Robustness of Performance & Stability for Multistep & Updated Multistep MPC

Lars Grüne and Vryan Gil Palma

Chair of Applied Mathematics, University of Bayreuth

Rome, 12 November 2014

Contents

- Setting and preliminaries
- 3 algorithms: standard, m-step and updated m-step MPC
- Nominal stability and performance of the m-step MPC
- Perturbed Systems
- The m-step and the updated m-step MPC in the perturbed setting

Preliminary Setup

Consider the nonlinear discrete time control system

$$x(k+1) = f(x(k), u(k))$$

- state $x \in \mathbb{X} \subseteq X$ and control value $u \in \mathbb{U} \subseteq U$

Preliminary Setup

Consider the nonlinear discrete time control system

$$x(k+1) = f(x(k), u(k))$$

- state $x \in \mathbb{X} \subseteq X$ and control value $u \in \mathbb{U} \subseteq U$
- $x_u(\cdot, x_0)$ – trajectory when initial state x_0 is driven by control sequence $u(\cdot)$

Preliminary Setup

Consider the nonlinear discrete time control system

$$x(k+1) = f(x(k), u(k))$$

- state $x \in \mathbb{X} \subseteq X$ and control value $u \in \mathbb{U} \subseteq U$
- $x_u(\cdot, x_0)$ – trajectory when initial state x_0 is driven by control sequence $u(\cdot)$
- a time-dependent feedback law $\mu : \mathbb{X} \times \mathbb{N}_0 \to \mathbb{U}$ yields the **feedback controlled system**

$$x(k+1) = f(x(k), \mu(x(\tilde{k}), k))$$

where $\tilde{k} = \tilde{k}(k) \leq k$

Preliminary Setup

- Problem: find μ that 'solves' the **infinite horizon OCP**

$$\begin{aligned} \text{min} \quad & J_\infty(x_0, u(\cdot)) := \sum_{k=0}^{\infty} \ell\left(x_u(k, x_0), u(k)\right) \\ \text{w.r.t.} \quad & u(\cdot) \in \mathbb{U}^\infty(x_0) \end{aligned}$$

Preliminary Setup

- Problem: find μ that 'solves' the **infinite horizon OCP**

$$\begin{aligned} \text{min} \quad & J_\infty\left(x_0, u(\cdot)\right) := \sum_{k=0}^{\infty} \ell\left(x_u(k, x_0), u(k)\right) \\ \text{w.r.t.} \quad & u(\cdot) \in \mathbb{U}^\infty(x_0) \end{aligned}$$

- **optimal value function:**

$$V_\infty(x_0) := \inf_{u(\cdot) \in \mathbb{U}^\infty(x_0)} J_\infty\left(x_0, u(\cdot)\right)$$

- **closed-loop performance of μ:**

$$J_\infty^{\mathsf{cl}}\left(x_0, \mu\right) := \sum_{k=0}^{\infty} \ell\left(x_\mu(k, x_0), \mu(x_\mu(\tilde{k}, x_0), k)\right)$$

Preliminary Setup

- Problem: find μ that 'solves' the **infinite horizon OCP**

$$\begin{aligned} \text{min} \quad & J_\infty(x_0, u(\cdot)) := \sum_{k=0}^{\infty} \ell(x_u(k, x_0), u(k)) \\ \text{w.r.t.} \quad & u(\cdot) \in \mathbb{U}^\infty(x_0) \end{aligned}$$

- **optimal value function:**

$$V_\infty(x_0) := \inf_{u(\cdot) \in \mathbb{U}^\infty(x_0)} J_\infty(x_0, u(\cdot))$$

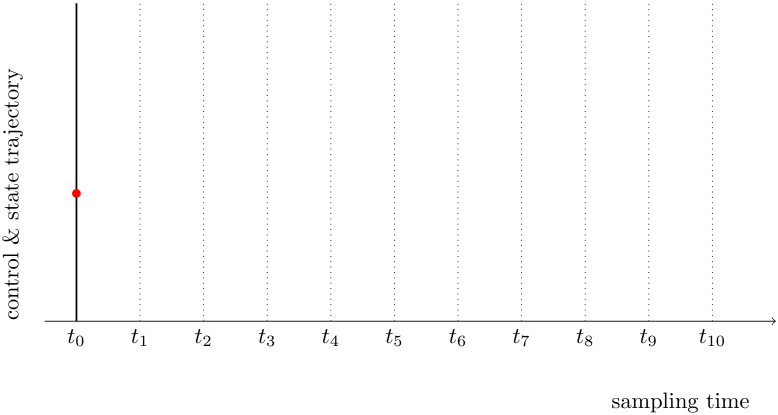
- **closed-loop performance of μ:**

$$J_\infty^{\mathsf{cl}}(x_0, \mu) := \sum_{k=0}^{\infty} \ell\Big(x_\mu(k, x_0), \mu(x_\mu(\tilde{k}, x_0), k)\Big)$$

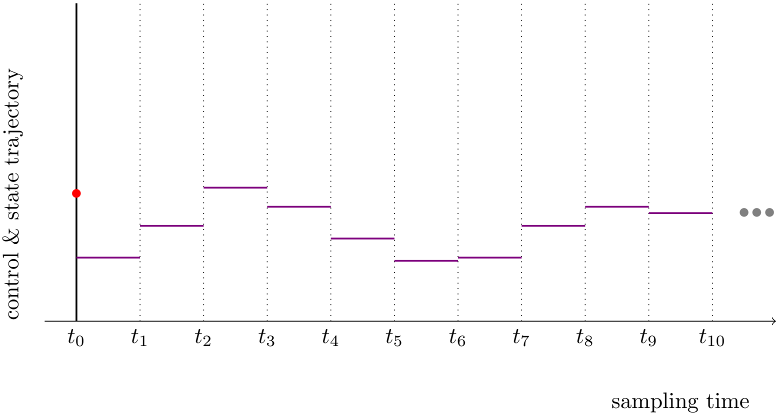
- find μ such that

$$V_\infty(x_0) = J_\infty^{\mathsf{cl}}(x_0, \mu)$$

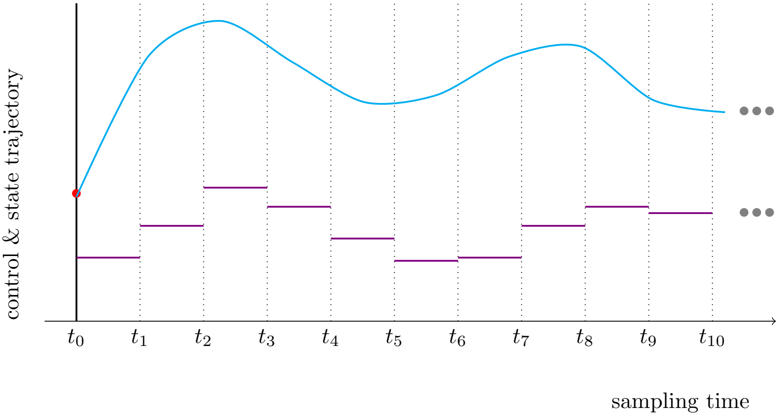
Infinite Horizon Problem



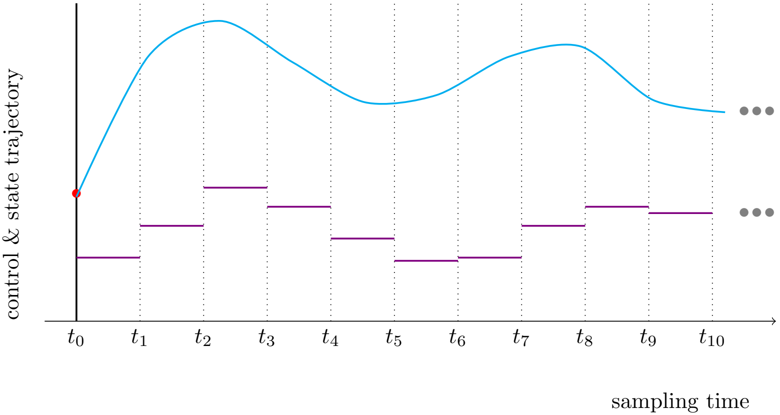
Infinite Horizon Problem



Infinite Horizon Problem



Infinite Horizon Problem



Issue: solving infinite horizon OCP is difficult.

Finite Horizon Problem

- **MPC:** receding horizon strategy to obtain the feedback law

Finite Horizon Problem

- **MPC:** receding horizon strategy to obtain the feedback law
- the **finite horizon OCP** counterpart:

$$\mathcal{P}_N(x_0) \qquad \begin{array}{ll} \text{min} & J_N\left(x_0, u(\cdot)\right) := \sum_{k=0}^{N-1} \ell\left(x_u(k, x_0), u(k)\right) \\ \text{w.r.t.} & u(\cdot) \in \mathbb{U}^N(x_0) \end{array}$$

Finite Horizon Problem

- **MPC:** receding horizon strategy to obtain the feedback law
- the **finite horizon OCP** counterpart:

$$\mathcal{P}_N(x_0) \qquad \begin{array}{ll} \text{min} & J_N\left(x_0, u(\cdot)\right) := \sum_{k=0}^{N-1} \ell\left(x_u(k, x_0), u(k)\right) \\ \text{w.r.t.} & u(\cdot) \in \mathbb{U}^N(x_0) \end{array}$$

- **optimal value function:**

$$V_N(x_0) := \inf_{u(\cdot) \in \mathbb{U}^N(x_0)} J_N\left(x_0, u(\cdot)\right)$$

- **optimal control sequence:** $u^*(\cdot)$

Dynamic Programming Principle

Theorem

Let x_0 be an initial state. Let $u^*(0), u^*(1), \ldots, u^*(N-1)$ be an optimal control for $\mathcal{P}_N(x_0)$ and $x_{u^*}(0), x_{u^*}(1), \ldots, x_{u^*}(N)$ be the corresponding optimal state trajectory.

Then for any i, $i = 0, 1, \ldots, N-1$, the control sequence $u^*(i), u^*(i+1), \ldots, u^*(N-1)$ is optimal control for $\mathcal{P}_{N-i}(x_{u^*}(i))$.

Dynamic Programming Principle

Theorem

Let x_0 be an initial state. Let $u^*(0), u^*(1), \ldots, u^*(N-1)$ be an optimal control for $\mathcal{P}_N(x_0)$ and $x_{u^*}(0), x_{u^*}(1), \ldots, x_{u^*}(N)$ be the corresponding optimal state trajectory.

Then for any i, $i = 0, 1, \ldots, N-1$, the control sequence $u^*(i), u^*(i+1), \ldots, u^*(N-1)$ is optimal control for $\mathcal{P}_{N-i}(x_{u^*}(i))$.

- tails of optimal control sequence are also optimal control sequences for adjusted initial value, time & horizon.

Dynamic Programming Principle

Theorem

Let x_0 be an initial state. Let $u^*(0), u^*(1), \ldots, u^*(N-1)$ be an optimal control for $\mathcal{P}_N(x_0)$ and $x_{u^*}(0), x_{u^*}(1), \ldots, x_{u^*}(N)$ be the corresponding optimal state trajectory.

Then for any i, $i = 0, 1, \ldots, N-1$, the control sequence $u^*(i), u^*(i+1), \ldots, u^*(N-1)$ is optimal control for $\mathcal{P}_{N-i}(x_{u^*}(i))$.

- tails of optimal control sequence are also optimal control sequences for adjusted initial value, time & horizon.

$$\begin{array}{cccccc} u^*(0) & u^*(1) & u^*(2) & \ldots & u^*(N-1) & \\ x^*(0) & x^*(1) & x^*(2) & \ldots & x^*(N-1) & x^*(N) \end{array}$$

Dynamic Programming Principle

Theorem

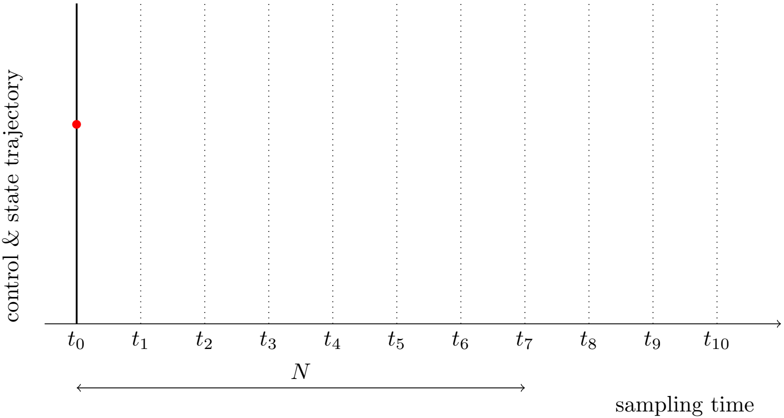
Let x_0 be an initial state. Let $u^*(0), u^*(1), \ldots, u^*(N-1)$ be an optimal control for $\mathcal{P}_N(x_0)$ and $x_{u^*}(0), x_{u^*}(1), \ldots, x_{u^*}(N)$ be the corresponding optimal state trajectory.

Then for any i, $i = 0, 1, \ldots, N-1$, the control sequence $u^*(i), u^*(i+1), \ldots, u^*(N-1)$ is optimal control for $\mathcal{P}_{N-i}(x_{u^*}(i))$.

- tails of optimal control sequence are also optimal control sequences for adjusted initial value, time & horizon.

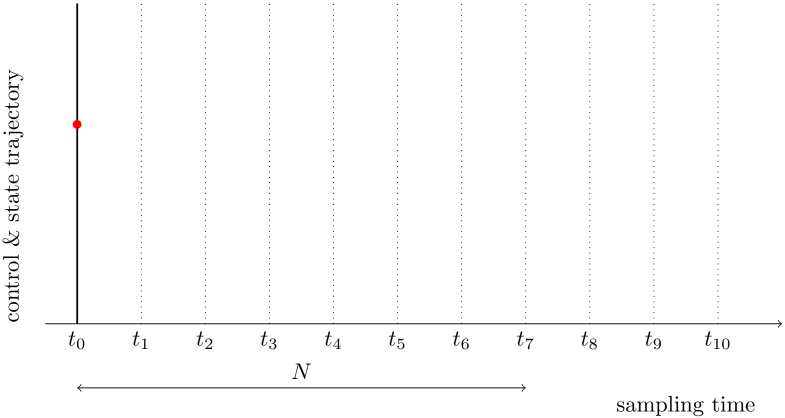
$$\begin{array}{ccccc} u^*(1) & u^*(2) & \ldots & u^*(N-1) & \\ x^*(1) & x^*(2) & \ldots & x^*(N-1) & x^*(N) \end{array}$$

Standard MPC Algorithm



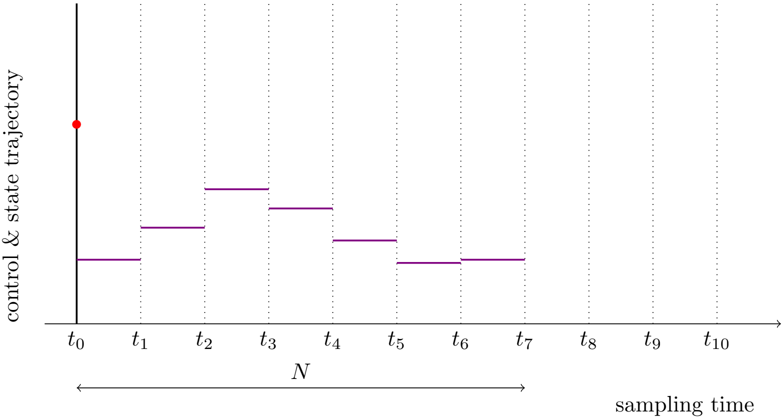
Given prediction horizon N and initial value x_0.

