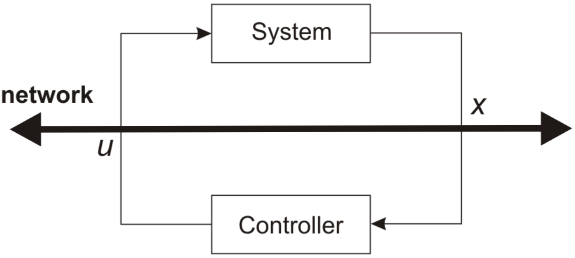
Optimistic Planning for Networked Control Systems

L. Buşoniu, R. Postoyan, J. Daafouz

COPHY, 25/06/2013

Motivation



- Networked control systems: shared network
  $\Rightarrow$ communication limited, cannot transmit all the time
- Focus: discrete-time optimal control
- **Challenge:** No solution for general nonlinear dynamics and general cost functions

Goal

- Design near-optimal control for a general class of nonlinear systems
- For general, nonquadratic cost
- Transmission intervals fixed (clock-triggered) or adapted to last measured state (self-triggered)

Setting

- System $x_{k+1} = f(x_k, u_k)$, state $x \in X \subseteq \mathbb{R}^m$, action $u \in U$
- Reward function $\rho(x_k, u_k)$
- For any x_0, find **an action sequence** $\boldsymbol{u}_\infty = (u_0, u_1, \dots)$ to maximize the **discounted return**:

$$R_{x_0}(\boldsymbol{u}_\infty) = \sum_{k=0}^{\infty} \gamma^k \rho(x_k, u_k)$$

where discount factor $\gamma \in [0, 1)$

Assumptions

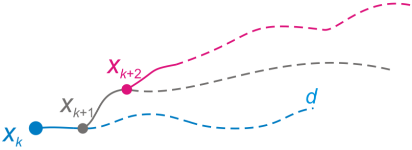
- Dynamics f known and noise-free
- Finite, discrete action space $U = \{u^1, \dots, u^K\}$
- Bounded reward function $\rho(x, u) \in [0, 1], \forall x, u$

Principle

Input: state x, computation budget n ($\sim$# simulations)
explore iteratively action sequences from x
Output: near-optimal sequence $\boldsymbol{u}^*_{d^*}$ with length d^*

(Hren & Munos, 2008)

- Adapts AI algorithm A* to infinite-horizon control
- A general type of model-predictive control
- Usually only the first action of $\boldsymbol{u}^*_{d^*}$ is sent to actuator
  $\Rightarrow$ receding-horizon control

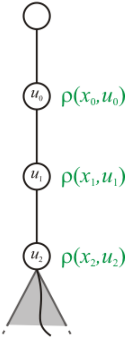


Values

- Finite sequence $\boldsymbol{u}_d$ also seen as **set** of infinite sequences $(u_0, \dots, u_{d-1}, \star, \star, \dots)$

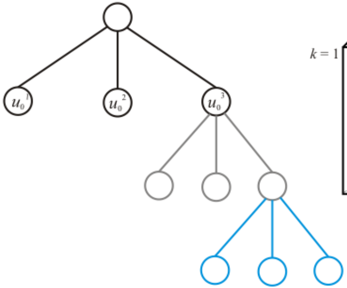
- $\nu(\boldsymbol{u}_d) = \sum_{k=0}^{d-1} \gamma^k \rho(x_k, u_k)$
  lower bound on returns of $\boldsymbol{u}_\infty \in \boldsymbol{u}_d$
- $b(\boldsymbol{u}_d) = \nu(\boldsymbol{u}_d) + \frac{\gamma^d}{1-\gamma}$
  upper bound on returns of $\boldsymbol{u}_\infty \in \boldsymbol{u}_d$

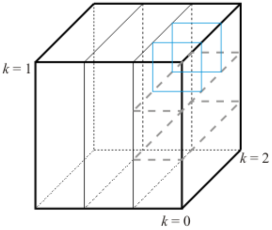
- $v(\boldsymbol{u}_d) = \sup_{\boldsymbol{u}_\infty \in \boldsymbol{u}_d} R(\boldsymbol{u}_\infty)$
  value of applying $\boldsymbol{u}_d$ and then acting optimally



