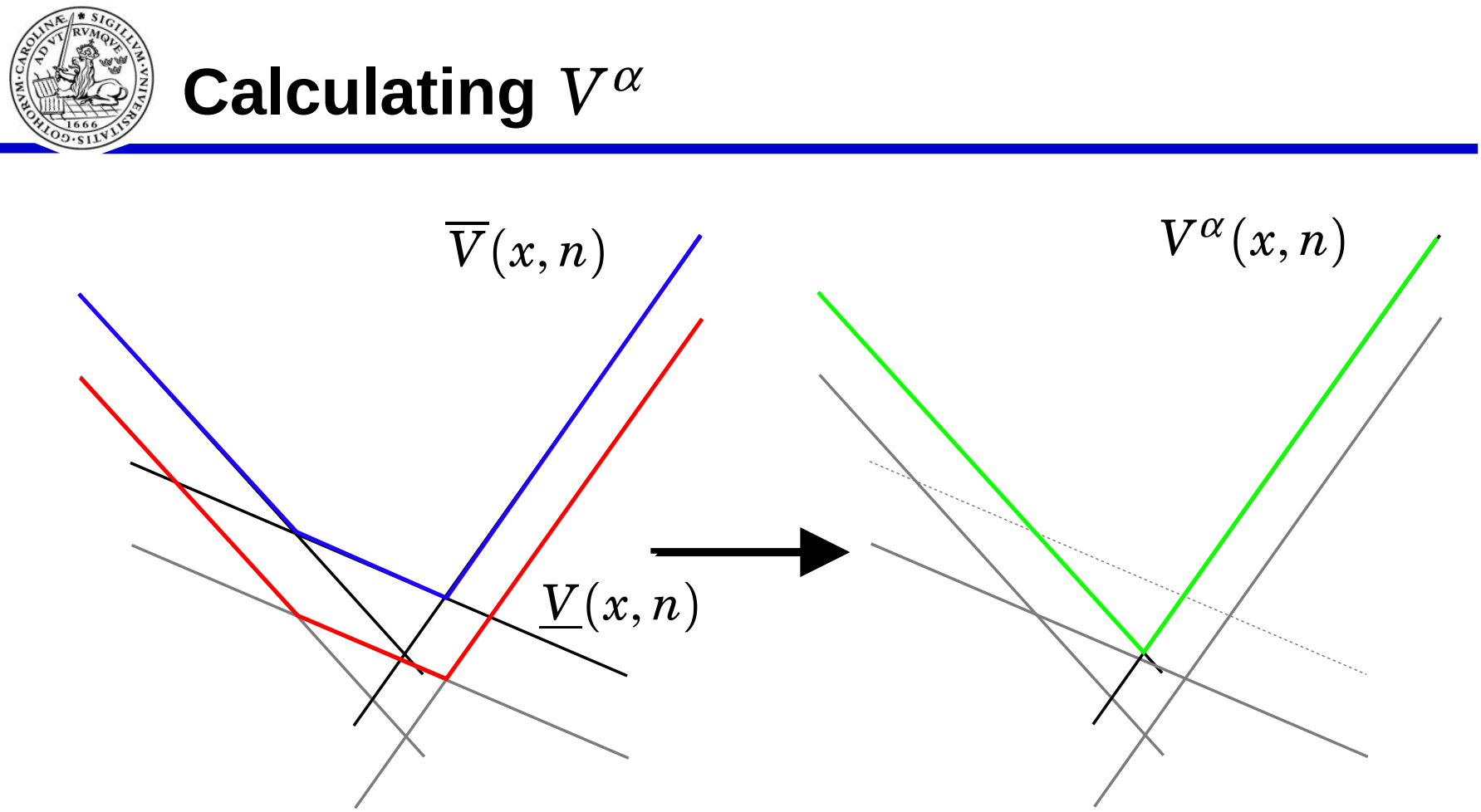
$$\underline{V}(x,n) = \underset{p \in \underline{P}(n)}{\text{select}}\, p(x)$$