Standard MPC Algorithm



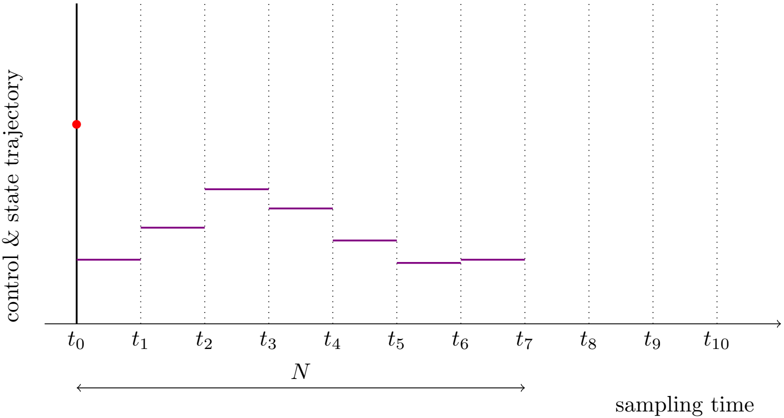
Solve the OCP $\mathcal{P}_N(x_0)$

Standard MPC Algorithm



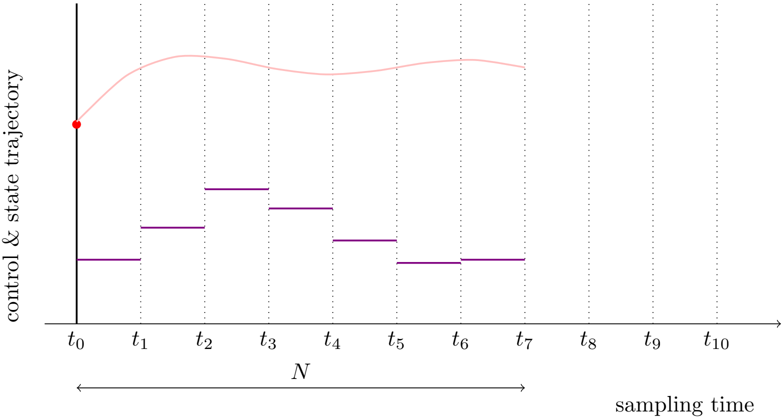
Solve the OCP $\mathcal{P}_N(x_0)$

Standard MPC Algorithm



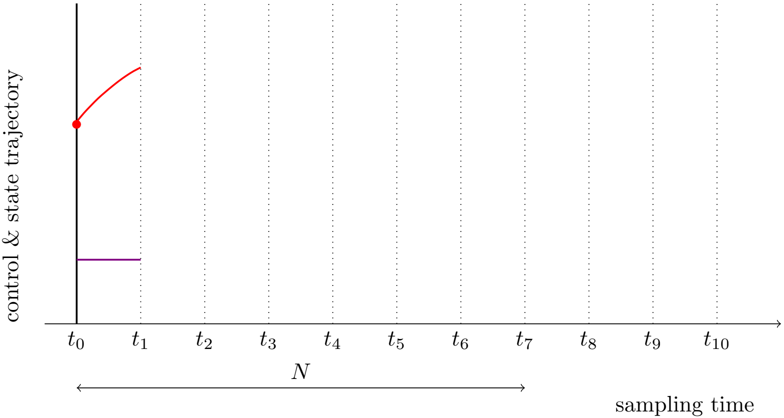
Denote optimal control sequence by $u^*(\cdot)$.

Standard MPC Algorithm



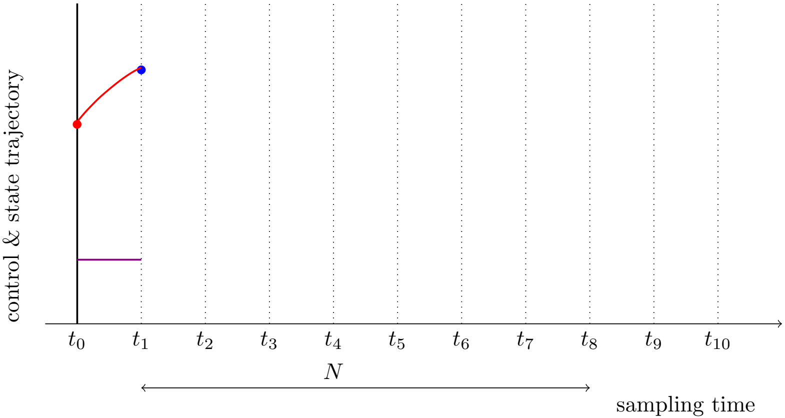
Denote optimal control sequence by $u^*(\cdot)$.

Standard MPC Algorithm



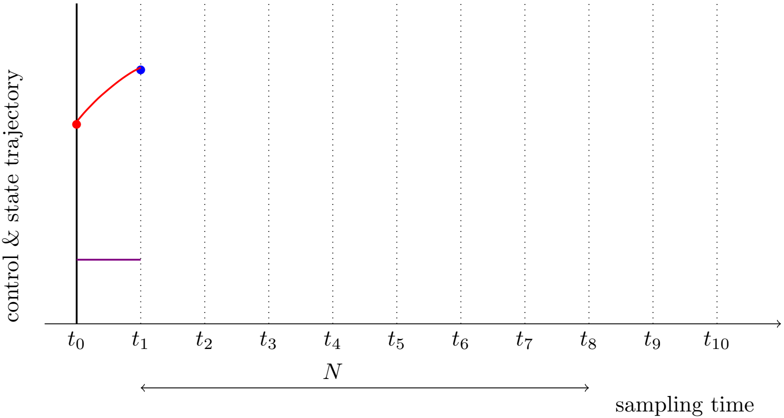
Define NMPC-feedback $\mu_N(x_0, t_0) := u^*(0)$.

Standard MPC Algorithm



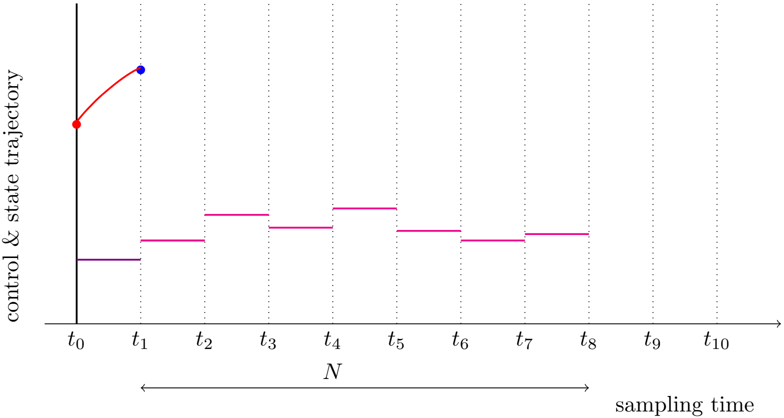
Use the feedback to generate the next state.

Standard MPC Algorithm



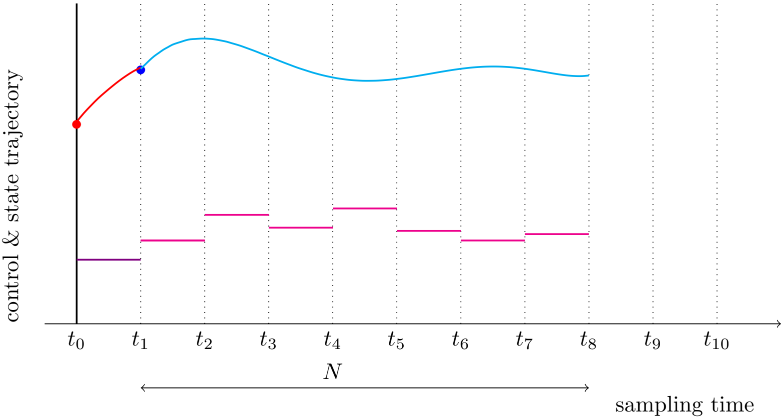
Repeat the process.

Standard MPC Algorithm



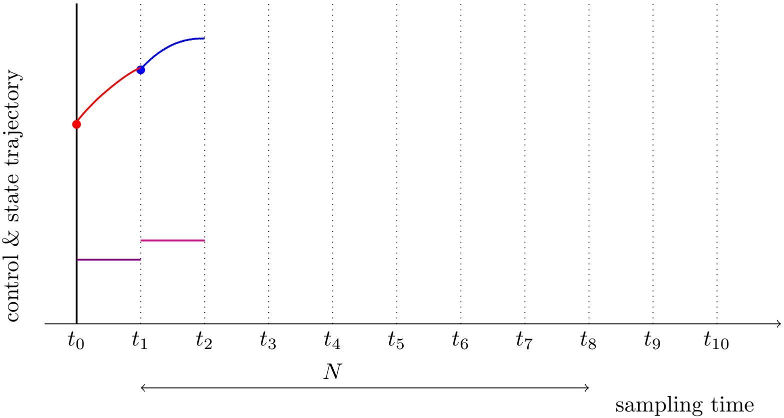
Repeat the process.

Standard MPC Algorithm



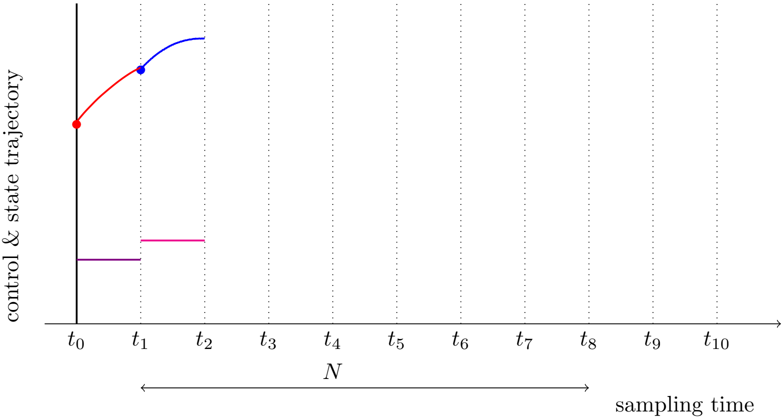
Repeat the process.

Standard MPC Algorithm



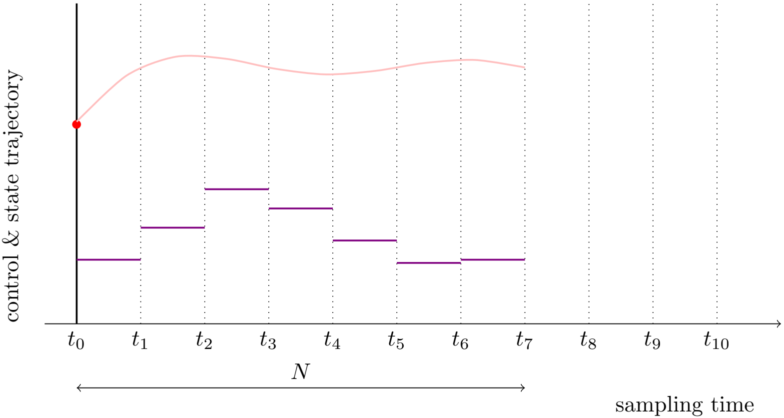
feedback law: μ_N trajectory/solution: $x_{\mu_N}(k, x_0)$

Standard MPC Algorithm

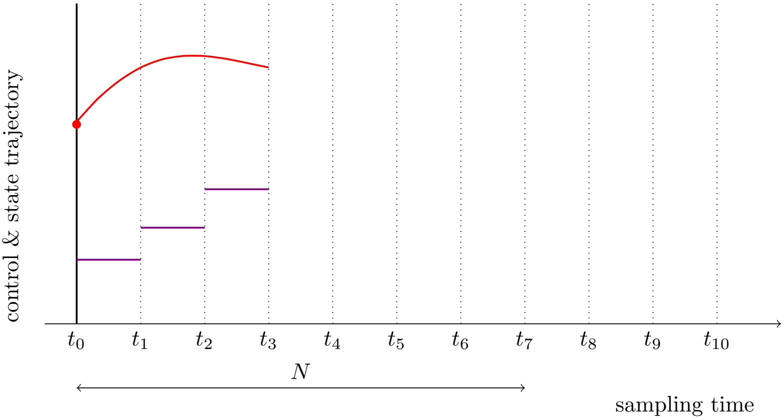


Challenge/Issue: Solving OCP every step is expensive.

Multistep or m-step MPC Algorithm

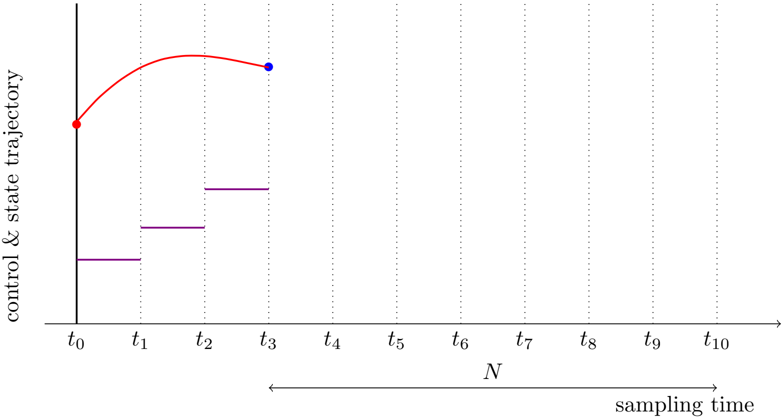


Multistep or m-step MPC Algorithm



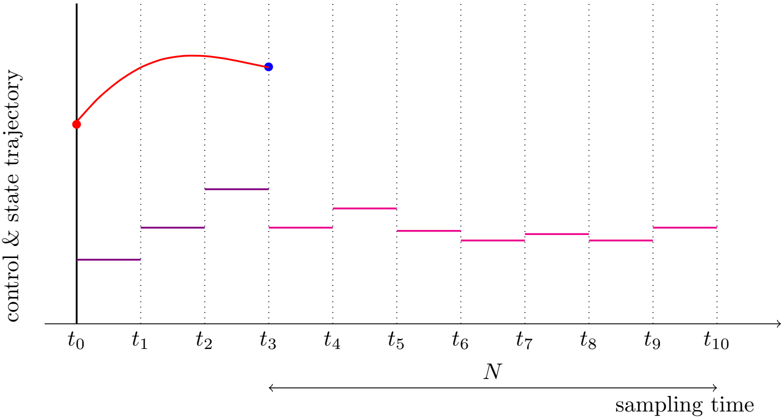
Multistep feedback - implement the first m elements.