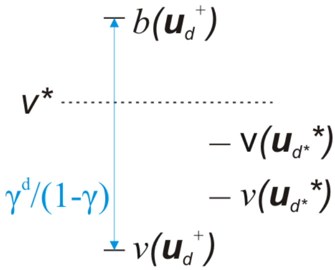
Algorithm (Hren & Munos, 2008)

Initialize empty sequence $\boldsymbol{u}_0$ (= all infinite sequences)
loop n times
  Select **optimistic** leaf sequence $\boldsymbol{u}_d^\dagger$, maximizing b
  Expand $\boldsymbol{u}_d^\dagger$: initialize all values for the $d+1$-th action
end loop
return greedy $\boldsymbol{u}^*_{d^*}$ maximizing ν





Near-optimality



For any expanded $\boldsymbol{u}_d^{\dagger}$:

- $b(\boldsymbol{u}_d^{\dagger}) \geq v^*$, otherwise it wouldn't have been selected
- $\nu(\boldsymbol{u}_d^{\dagger}) \leq \nu(\boldsymbol{u}_{d^*}^{*})$ since $\boldsymbol{u}_{d^*}^{*}$ maximizes ν
- $v(\boldsymbol{u}_{d^*}^{*}) \geq \nu(\boldsymbol{u}_{d^*}^{*})$ by definition
- So $v^* - v(\boldsymbol{u}_{d^*}^{*}) \leq \frac{\gamma^d}{1-\gamma}$

Moreover, deepest expanded $d = d^*$

Relation to budget n

- Algorithm only expands in near-optimal subtree:

$$\mathcal{T}^* = \left\{ \boldsymbol{u}_d \;\middle|\; v^* - v(\boldsymbol{u}_d) \leq \frac{\gamma^d}{1-\gamma} \right\}$$

- Define κ = asymptotic branching factor of $\mathcal{T}^*$: **complexity measure** of optimal control problem
- So to reach depth d, $n = O(d^\kappa)$ expansions required

$$\Rightarrow \quad d^* = \Omega\left(\frac{\log n}{\log \kappa}\right), \quad v^* - v(\boldsymbol{u}^*_{d^*}) = O\left(n^{-\frac{\log 1/\gamma}{\log \kappa}}\right)$$

Summary of OP guarantees

Recall everything in fact depends on x (v, ν, b, d^*, κ)

1. OP returns a long sequence $\boldsymbol{u}_{d(x)}$, $d(x) = \Omega(\frac{\log n}{\log \kappa(x)})$
2. This sequence is near-optimal:

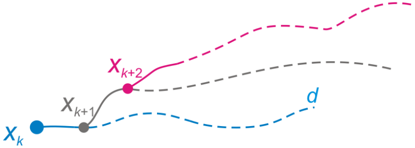
$$v_x^* - v_x(\boldsymbol{u}_{d(x)}) \leq \frac{\gamma^{d(x)}}{1-\gamma} = O(n^{-\frac{\log 1/\gamma}{\log \kappa(x)}})$$

- General optimal control, paid by exponential computation $n = O(\kappa(x)^{d(x)})$
- But $\kappa(x)$ can be small in interesting problems!

(Hren & Munos, 2008)

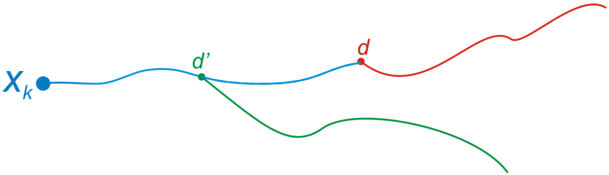
Idea

- Usually only first action of each sequence is sent to actuator



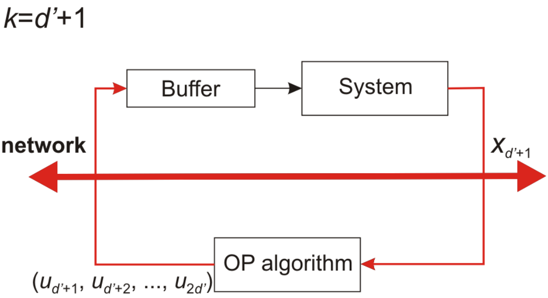
- But recall: OP returns **long sequences**!