Let $P(n) \subseteq \overline{P}(n)$ be the smallest set such that

$$V^{\alpha}(x,n) := \underset{p \in P(n)}{\text{select}}\, p(x) \geq \underline{V}(x,n)$$

By construction, it then holds that

$$V^{*}(x,n) \geq V^{\alpha}(x,n) \geq V^{\alpha *}(x,n)$$


Calculating V^{α}
$\overline{V}(x,n)$
$\underline{V}(x,n)$
$V^{\alpha}(x,n)$

Applications

The method can be applied to discrete-step (time) DP with a large but finite value representation.

For example:

- Optimal control of LTI system with piecewise linear cost (as seen in example).
- Partially Observable Markov Decision Process (POMDP).
- Optimal control of switched linear system with quadratic costs.
- Validation.
- Many others to be investigated.

Application: Piecewise linear cost

Let's go into the example in more detail:

$$x(n+1) = Ax(n) + Bu(n)$$

$$-1 \leq u(n) \leq 1$$

Assume

$$V^{\alpha}(x, n+1) = \max_{p \in P(n+1)} p^T \begin{pmatrix} x \\ 1 \end{pmatrix}$$

Bellman's equation:

$$\overline{V}(x, n) = \min_{-1 \leq u \leq 1} \left\{ V^{\alpha}\left(Ax + Bu, n+1\right) + \varphi(x) \right\}$$

Ignore $\varphi(x)$.

$$\overline{V}(x, n) = \min_{-1 \leq u \leq 1} \max_{p \in P(n+1)} p^T \begin{pmatrix} Ax + Bu \\ 1 \end{pmatrix}$$

Piecewise linear cost

This is equal to the linear program

$$\overline{V}(x,n) = \min_{u,g} g \quad \text{s.t.}$$
$$\pi_p^T \begin{pmatrix} Ax + Bu \\ 1 \end{pmatrix} \leq g, \quad \forall p \in P(n+1)$$

Solve using for all x using

- Brute-force method:
  - Calculate dual
  - Enumerate all feasible extreme points of the dual polyhedron. Solution = maximum of all.
- *or* use Multi-Parametric Linear Programming.

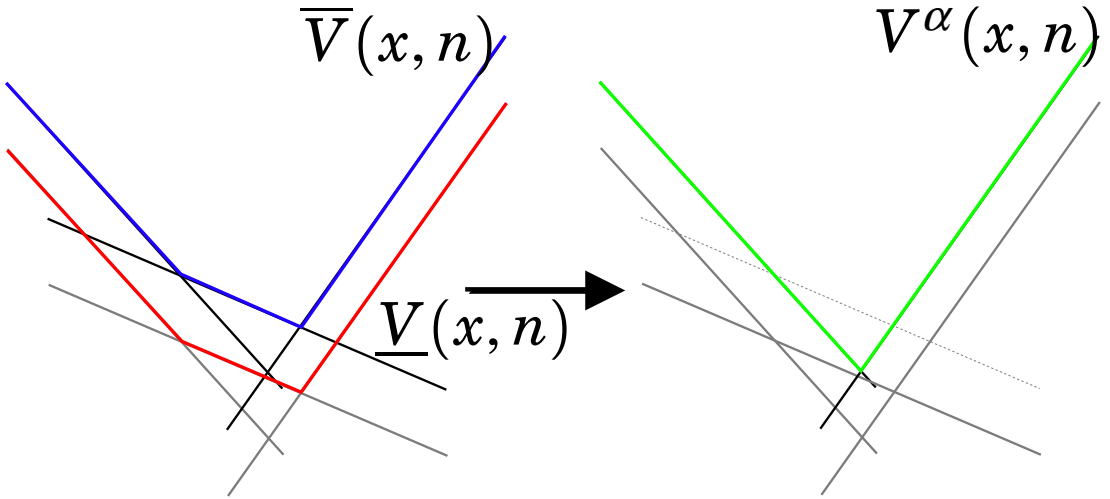
Piecewise linear cost

$$\Rightarrow \quad \overline{V}(x,n) = \max_{p \in \overline{P}(n)} p^T \begin{pmatrix} x \\ 1 \end{pmatrix}$$