Multistep or m-step MPC Algorithm



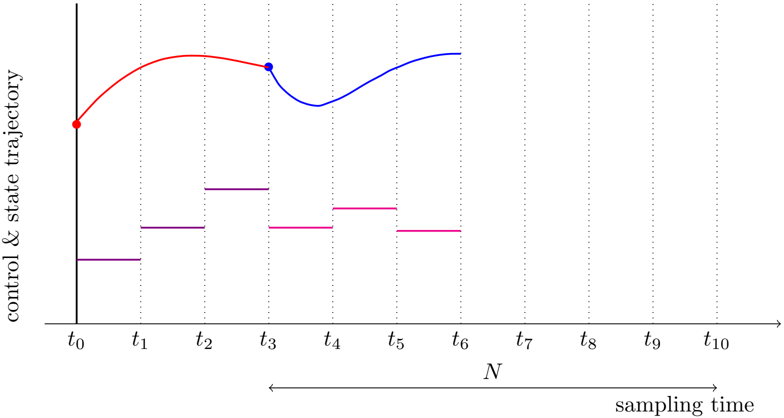
$u^\star(0), u^\star(1), \ldots, u^\star(m-1).$

Multistep or m-step MPC Algorithm



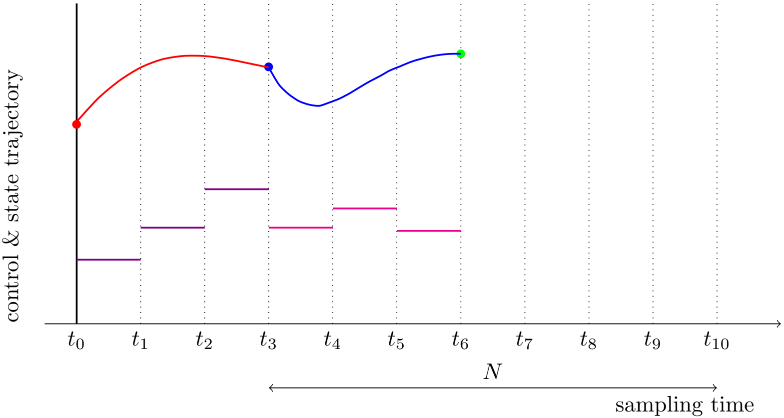
Then proceed with the next optimization.

Multistep or m-step MPC Algorithm



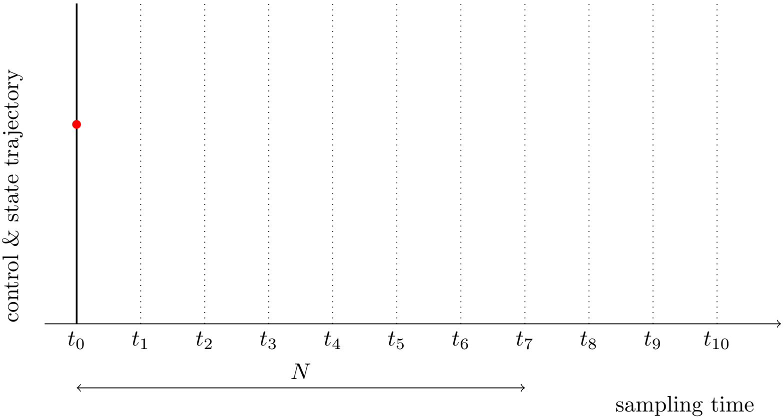
m is called **control horizon**.

Multistep or m-step MPC Algorithm



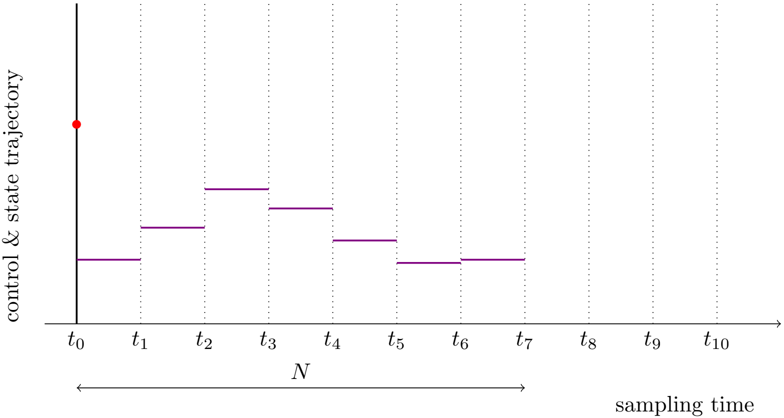
feedback law: $\mu_{N,m}$ trajectory/solution: $x_{\mu_{N,m}}(k, x_0)$

Updated m-step MPC Algorithm



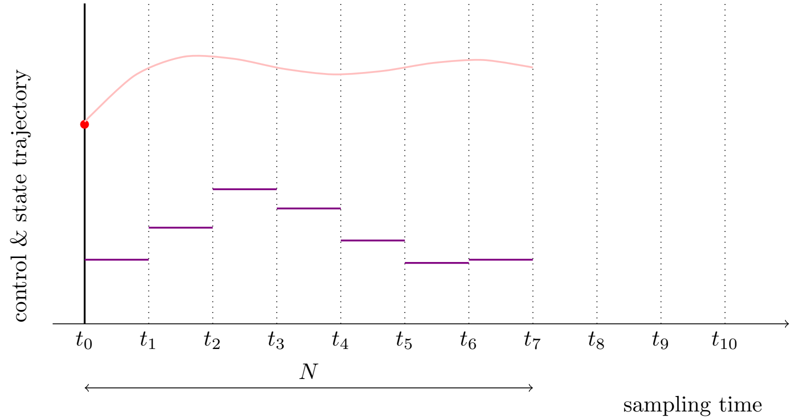
Given prediction horizon N and initial value x_0.

Updated m-step MPC Algorithm



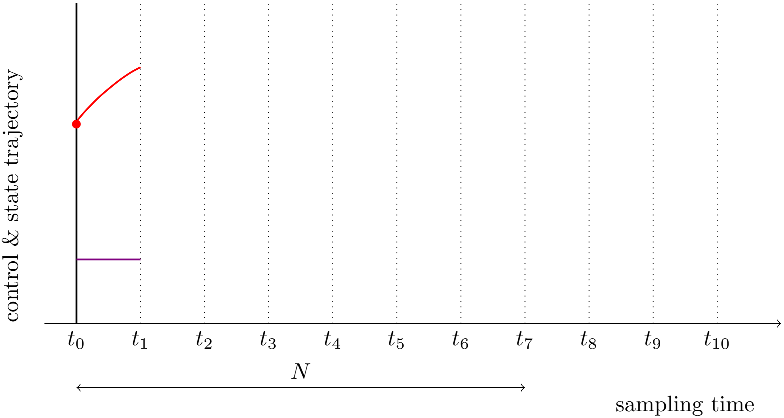
Solve $\mathcal{P}_N(x_0)$

Updated m-step MPC Algorithm



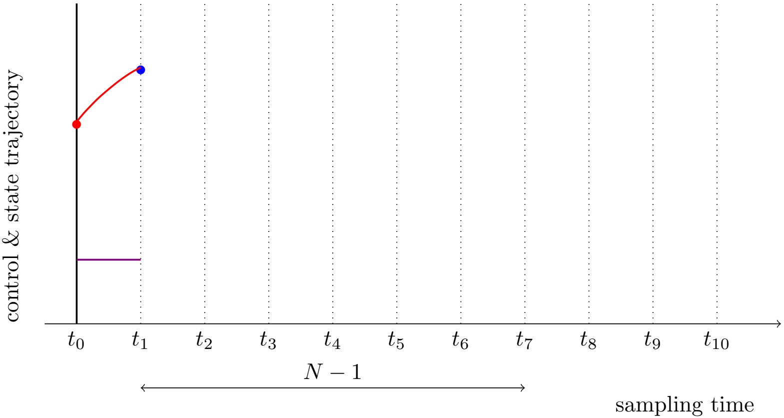
Solve $\mathcal{P}_N(x_0)$

Updated m-step MPC Algorithm



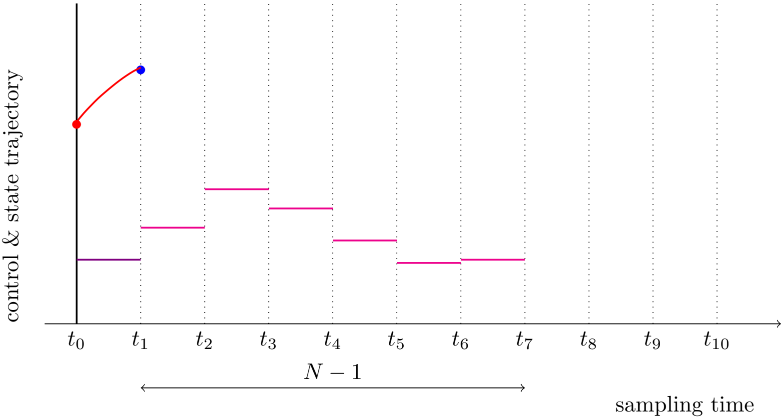
Implement the first element of the optimal control.

Updated m-step MPC Algorithm



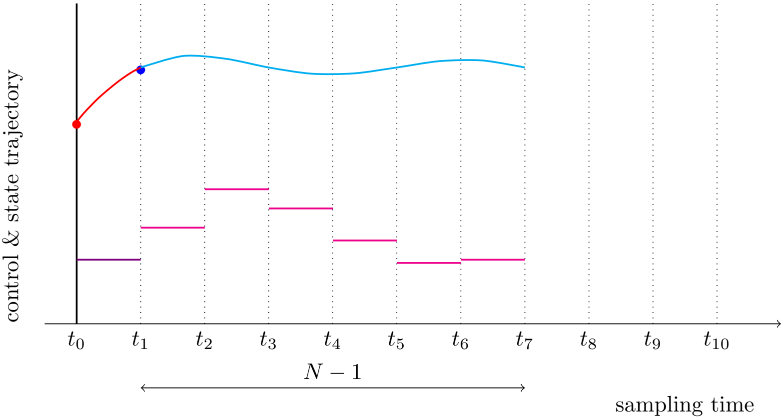
Re-optimize over shrunken horizon, i.e., solve $\mathcal{P}_{N-1}(x(t_1))$.

Updated m-step MPC Algorithm



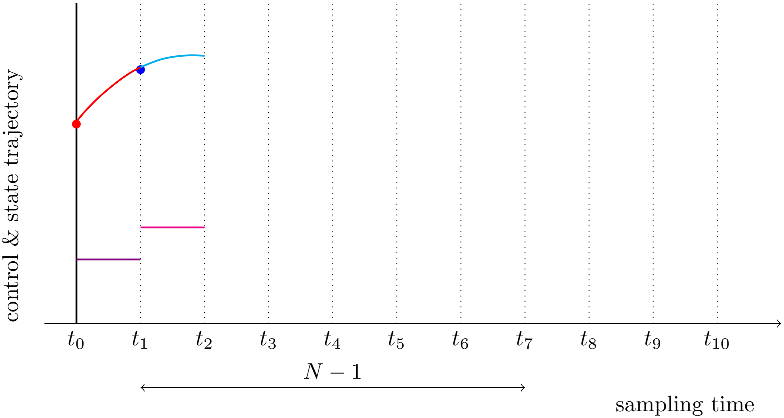
Re-optimize over shrunken horizon, i.e., solve $\mathcal{P}_{N-1}(x(t_1))$.

Updated m-step MPC Algorithm



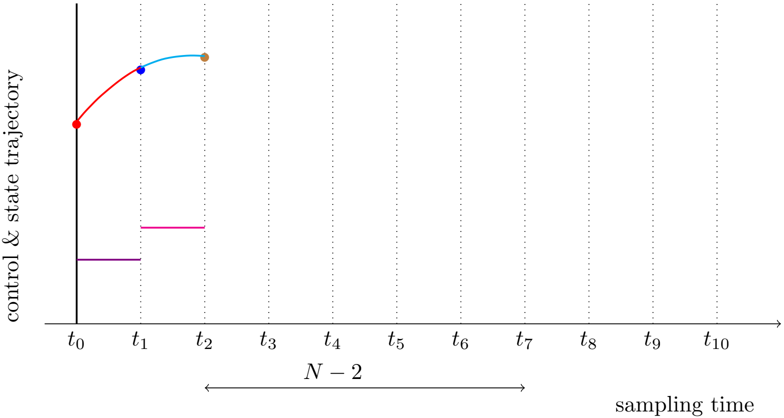
Re-optimize over shrunken horizon, i.e., solve $\mathcal{P}_{N-1}(x(t_1))$.

Updated m-step MPC Algorithm



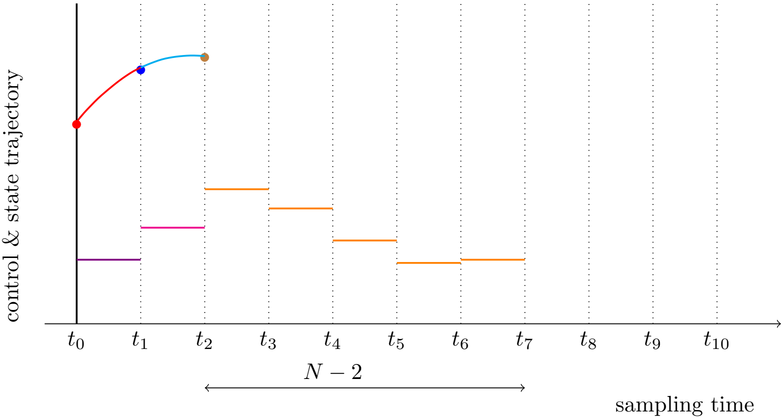
Implement the first element of the optimal control.