$\Rightarrow$ Instead of first action, send a longer subsequence



NCS Architecture

- Sporadically closes loop and transmits (sub)sequences, each applied in open-loop
- Requires a **buffer** to store the (sub)sequences



Strategy 1

Self-triggered OP (STOP)

Input: computation budget n

loop

measure state x_k

apply OP at x_k with budget n to obtain $\boldsymbol{u}_{d(x_k)}$

transmit initial subsequence $\boldsymbol{u}_{d'(x_k)}$

$k \leftarrow k + d'(x_k)$, wait $d'(x_k)$ steps

end loop

Adaptive, long transmission intervals

$$d'(x_k) \sim d(x_k) = \Omega\left(\frac{\log n}{\log \kappa(x_k)}\right)$$

Strategy 2

OP can also be called with a desired sequence length d

Clock-triggered OP (COP)

Input: transmission interval d

loop

measure state x_k

apply OP at x_k with desired length d to obtain $\boldsymbol{u}_d$

transmit initial subsequence $\boldsymbol{u}_{d'}$

$k \leftarrow k + d'$, wait d' steps

end loop

Bounded computation to reach d, $n(x_k) = O(\kappa(x_k)^d)$

Near-optimality guarantee

Theorem 1

The overall sequence obtained in closed loop is **near-optimal**:

- $\frac{\gamma^{d(x_0)}}{1-\gamma}$-optimal in STOP;
- $\frac{\gamma^{d}}{1-\gamma}$-optimal in COP.

Near-optimality at first step dominates;
no change in bound with shorter or longer sequences (d')

Shorter sequences

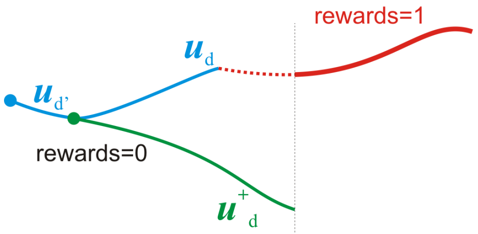
Depending on problem, shorter sequences better or worse

Theorem 2

Bounded loss from applying subsequence $\boldsymbol{u}_{d'}$ followed by new sequence $\boldsymbol{u}_d^+$:

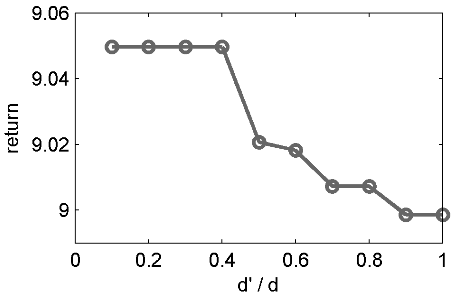
$$v_x((\boldsymbol{u}_{d'}, \boldsymbol{u}_d^+)) \geq v_x(\boldsymbol{u}_d) - \frac{\gamma^{d+d'}}{1-\gamma}$$

which is **tight in the worst case**.



Shorter sequences (cont'd)

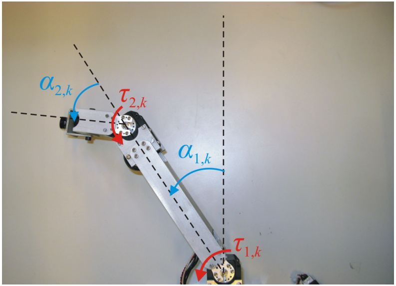
In practice shorter sequences often better, e.g. DC motor