Using

$$\varphi^{\alpha}(x) = \alpha\varphi(x)$$

$$\underline{V}(x,n) = \max_{p \in \underline{P}(n)} p^T \begin{pmatrix} x \\ 1 \end{pmatrix}$$



Relaxing algorithm

1. Let $P = \emptyset$.
2. Pick one $\underline{p} \in \underline{P}$.
3. If there exist x_0 s.t.

$$\underline{p}^T \begin{pmatrix} x_0 \\ 1 \end{pmatrix} \geq \max_{p \in P(n)} p^T \begin{pmatrix} x_0 \\ 1 \end{pmatrix}$$

   then

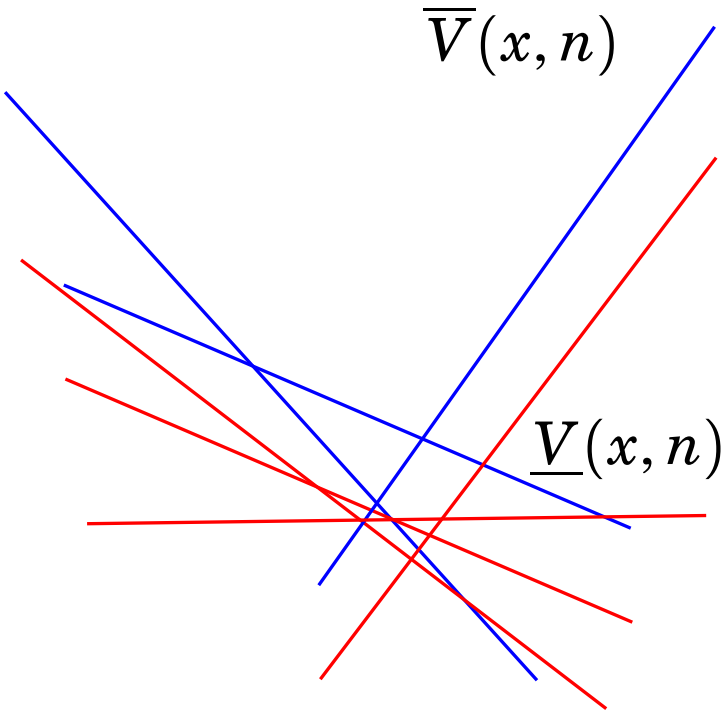
$$p_{\text{new}} = \operatorname*{argmax}_{p \in \overline{P}(n)} p^T \begin{pmatrix} x_0 \\ 1 \end{pmatrix}$$

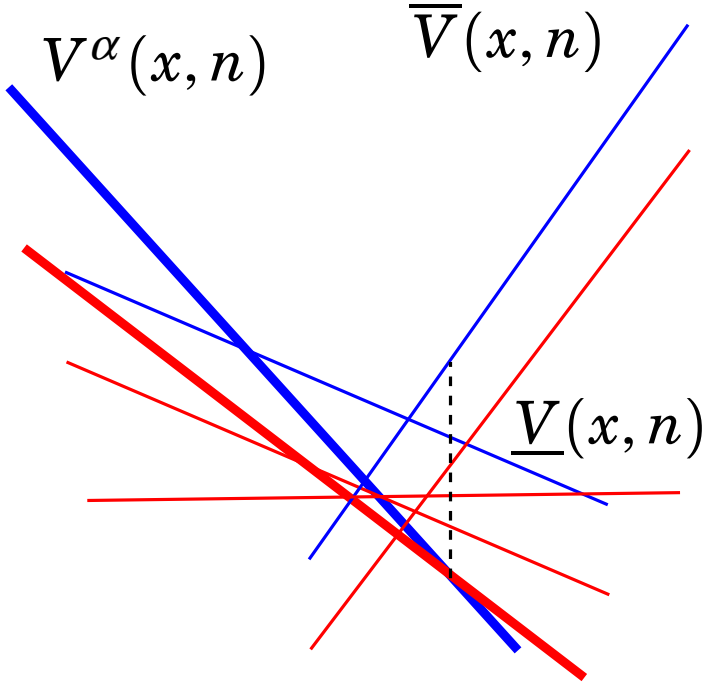
   Add p_{new} to P.