Updated m-step MPC Algorithm



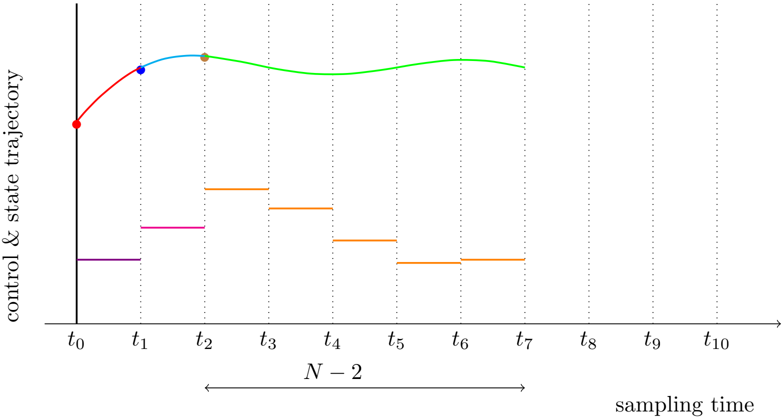
Solve $\mathcal{P}_{N-2}(x(t_2))$.

Updated m-step MPC Algorithm



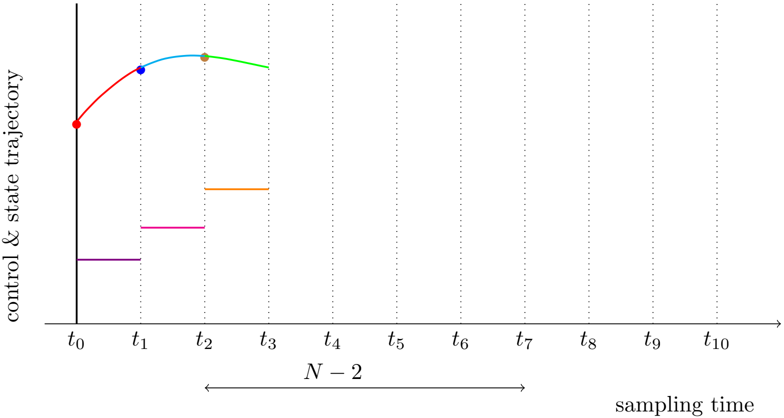
Solve $\mathcal{P}_{N-2}(x(t_2))$.

Updated m-step MPC Algorithm



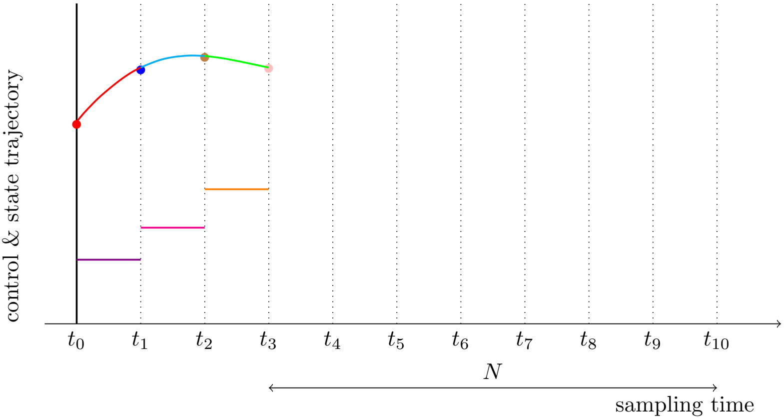
Solve $\mathcal{P}_{N-2}(x(t_2))$.

Updated m-step MPC Algorithm



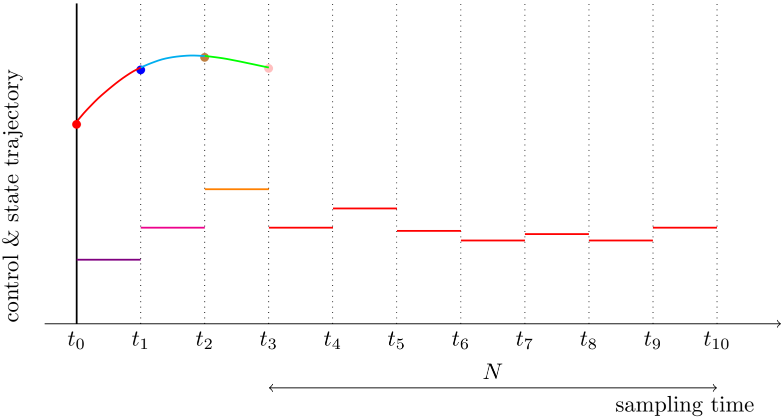
Implement the first element of the optimal control.

Updated m-step MPC Algorithm



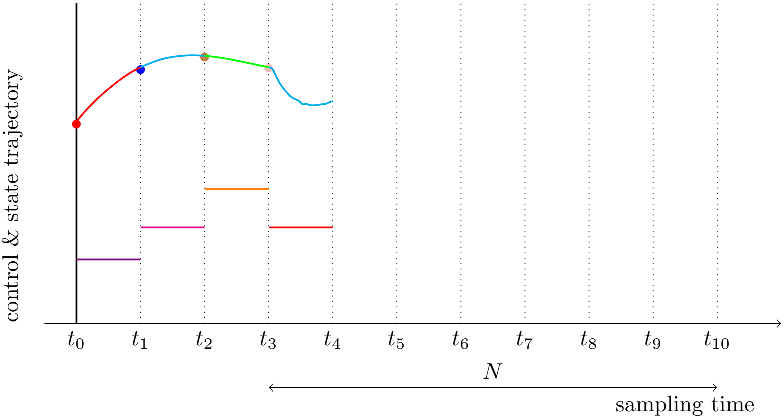
Solve $\mathcal{P}_N(x(t_3))$.

Updated m-step MPC Algorithm



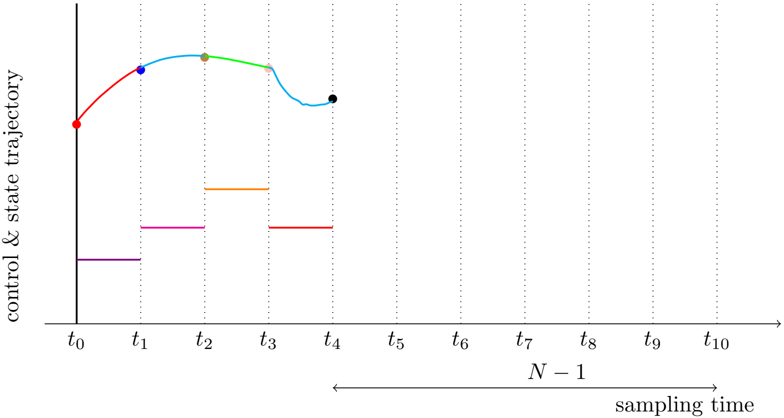
Solve $\mathcal{P}_N(x(t_3))$.

Updated m-step MPC Algorithm



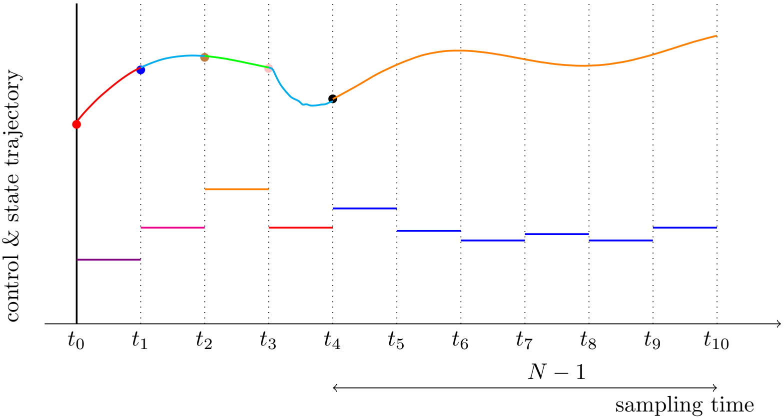
Implement the first element of the optimal control.

Updated m-step MPC Algorithm



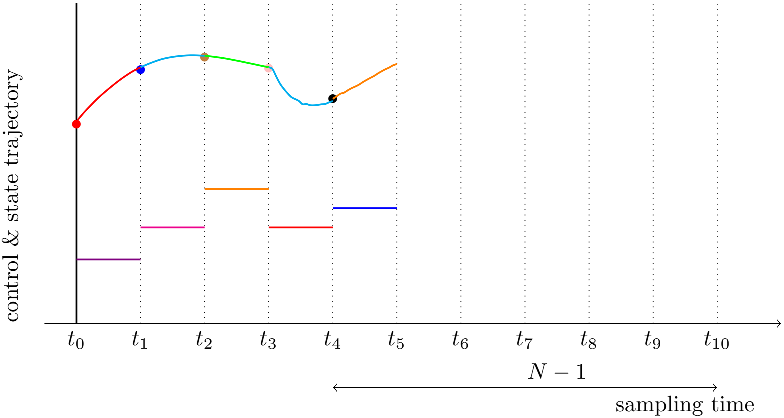
Solve $\mathcal{P}_{N-1}(x(t_4))$.

Updated m-step MPC Algorithm



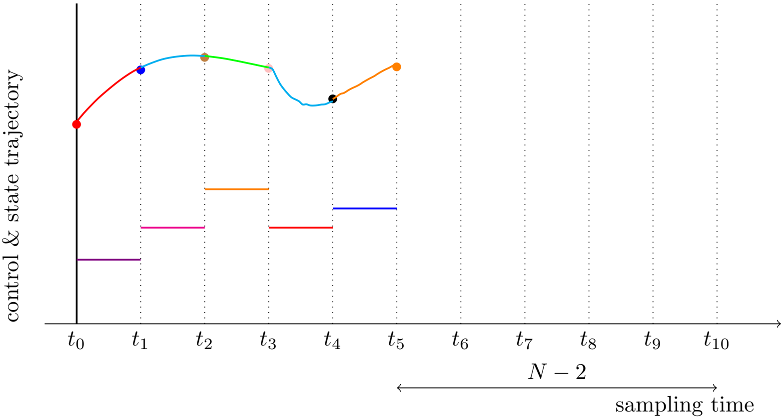
Solve $\mathcal{P}_{N-1}(x(t_4))$.

Updated m-step MPC Algorithm



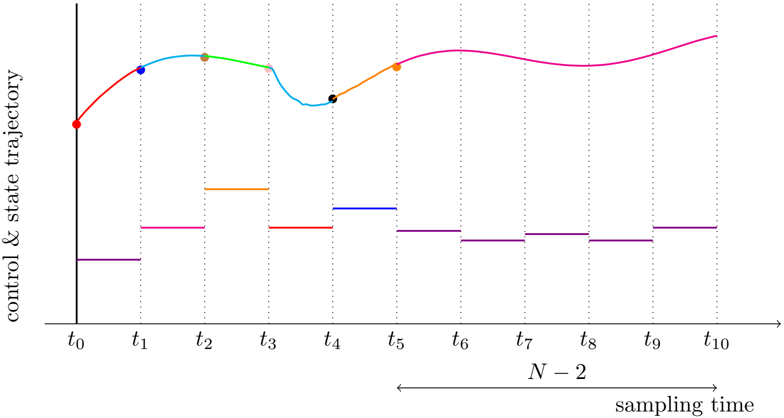
Implement the first element of the optimal control.

Updated m-step MPC Algorithm



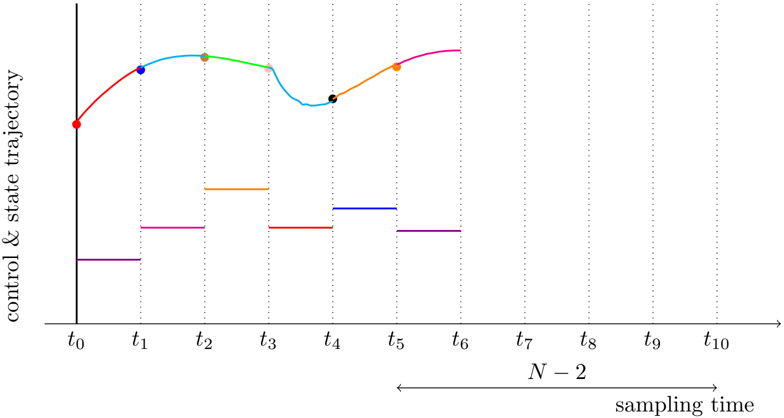
Solve $\mathcal{P}_{N-2}(x(t_5))$.

Updated m-step MPC Algorithm



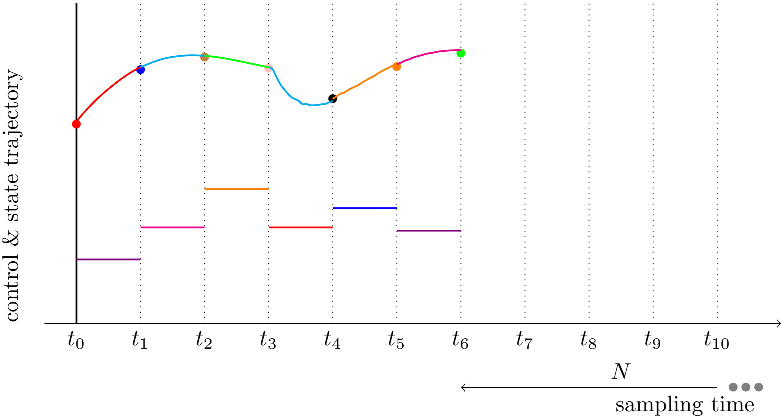
Solve $\mathcal{P}_{N-2}(x(t_5))$.

Updated m-step MPC Algorithm



Implement the first element of the optimal control.