Some related work

- Eqtami et al. (CDC 2011): nonlinear MPC, quadratic costs, applies subsequences: we handle nonquadratic cost, characterize OP solver
- Linear MPC with subsequences: Henriksson et al. (IFAC ACCP 2012), Barradas Berglind et al. (IFAC NLMPCC 2012)
- Antunes et al. (CDC 2012): exploits dynamic programming (like OP); linear, discounted quadratic costs
- Chaillet & Bicchi (CDC 2008): applies subsequences to deal with delays

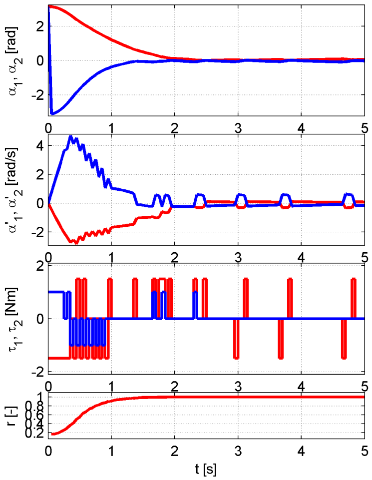
Example: nonlinear robot arm

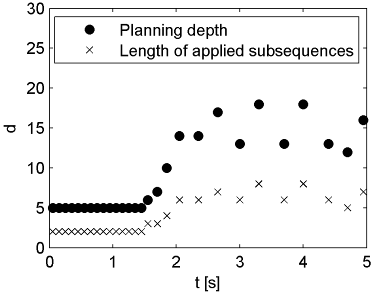


- State $[\alpha_1, \alpha_2, \dot{\alpha}_1, \dot{\alpha}_2]^\top$
- Action $[\tau_1, \tau_2]^\top$, $U = \{-1.5, 0, 1.5\} \times \{-1, 0, 1\}$
- $T_\mathrm{s} = 0.05\,\mathrm{s}$
- Rewards to reach zero state: $-x_k{}^\top Q x_k - u_k{}^\top R u_k$, $\gamma = 0.95$

Robot arm results

STOP from $x_0 = [\pi, \pi, 0, 0]^\top$, $n = 1000$, $d'(x) = \lceil 0.4 \cdot d(x) \rceil$





Less than 1.22% loss in return w.r.t. transmitting at each step

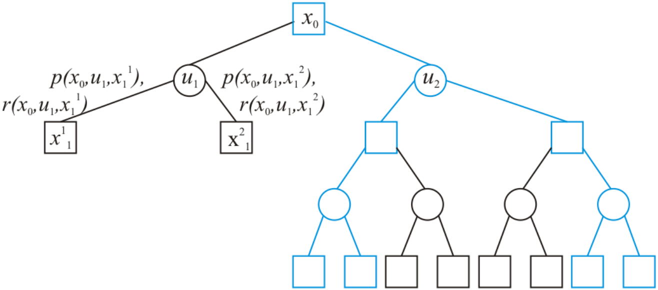
Summary & Ongoing work

- Near-optimal control in nonlinear NCS
- Theoretical guarantees and promising simulations

Ongoing work

- Classes of stochastic dynamics (uncertainty)
- Deal with computational cost: parallelization, new strategies

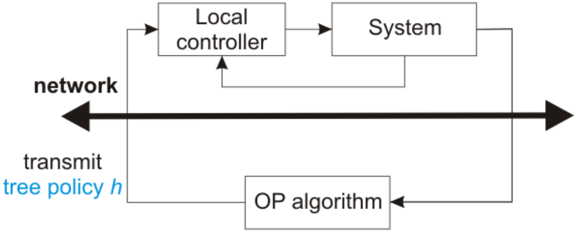
Stochastic case



Sequence replaced by closed-loop policy *h* (subtree), characterized by:

1. Near-optimality δ
2. Effective depth *d* s.t. $\gamma^d = \mathrm{E}\left\{\gamma^{d(x)}\right\} \leq \delta(1-\gamma)$

Stochastic case: Architecture



- Requires local, computationally cheap state feedback controller
- ... connected via network to the OP controller

Stochastic case: Strategies

- Set budget n
  $\Rightarrow$ small distance from optimal δ, large effective depth d
- Set near-optimality δ (corresponding to d)
  $\Rightarrow$ bounded computation