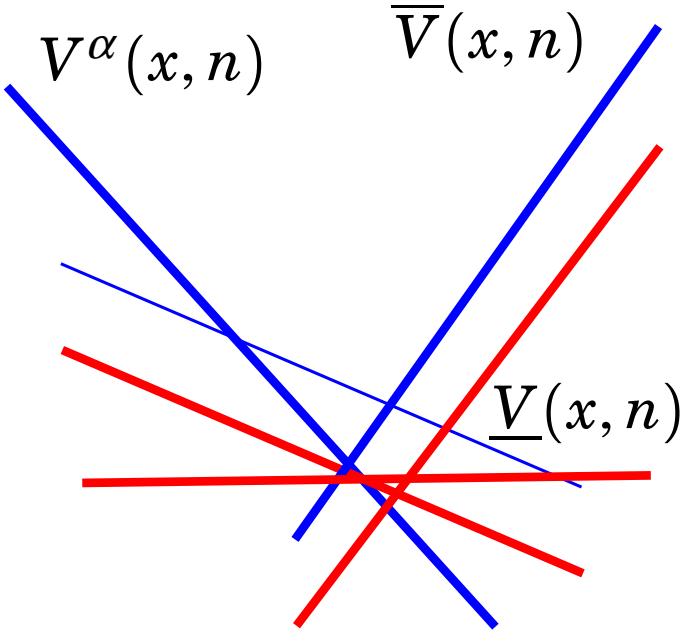
   else

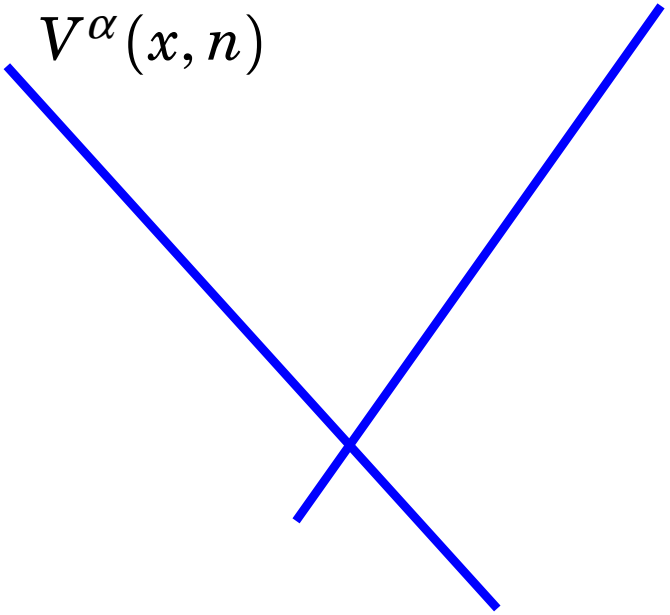
   Remove $\underline{p}$ from $\underline{P}$.