Updated m-step MPC Algorithm



feedback law: $\hat{\mu}_{N,m}$ trajectory/solution: $x_{\hat{\mu}_{N,m}}(k, x_0)$

Nominal Setting

Inverted Pendulum

$$
\begin{aligned}
\dot{x}_1(t) &= x_2(t) \\
\dot{x}_2(t) &= -\frac{g}{\ell}\sin(x_1(t)) - \frac{k_L}{l}\arctan(1000x_2(t))x_2^2(t) - \frac{u(t)}{l}\cos(x_1(t)) \\
&\quad -k_R\left(\frac{4ax_2(t)}{1+4(ax_2(t))^2} + \frac{2\arctan(bx_2(t))}{\pi}\right) \\
\dot{x}_3(t) &= x_4(t) \\
\dot{x}_4(t) &= u(t)
\end{aligned}
$$

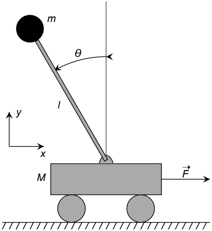


Image from http://en.wikipedia.org/wiki/Inverted_pendulum

Nominal Setting

Inverted Pendulum

$$
\begin{aligned}
\dot{x}_1(t) &= x_2(t) \\
\dot{x}_2(t) &= -\frac{g}{\ell}\sin(x_1(t)) - \frac{k_L}{l}\arctan(1000x_2(t))x_2^2(t) - \frac{u(t)}{l}\cos(x_1(t)) \\
&\quad -k_R\left(\frac{4ax_2(t)}{1+4(ax_2(t))^2} + \frac{2\arctan(bx_2(t))}{\pi}\right) \\
\dot{x}_3(t) &= x_4(t) \\
\dot{x}_4(t) &= u(t)
\end{aligned}
$$

- stabilize the upright position $x_* = ((2k+1)\pi, 0, 0, 0),\ k \in \mathbb{N}$,
- sampling period $T = 0.2$, prediction horizon $N = 15$, initial value $x_0 = (-\pi - 0.1, 0, -0.1, 0)$

Nominal Setting

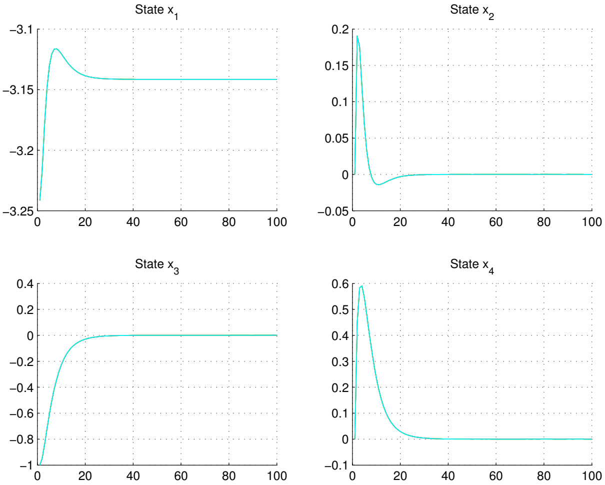


Figure: the standard MPC scheme (cyan), 1-step (red) and updated 1-step (green) MPC schemes

Nominal Setting

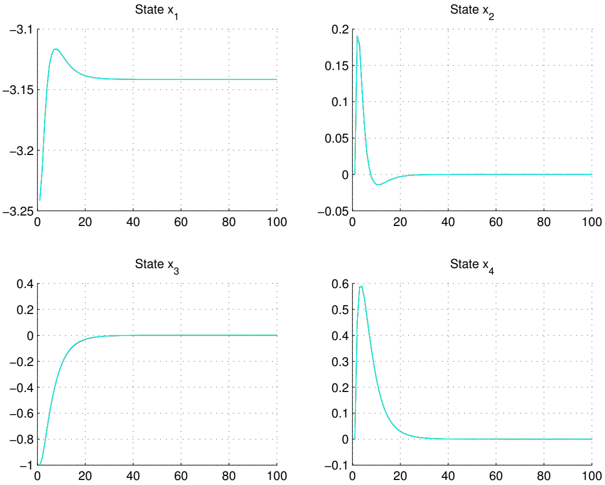


Figure: the standard MPC scheme (cyan), 2-step (red) and updated 2-step (green) MPC schemes

Nominal Setting

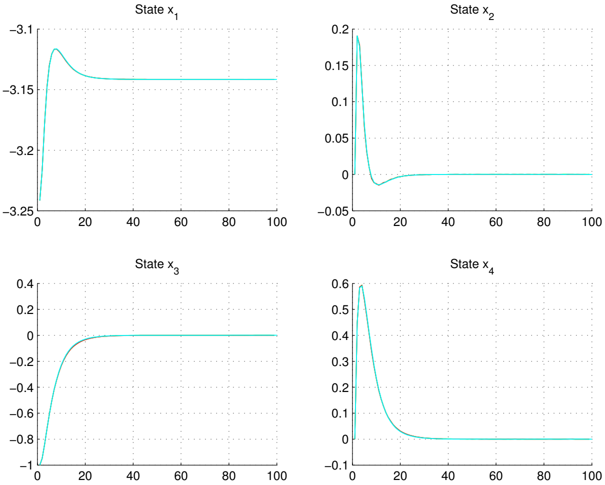


Figure: the standard MPC scheme (cyan), 3-step (red) and updated 3-step (green) MPC schemes

Nominal Setting

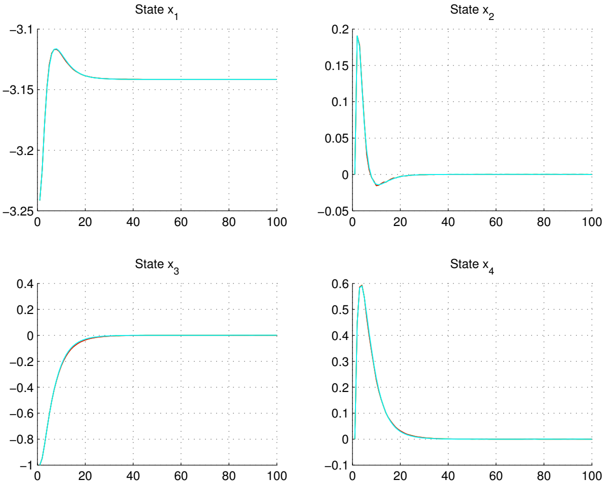


Figure: the standard MPC scheme (cyan), 4-step (red) and updated 4-step (green) MPC schemes

Nominal Setting

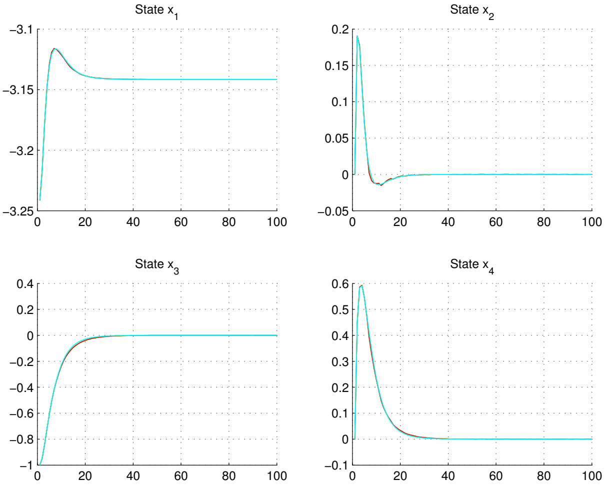


Figure: the standard MPC scheme (cyan), 5-step (red) and updated 5-step (green) MPC schemes

Nominal Setting

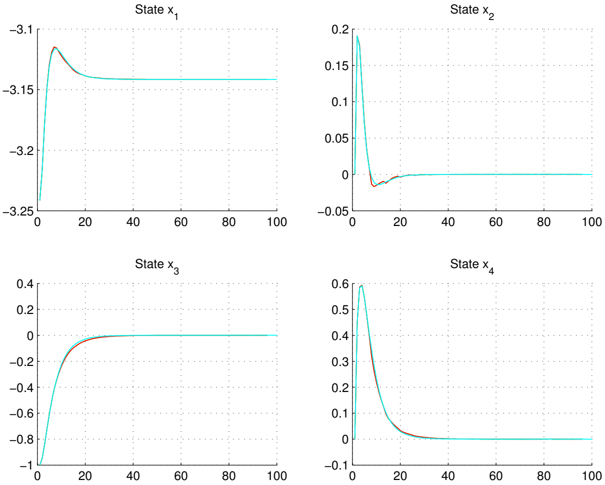
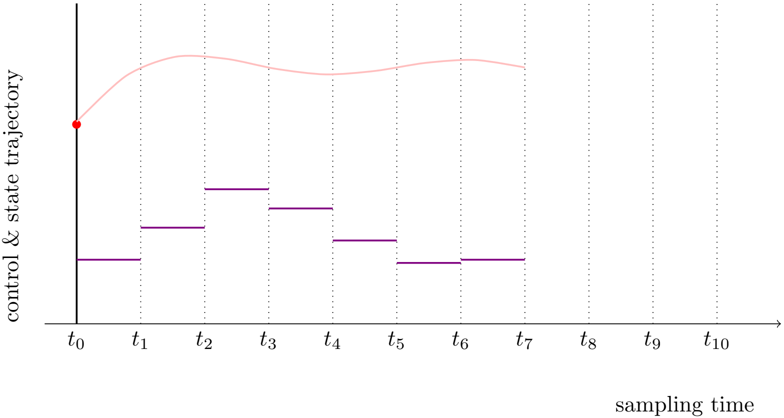
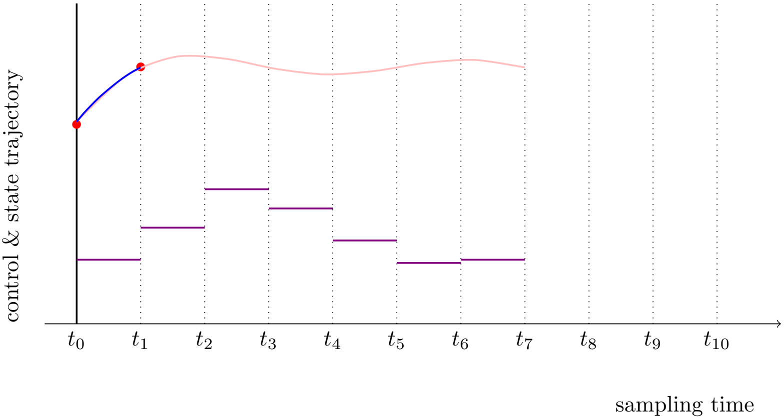


Figure: the standard MPC scheme (cyan), 6-step (red) and updated 6-step (green) MPC schemes

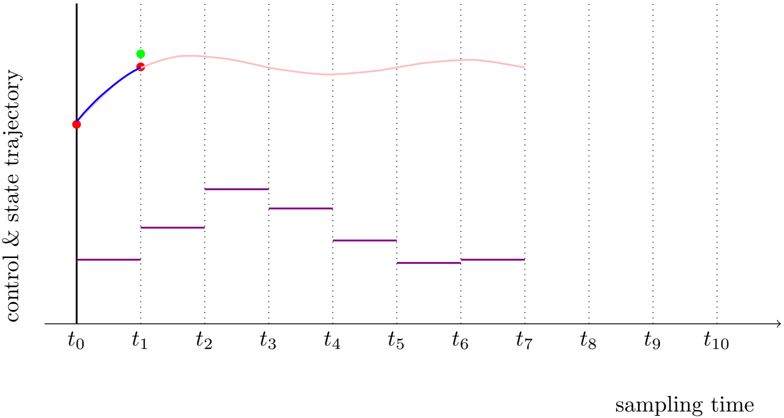
Perturbed Systems



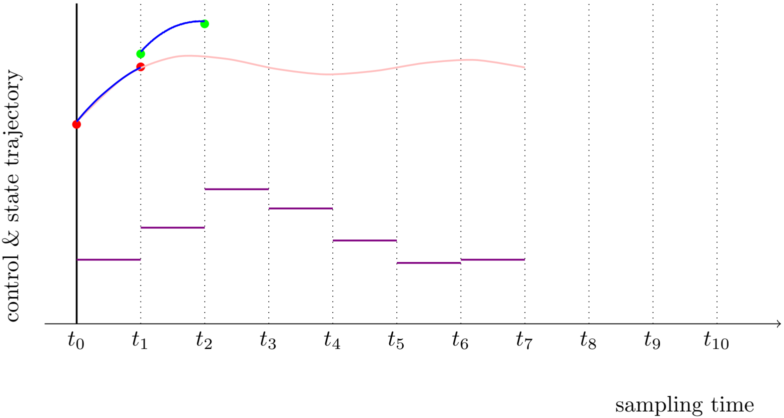
Perturbed Systems



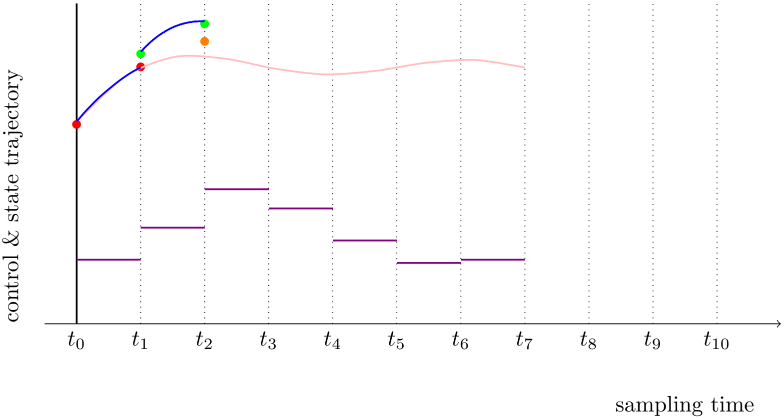
Perturbed Systems



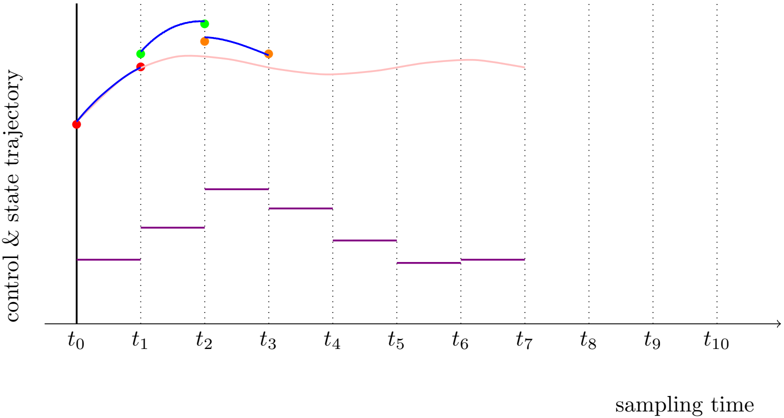
Perturbed Systems



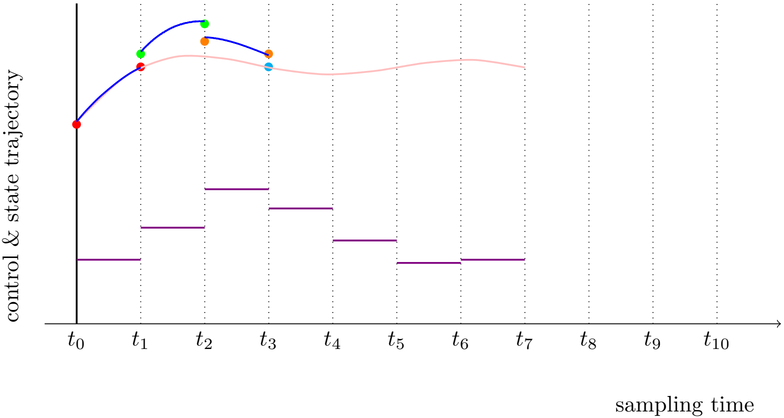
Perturbed Systems



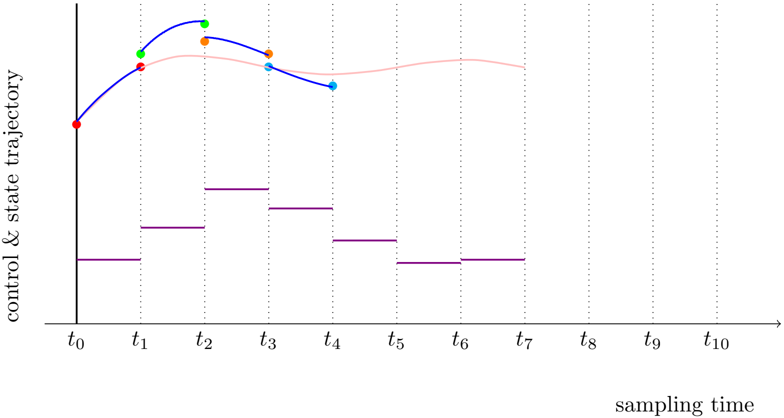
Perturbed Systems



Perturbed Systems



Perturbed Systems



Perturbed Systems

- control moves of the m-step MPC are based on old info

Perturbed Systems

- control moves of the m-step MPC are based on old info
- **nominal** closed-loop system

$$x(k+1) = f(x(k), \mu(x(\tilde{k}), k))$$

- **perturbed** closed-loop system

$$\tilde{x}(k+1) = f(\tilde{x}(k), \mu(\tilde{x}(\tilde{k}), k))) + d(k)$$

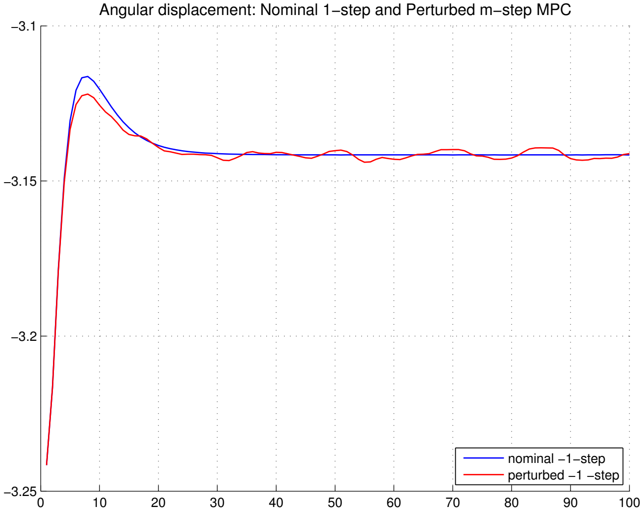
Perturbed Systems

Inverted Pendulum

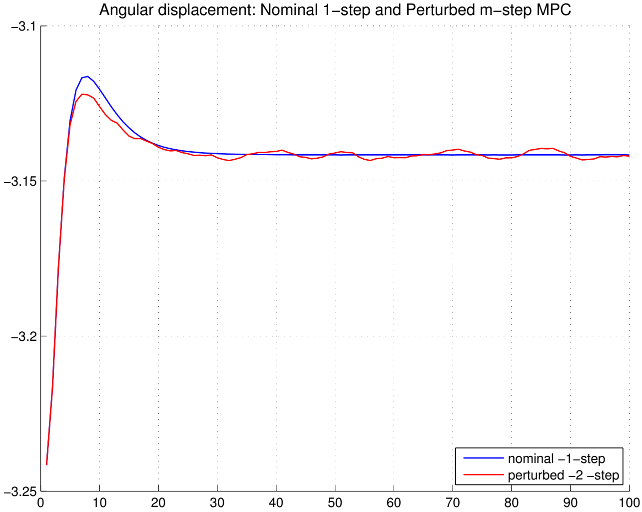
$$
\begin{aligned}
\dot{x}_1(t) &= x_2(t) \\
\dot{x}_2(t) &= -\frac{g}{\ell}\sin(x_1(t)) - \frac{k_L}{l}\arctan(1000x_2(t))x_2^2(t) - \frac{u(t)}{l}\cos(x_1(t)) \\
&\quad -k_R\left(\frac{4ax_2(t)}{1+4(ax_2(t))^2} + \frac{2\arctan(bx_2(t))}{\pi}\right) \\
\dot{x}_3(t) &= x_4(t) \\
\dot{x}_4(t) &= u(t)
\end{aligned}
$$

- stabilize the upright position $x_* = ((2k+1)\pi, 0, 0, 0),\ k \in \mathbb{N}$,
- sampling period $T = 0.2$, prediction horizon $N = 15$, initial value $x_0 = (-\pi - 0.1, 0, -0.1, 0)$
- randomly generated perturbation sequence $d(k) = [0, 0, d_3(k), 0]^\top, \quad d_3(k) \in [-0.05, 0]$

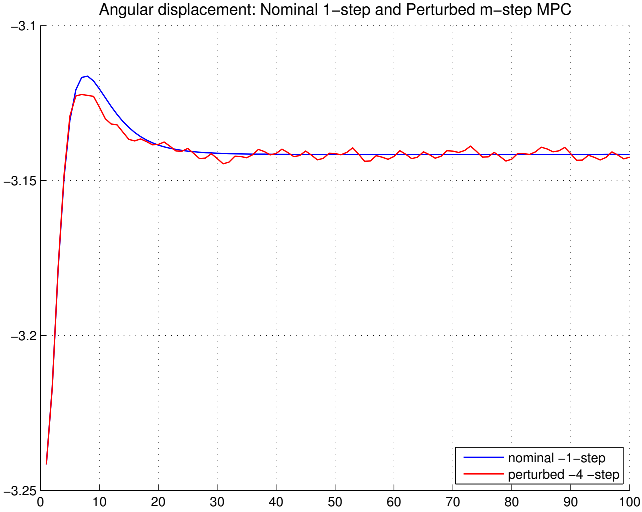
m-step MPC in Perturbed Setting



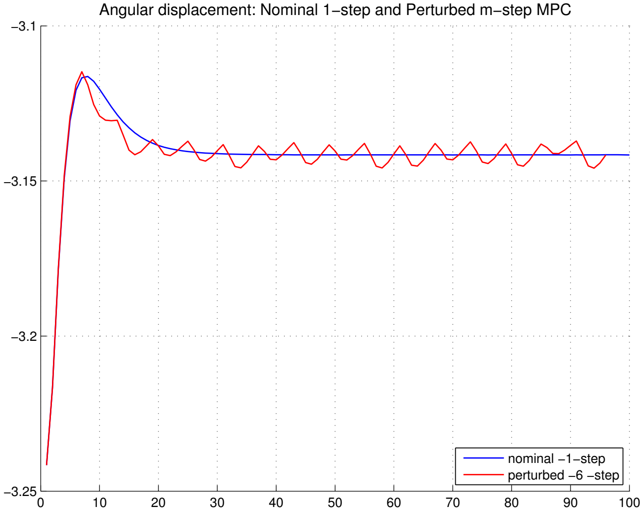
m-step MPC in Perturbed Setting



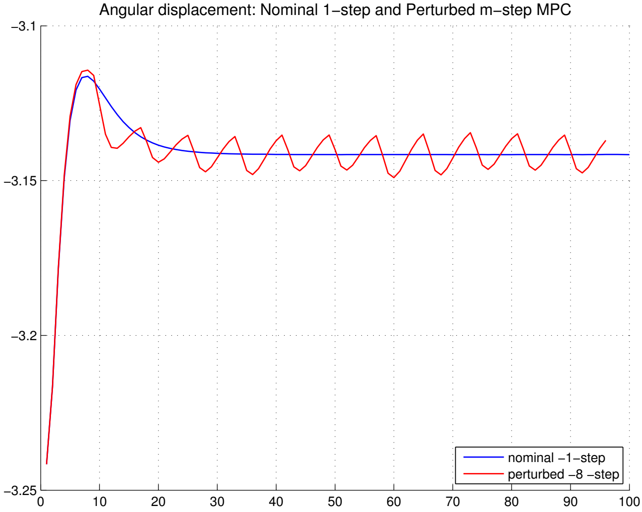
m-step MPC in Perturbed Setting



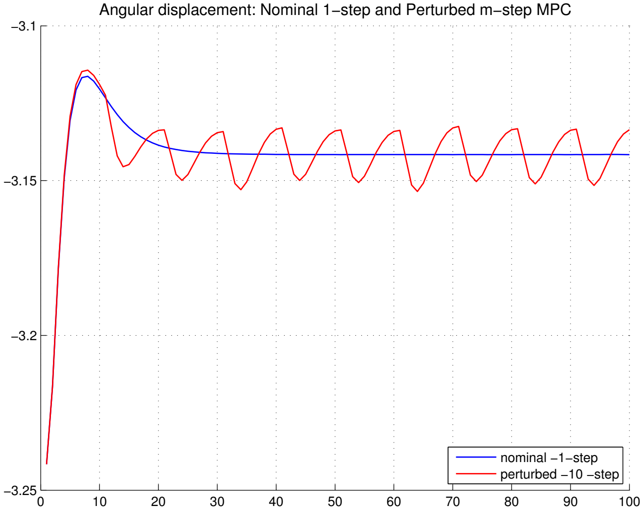
m-step MPC in Perturbed Setting



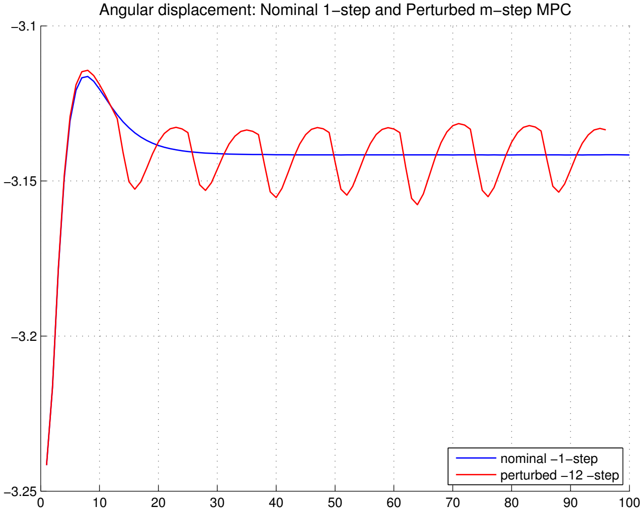
m-step MPC in Perturbed Setting



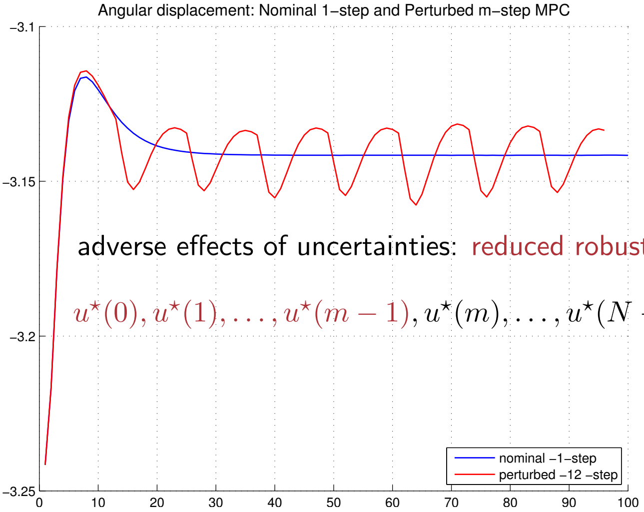
m-step MPC in Perturbed Setting



m-step MPC in Perturbed Setting



m-step MPC in Perturbed Setting



adverse effects of uncertainties: reduced robustness

$$u^\star(0), u^\star(1), \ldots, u^\star(m-1), u^\star(m), \ldots, u^\star(N-1)$$

Updated m-step MPC in Perturbed Setting

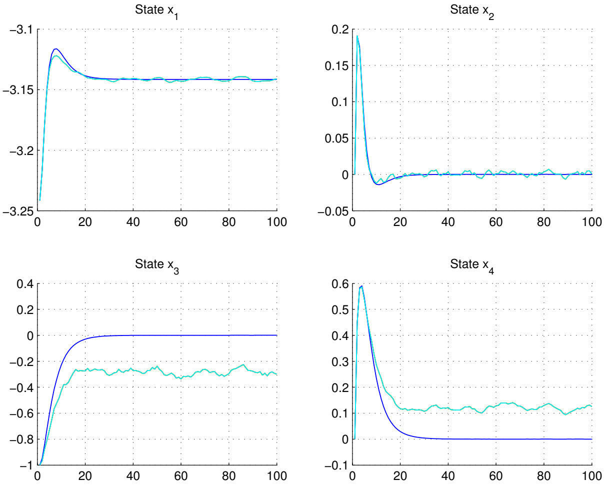


Figure: the nominal 1-step MPC scheme (blue), the standard MPC scheme (cyan), 1-step (red) and updated 1-step (green) MPC schemes for the perturbed system

Updated m-step MPC in Perturbed Setting

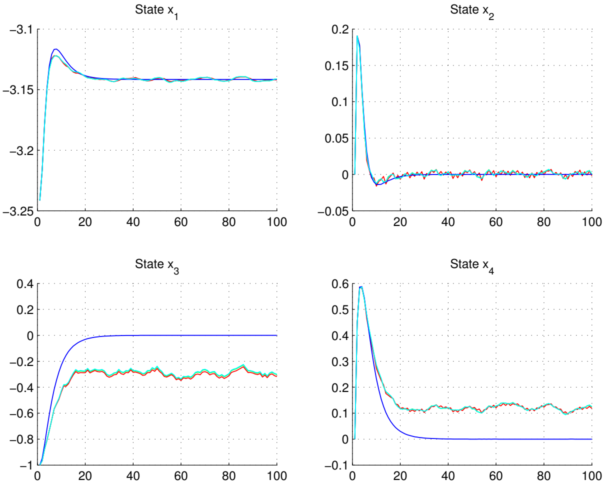


Figure: the nominal 2-step MPC scheme (blue), the standard MPC scheme (cyan), 2-step (red) and updated 2-step (green) MPC schemes for the perturbed system

Updated m-step MPC in Perturbed Setting

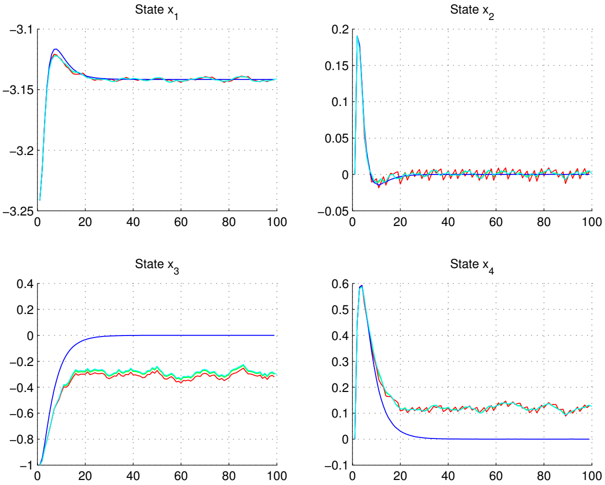


Figure: the nominal 3-step MPC scheme (blue), the standard MPC scheme (cyan), 3-step (red) and updated 3-step (green) MPC schemes for the perturbed system