4. Repeat from 2 until $\underline{P} = \emptyset$.

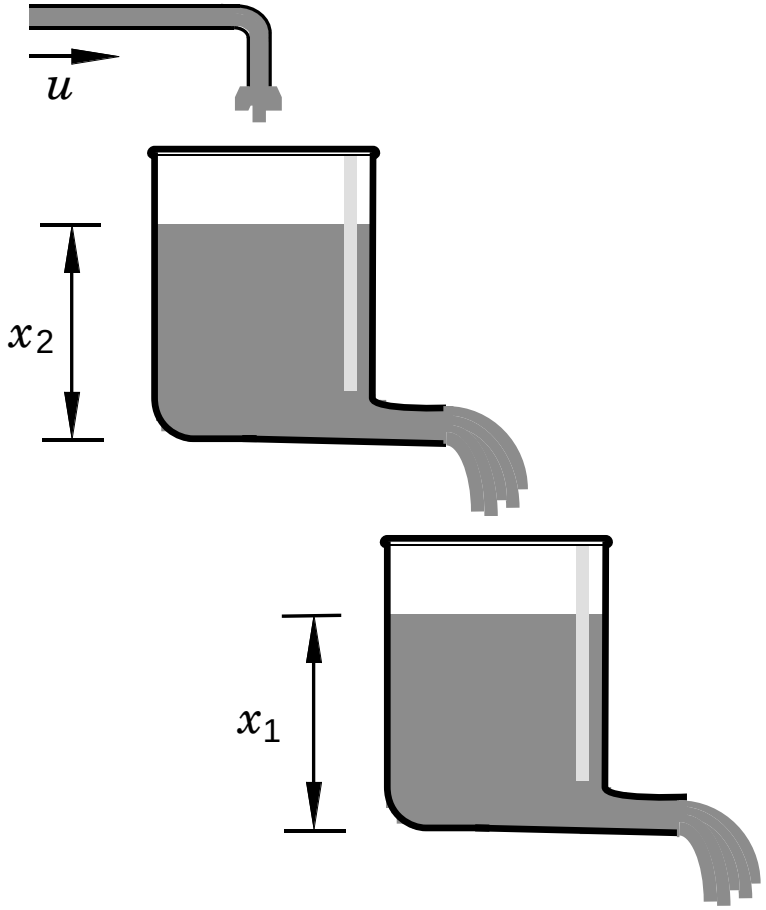

$\overline{V}(x,n)$
$\underline{V}(x,n)$


$V^{\alpha}(x,n)$
$\overline{V}(x,n)$
$\underline{V}(x,n)$


$V^{\alpha}(x,n)$
$\overline{V}(x,n)$
$\underline{V}(x,n)$


$V^{\alpha}(x,n)$

Example: Double water tank



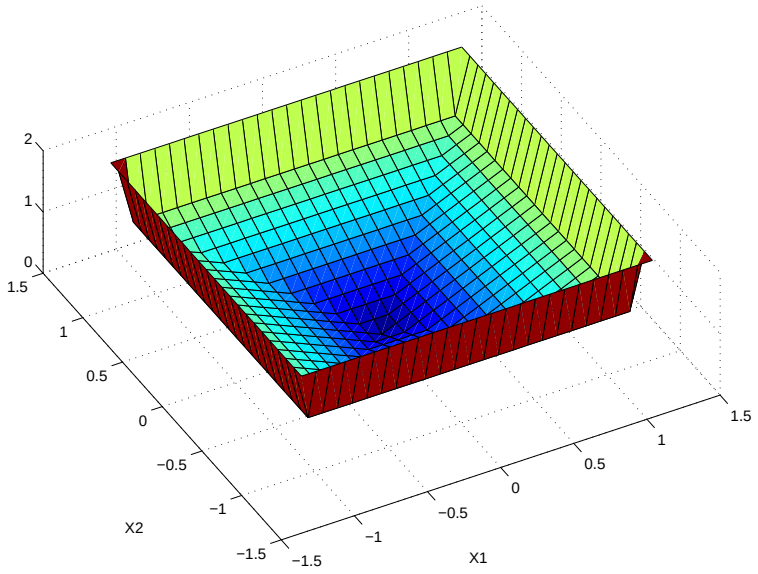
$$x(n+1) = \begin{pmatrix} 0.95 & 0.1 \\ 0 & 0.95 \end{pmatrix} x(n) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u(n), \quad -0.1 \leq u \leq 0.1$$