Updated m-step MPC in Perturbed Setting

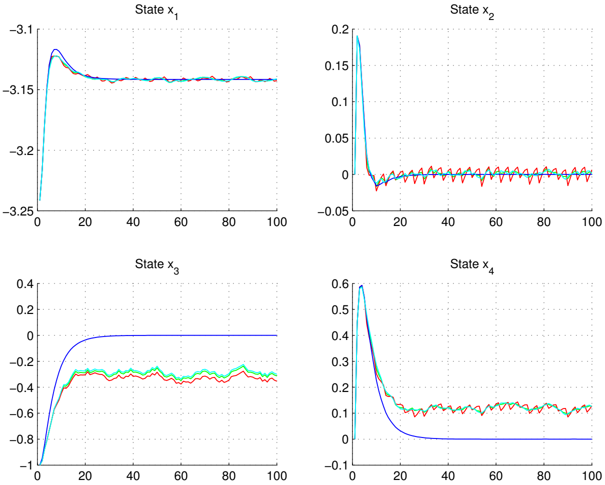


Figure: the nominal 4-step MPC scheme (blue), the standard MPC scheme (cyan), 4-step (red) and updated 4-step (green) MPC schemes for the perturbed system

Updated m-step MPC in Perturbed Setting

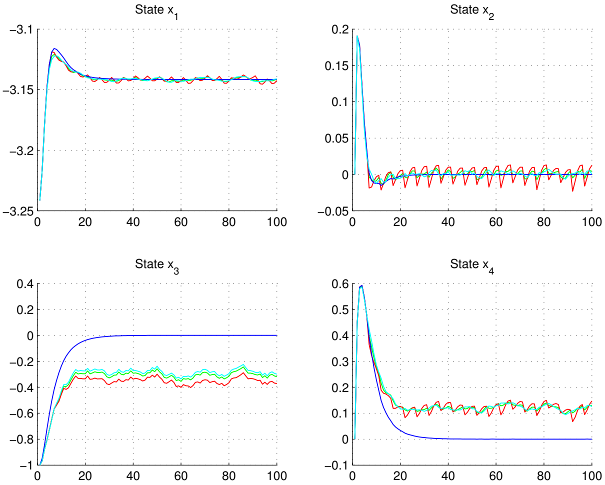


Figure: the nominal 5-step MPC scheme (blue), the standard MPC scheme (cyan), 5-step (red) and updated 5-step (green) MPC schemes for the perturbed system

Updated m-step MPC in Perturbed Setting

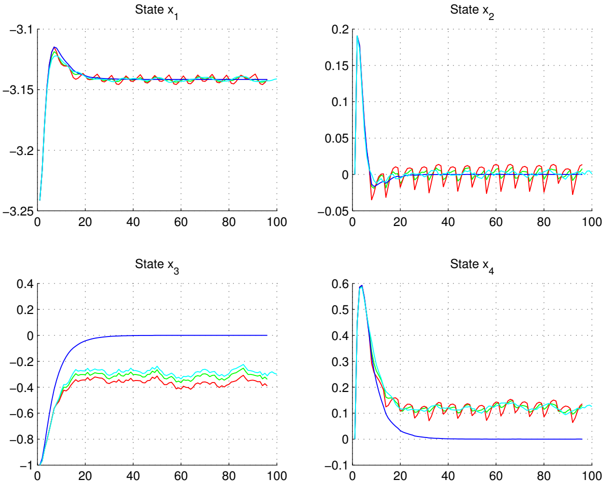


Figure: the nominal 6-step MPC scheme (blue), the standard MPC scheme (cyan), 6-step (red) and updated 6-step (green) MPC schemes for the perturbed system

Updated m-step MPC in Perturbed Setting

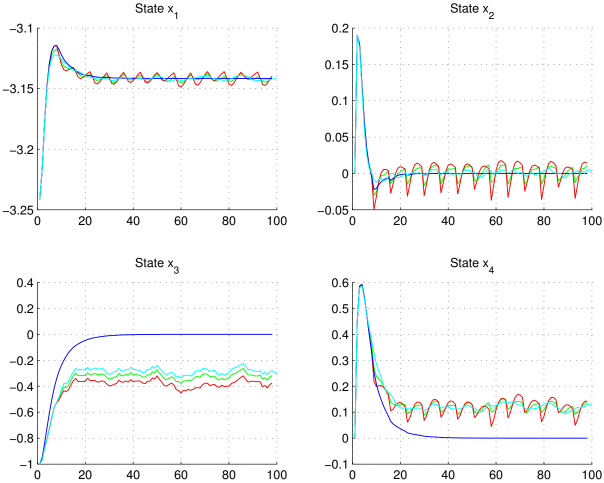


Figure: the nominal 7-step MPC scheme (blue), the standard MPC scheme (cyan), 7-step (red) and updated 7-step (green) MPC schemes for the perturbed system

Updated m-step MPC in Perturbed Setting

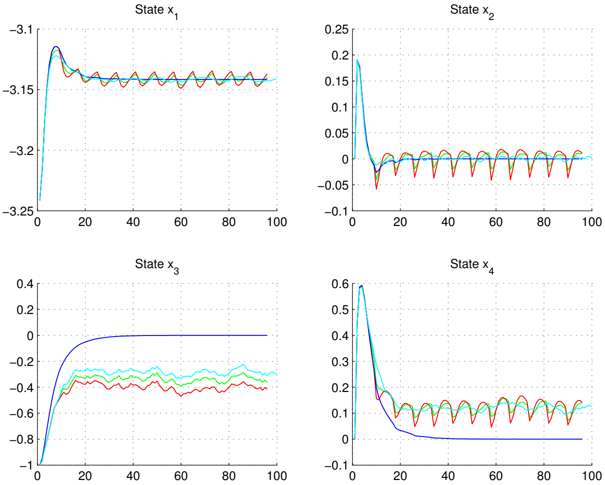


Figure: the nominal 8-step MPC scheme (blue), the standard MPC scheme (cyan), 8-step (red) and updated 8-step (green) MPC schemes for the perturbed system

Updated m-step MPC in Perturbed Setting

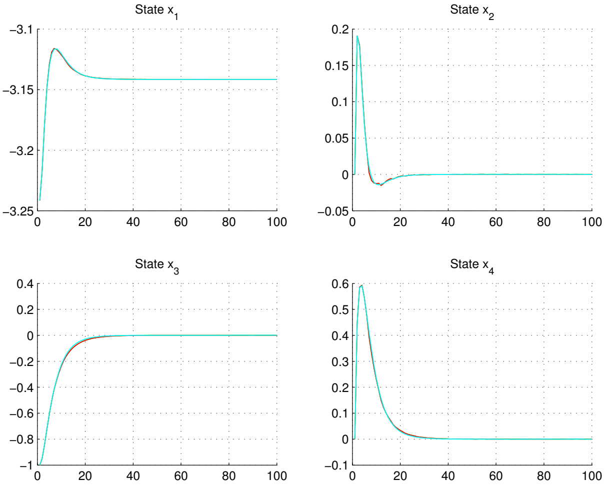


Figure: $\|d_3(k)\| = 0$, the nominal 5-step MPC scheme (blue), the standard MPC scheme (cyan), 5-step (red) and updated 5-step (green) MPC schemes for the perturbed system

Updated m-step MPC in Perturbed Setting

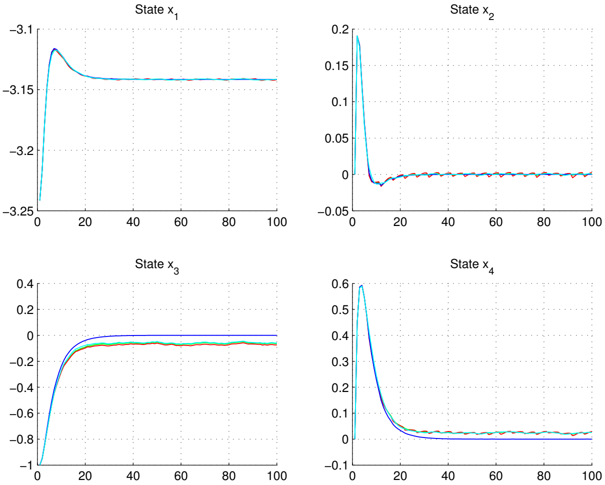


Figure: $\|d_3(k)\| = 10^{-2}$, the nominal 5-step MPC scheme (blue), the standard MPC scheme (cyan), 5-step (red) and updated 5-step (green) MPC schemes for the perturbed system

Updated m-step MPC in Perturbed Setting

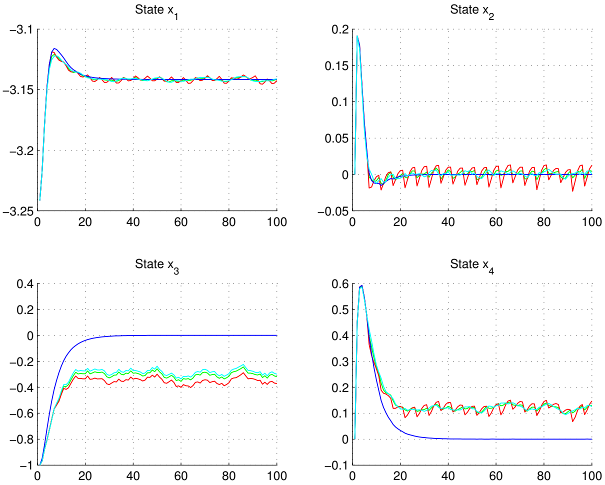


Figure: $\|d_3(k)\| = 10^{-1}/2$, the nominal 5-step MPC scheme (blue), the standard MPC scheme (cyan), 5-step (red) and updated 5-step (green) MPC schemes for the perturbed system

Updated m-step MPC in Perturbed Setting

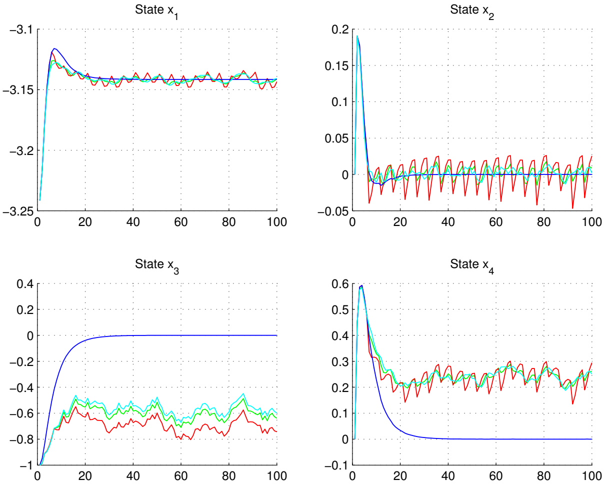


Figure: $\|d_3(k)\| = 10^{-1}$, the nominal 5-step MPC scheme (blue), the standard MPC scheme (cyan), 5-step (red) and updated 5-step (green) MPC schemes for the perturbed system

Updated m-step MPC in Perturbed Setting

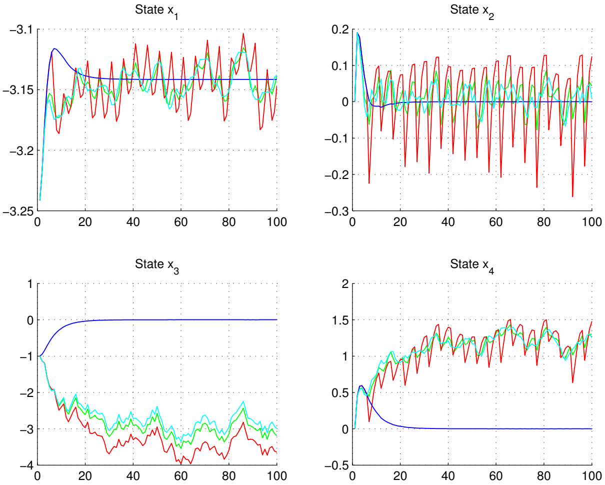


Figure: $\|d_3(k)\| = 10^{-0}/2$, the nominal 5-step MPC scheme (blue), the standard MPC scheme (cyan), 5-step (red) and updated 5-step (green) MPC schemes for the perturbed system

Updated m-step MPC in Perturbed Setting

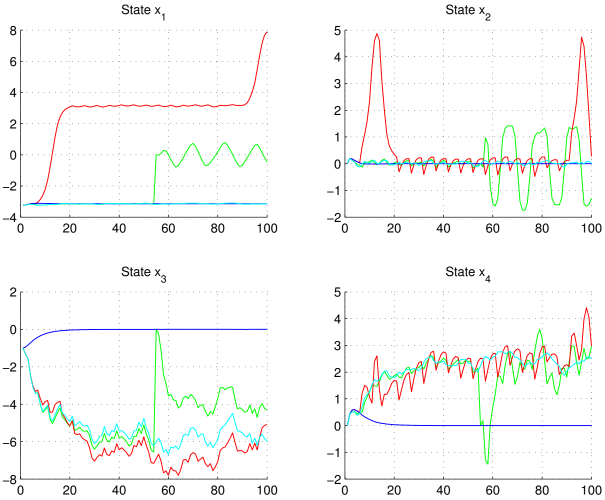


Figure: $\|d_3(k)\| = 10^{-0}$, the nominal 5-step MPC scheme (blue), the standard MPC scheme (cyan), 5-step (red) and updated 5-step (green) MPC schemes for the perturbed system

Theoretical Results

- Main Question:

  How can we quantify the benefit of the updates (i.e., re-optimization) of the updated MPC scheme?

Theoretical Results

- Main Question:

  How can we quantify the benefit of the updates
  (i.e., re-optimization) of the updated MPC scheme?