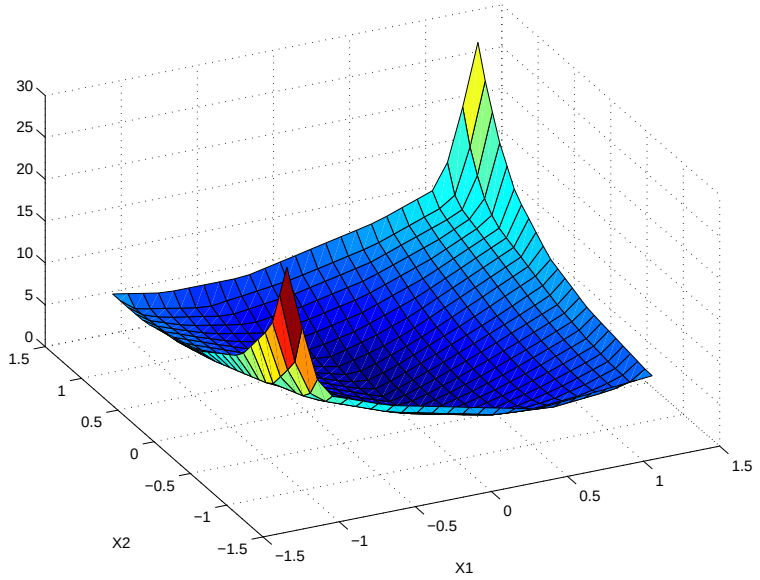
We want $-1 \leq x_1 \leq 1$ and $-1 \leq x_2 \leq 1$

Example: Double water tank

$$V^{\alpha}(x, N-10) =$$

$$\varphi(x) =$$



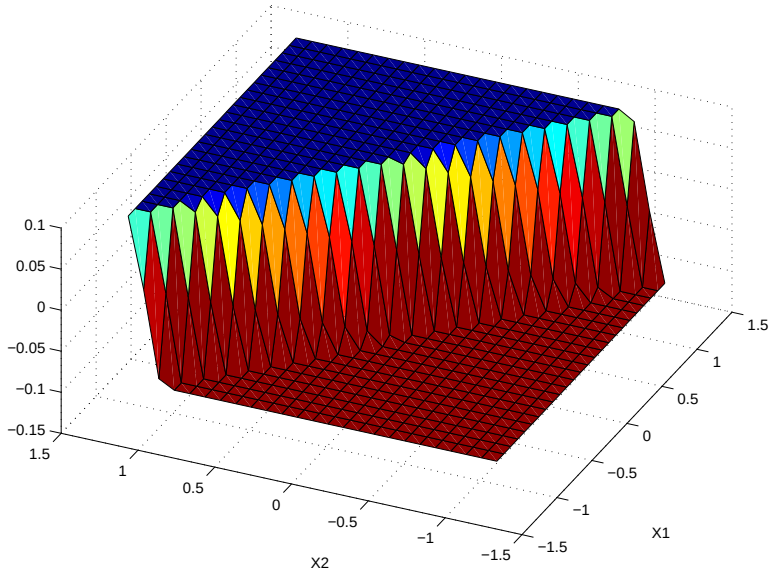


$$\varphi^{\alpha}(x) = 0.6\varphi(x)$$

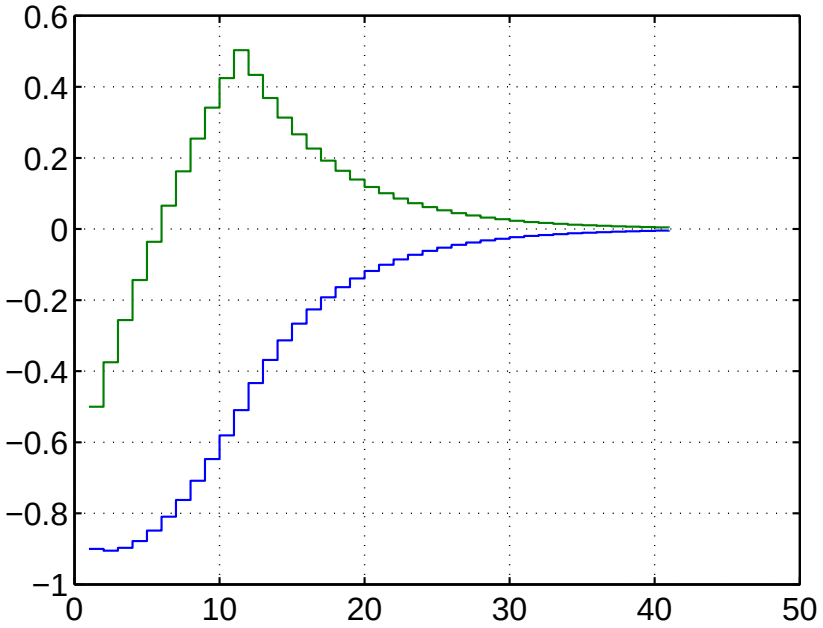
$$|P| \approx 75$$

Example: Double water tank

$u = \mu(x) =$



$x(k) =$



Application: Linear system switching

Switching system

$$x(n+1) = A_{k(n)}x(n) + B_{k(n)}u(n), \quad k(n) \in K$$

where K is index set of system matrices.

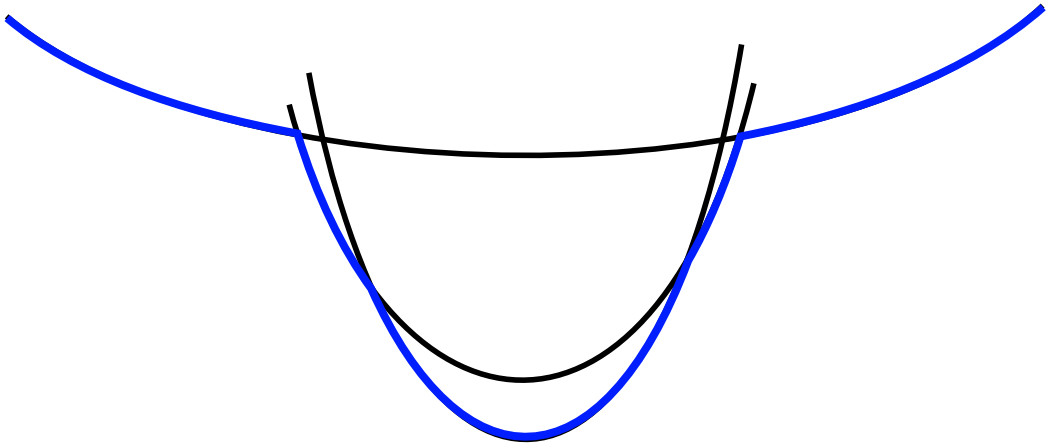
Find a feedback controller for $u(n)$ and $k(n)$ that minimizes

$$J = \sum_{n=0}^{N} \begin{pmatrix} x \\ u \end{pmatrix}^T Q \begin{pmatrix} x \\ u \end{pmatrix}$$

Linear system switching

Gives

$$V(x,n) = \min_{p \in P(n)} x^T S_p x$$



Gives one S_p per possible switching sequence $\Rightarrow$ exponential growth with time.

Using $V^{\alpha}(x,n)$ instead can reduce number drastically.

Conclusions

This talk has presented a method to simplify DP .

The method returns suboptimal solution with user-definable error bound . This allows solution-time-versus-accuracy tuning.

The method is applicable to DP problems with large but finite value function parameterization.

References:

- Bo Lincoln and Anders Rantzer: “Suboptimal dynamic programming with error bounds”, submitted to CDC’02.
- Bo Lincoln and Anders Rantzer: “Optimizing Linear System Switching”, CDC’01.