- Analyze the feedback laws $\mu_{N,m}$ and $\hat{\mu}_{N,m}$

Comparison Functions

$$\mathcal{K} = \left\{ \rho : \mathbb{R}_0^+ \to \mathbb{R}_0^+ \;\middle|\; \begin{array}{c} \rho \text{ is continuous, } \rho(0) = 0 \\ \text{and is strictly increasing} \end{array} \right\}$$

Comparison Functions

$$\mathcal{K} = \left\{ \rho : \mathbb{R}_0^+ \to \mathbb{R}_0^+ \;\middle|\; \begin{array}{c} \rho \text{ is continuous, } \rho(0) = 0 \\ \text{and is strictly increasing} \end{array} \right\}$$

$$\mathcal{K}_\infty = \left\{ \rho : \mathbb{R}_0^+ \to \mathbb{R}_0^+ \;\middle|\; \rho \in \mathcal{K} \text{ and is unbounded} \right\}$$

Comparison Functions

$$\mathcal{K} = \left\{ \rho : \mathbb{R}_0^+ \to \mathbb{R}_0^+ \;\middle|\; \begin{array}{c} \rho \text{ is continuous, } \rho(0) = 0 \\ \text{and is strictly increasing} \end{array} \right\}$$

$$\mathcal{K}_\infty = \left\{ \rho : \mathbb{R}_0^+ \to \mathbb{R}_0^+ \;\middle|\; \rho \in \mathcal{K} \text{ and is unbounded} \right\}$$

$$\mathcal{KL} = \left\{ \beta : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+ \;\middle|\; \begin{array}{c} \beta \text{ is continuous,} \\ \forall r \lim_{t\to\infty} \beta(r,t) = 0 \text{ and} \\ \forall t \geq 0,\ \beta(\cdot, t) \in \mathcal{K}_\infty \end{array} \right\}$$

Comparison Functions

$$\mathcal{K} = \left\{ \rho : \mathbb{R}_0^+ \to \mathbb{R}_0^+ \;\middle|\; \begin{array}{c} \rho \text{ is continuous, } \rho(0) = 0 \\ \text{and is strictly increasing} \end{array} \right\}$$

$$\mathcal{K}_\infty = \left\{ \rho : \mathbb{R}_0^+ \to \mathbb{R}_0^+ \;\middle|\; \rho \in \mathcal{K} \text{ and is unbounded } \right\}$$

$$\mathcal{KL} = \left\{ \beta : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+ \;\middle|\; \begin{array}{c} \beta \text{ is continuous,} \\ \forall r \lim_{t\to\infty} \beta(r,t) = 0 \text{ and} \\ \forall t \geq 0,\ \beta(\cdot, t) \in \mathcal{K}_\infty \end{array} \right\}$$

$$\mathcal{KL}_0 = \left\{ \beta : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+ \;\middle|\; \begin{array}{c} \beta \text{ is continuous,} \\ \forall r \lim_{t\to\infty} \beta(r,t) = 0 \text{ and} \\ \forall t \geq 0, \beta(\cdot, t) \in \mathcal{K}_\infty \\ \text{or } \beta(\cdot, t) \equiv 0 \end{array} \right\}$$

Stabilizing MPC without terminal constraints

- Aim: to yield a controller stabilizing a given equilibrium

Stabilizing MPC without terminal constraints

- Aim: to yield a controller stabilizing a given equilibrium

x^* is **asymptotically stable** if there exists $\beta \in \mathcal{KL}$ s.t.

$$\|x_\mu(k, x_0)\|_{x^*} \leq \beta(\|x_0\|_{x^*}, k)$$

for all $x_0 \in \mathbb{X}$ and all $k \in \mathbb{N}$.

Assumption 1 (A1)

There exist $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ s.t.

$$\alpha_1(\|x\|_{x_*}) \leq \ell^*(x) \leq \alpha_2(\|x\|_{x_*})$$

for all $x \in \mathbb{X}$, where $\ell^*(x) \;:=\; \inf_{u \in \mathbb{U}} \ell(x, u)$.

Stabilizing MPC without terminal constraints

Proposition 1 (P1)

Consider time-dependent $\mu : \mathbb{X} \times \mathbb{N}_0 \to U$ and $V : X \to \mathbb{R}_0^+$ satisfying the relaxed dynamic programming inequality

$$V(x_0) \geq V(x_\mu(m)) + \alpha \sum_{k=0}^{m-1} \ell(x_\mu(k), \mu(x_\mu(\tilde{k}), k))$$

for some $\alpha \in (0, 1]$, some $m \geq 1$ and all $x_0 = x_\mu(0) \in \mathbb{X}$.

Stabilizing MPC without terminal constraints

Proposition 1 (P1)

Consider time-dependent $\mu : \mathbb{X} \times \mathbb{N}_0 \to U$ and $V : X \to \mathbb{R}_0^+$ satisfying the relaxed dynamic programming inequality

$$V(x_0) \geq V(x_\mu(m)) + \alpha \sum_{k=0}^{m-1} \ell(x_\mu(k), \mu(x_\mu(\tilde{k}), k))$$

for some $\alpha \in (0, 1]$, some $m \geq 1$ and all $x_0 = x_\mu(0) \in \mathbb{X}$. Then for all $x \in \mathbb{X}$, the ff. estimate holds:

$$V_\infty(x) \leq J_\infty^{\mathsf{cl}}(x, \mu) \leq V(x)/\alpha$$

Stabilizing MPC without terminal constraints

Proposition 1 (P1)

Consider time-dependent $\mu : \mathbb{X} \times \mathbb{N}_0 \to U$ and $V : X \to \mathbb{R}_0^+$ satisfying the relaxed dynamic programming inequality

$$V(x_0) \geq V(x_\mu(m)) + \alpha \sum_{k=0}^{m-1} \ell(x_\mu(k), \mu(x_\mu(\tilde{k}), k))$$

for some $\alpha \in (0, 1]$, some $m \geq 1$ and all $x_0 = x_\mu(0) \in \mathbb{X}$.
Then for all $x \in \mathbb{X}$, the ff. estimate holds:

$$V_\infty(x) \leq J_\infty^{\mathsf{cl}}(x, \mu) \leq V(x)/\alpha$$

If (A1) holds and $\exists\, \alpha_3, \alpha_4 \in \mathcal{K}_\infty$ with

$$\alpha_3(\|x\|_{x_*}) \leq V(x) \leq \alpha_4(\|x\|_{x_*})$$

then x_* is asymptotically stable for the closed-loop system.

Controllability Assumption

- Show: $\mu_{N,m}$ and V_N satisfy the RDPI.

Controllability Assumption

- Show: $\mu_{N,m}$ and V_N satisfy the RDPI.
- How do we compute α?

Controllability Assumption

- Show: $\mu_{N,m}$ and V_N satisfy the RDPI.
- How do we compute α?

Assumption 2 (A2)

There exists $B_k \in \mathcal{K}_\infty$ such that the optimal value functions of $\mathcal{P}_k(x_0)$ satisfy

$$V_k(x) \leq B_k(\ell^*(x)) \quad \text{for all} \quad x \in \mathbb{X} \text{ and all} \quad k = 2, \ldots, N$$

Controllability Assumption

- Show: $\mu_{N,m}$ and V_N satisfy the RDPI.
- How do we compute α?

Assumption 2 (A2)

There exists $B_k \in \mathcal{K}_\infty$ such that the optimal value functions of $\mathcal{P}_k(x_0)$ satisfy

$$V_k(x) \leq B_k(\ell^*(x)) \text{ for all } x \in \mathbb{X} \text{ and all } k = 2, \dots, N$$

- the system is exponentially controllable w.r.t. ℓ

$\Longrightarrow$ existence of such a $B_k \in \mathcal{K}_\infty$

Controllability Assumption

- Show: $\mu_{N,m}$ and V_N satisfy the RDPI.
- How do we compute α?

Assumption 2 (A2)

There exists $B_k \in \mathcal{K}_\infty$ such that the optimal value functions of $\mathcal{P}_k(x_0)$ satisfy

$$V_k(x) \leq B_k(\ell^*(x)) \quad \text{for all} \quad x \in \mathbb{X} \text{ and all } \quad k = 2, \ldots, N$$

Consider the optimization problem

$$\begin{aligned} \alpha := \quad & \inf_{\lambda_0, \ldots, \lambda_{N-1}, \nu} \frac{\sum_{n=0}^{N-1} \lambda_n - \nu}{\sum_{n=0}^{m-1} \lambda_n} \\ \text{s.t.} \quad & \sum_{n=k}^{N-1} \lambda_n \leq B_{N-k}(\lambda_k), \quad k = 0, \ldots, N-2 \\ & \nu \leq \sum_{n=0}^{j-1} \lambda_{n+m} + B_{N-j}(\lambda_{j+m}), \quad j = 0, .., N-m-1 \\ & \sum_{n=0}^{m-1} \lambda_n > 0, \; \lambda_0, \ldots, \lambda_{N-1}, \nu \geq 0 \end{aligned} \qquad \mathcal{P}_\alpha$$

Nominal Stability and Performance [Grüne, 2009]

Assume (A2) and that the optimization problem

$$
\begin{aligned}
\alpha := \quad & \inf_{\lambda_0,\ldots,\lambda_{N-1},\nu} \frac{\sum_{n=0}^{N-1} \lambda_n - \nu}{\sum_{n=0}^{m-1} \lambda_n} \\
\text{s.t.} \quad & \textstyle\sum_{n=k}^{N-1} \lambda_n \leq B_{N-k}(\lambda_k), \quad k = 0, \ldots, N-2 \qquad \mathcal{P}_\alpha \\
& \nu \leq \textstyle\sum_{n=0}^{j-1} \lambda_{n+m} + B_{N-j}(\lambda_{j+m}), \quad j = 0, .., N-m-1 \\
& \textstyle\sum_{n=0}^{m-1} \lambda_n > 0, \ \lambda_0, \ldots, \lambda_{N-1}, \nu \geq 0
\end{aligned}
$$

has an optimal value $\alpha \in (0, 1]$.

Then the m-step feedback $\mu_{N,m}$ & optimal value fcn V_N satisfy the relaxed dynamic programming inequality

$$V_N(x_0) \geq V_N(x_{\mu_{N,m}}(m)) + \alpha \sum_{k=0}^{m-1} \ell(x_{\mu_{N,m}}(k), \mu_{N,m}(x_{\mu_{N,m}}(\tilde{k}), k))$$

Nominal Stability and Performance [Grüne, 2009]

Assume (A2) and that the optimization problem

$$
\begin{aligned}
\alpha := \quad & \inf_{\lambda_0,\ldots,\lambda_{N-1},\nu} \frac{\sum_{n=0}^{N-1} \lambda_n - \nu}{\sum_{n=0}^{m-1} \lambda_n} \\
\text{s.t.} \quad & \sum_{n=k}^{N-1} \lambda_n \leq B_{N-k}(\lambda_k), \quad k = 0, \ldots, N-2 \qquad\qquad \mathcal{P}_\alpha \\
& \nu \leq \sum_{n=0}^{j-1} \lambda_{n+m} + B_{N-j}(\lambda_{j+m}), \quad j = 0, .., N-m-1 \\
& \sum_{n=0}^{m-1} \lambda_n > 0, \; \lambda_0, \ldots, \lambda_{N-1}, \nu \geq 0
\end{aligned}
$$

has an optimal value $\alpha \in (0, 1]$.

Then the ff. suboptimality estimate holds:

$$V_\infty(x) \leq J_\infty^{\text{cl}}(x, \mu_{N,m}) \leq V_N(x)/\alpha \leq V_\infty(x)/\alpha \qquad \forall x \in \mathbb{X}$$

If (A1) also holds, then the closed loop is asymptotically stable.

α is the index of suboptimality

Nominal Stability and Performance [Grüne et al., 2010]

Theorem

Let B_k, $k = 2, \ldots, N$, be linear functions.
Define $\gamma_k := B_k(r)/r$.
Then $\alpha = 1$ if and only if $\gamma_{m+1} \leq 1$. Otherwise,

$$\alpha = 1 - \frac{(\gamma_{m+1} - 1) \prod_{i=m+2}^{N} (\gamma_i - 1) \prod_{i=N-m+1}^{N} (\gamma_i - 1)}{\left(\prod_{i=m+1}^{N} \gamma_i - (\gamma_{m+1} - 1) \prod_{i=m+2}^{N} (\gamma_i - 1) \right) \left(\prod_{i=N-m+1}^{N} \gamma_i - \prod_{i=N-m+1}^{N} (\gamma_i - 1) \right)}$$

Nominal Stability and Performance

Course of reasoning:

- (A2) ($V_k(x) \leq B_k(\ell^*(x))$) allows for the formulation of $\mathcal{P}_\alpha$
- If $\mathcal{P}_\alpha$ has a solution $\alpha \in (0, 1]$

  $\Longrightarrow$ Relaxed Dynamic Programming Inequality (RDPI)

  $\Longrightarrow$ assumptions of (P1) are fulfilled

  $\Longrightarrow$ asymptotic stability and performance estimates

Robust Stability

Consider $A \subset \mathbb{X}$ where μ is feasible.

Robust Stability

Consider $A \subset \mathbb{X}$ where μ is feasible.

$x^* \in A$ is **robustly stable on** A if there exists $\beta \in \mathcal{KL}$ such that:

Robust Stability

Consider $A \subset \mathbb{X}$ where μ is feasible.

$x^* \in A$ is **robustly stable on** A if there exists $\beta \in \mathcal{KL}$ such that:

$\forall \delta > 0,\ \forall \Delta > \delta,\ \ \exists\ \bar{d} > 0$ s.t.

Robust Stability

Consider $A \subset \mathbb{X}$ where μ is feasible.

$x^* \in A$ is **robustly stable on** A if there exists $\beta \in \mathcal{KL}$ such that:

$\forall \delta > 0, \ \forall \Delta > \delta, \quad \exists \ \bar{d} > 0$ s.t.

any perturbed solution $\tilde{x}_\mu(\cdot, x_0)$ with $\|d(k)\| \leq \bar{d}$

Robust Stability

Consider $A \subset \mathbb{X}$ where μ is feasible.

$x^* \in A$ is **robustly stable on** A if there exists $\beta \in \mathcal{KL}$ such that:

$\forall \delta > 0,\ \forall \Delta > \delta,\ \ \exists\, \bar{d} > 0$ s.t.

any perturbed solution $\tilde{x}_\mu(\cdot, x_0)$ with $\|d(k)\| \leq \bar{d}$

with $x_0 \in A$ and $\|x_0\|_{x^*} \leq \Delta$ satisfies

Robust Stability

Consider $A \subset \mathbb{X}$ where μ is feasible.

$x^* \in A$ is **robustly stable on A** if there exists $\beta \in \mathcal{KL}$ such that:

$\forall \delta > 0,\ \forall \Delta > \delta,\ \exists\, \bar{d} > 0$ s.t.

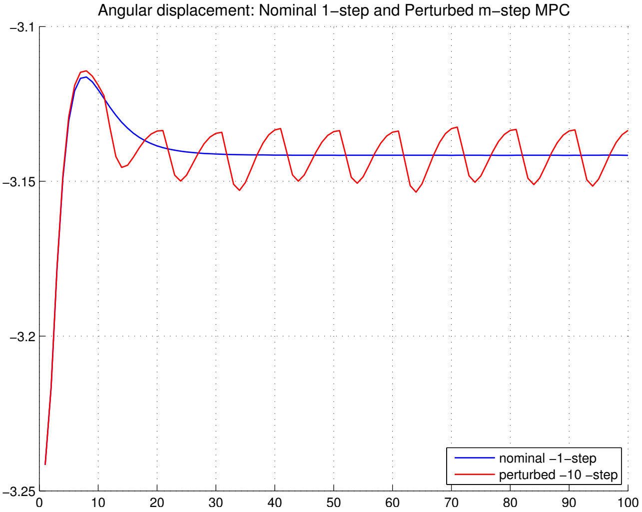
any perturbed solution $\tilde{x}_\mu(\cdot, x_0)$ with $\|d(k)\| \leq \bar{d}$

with $x_0 \in A$ and $\|x_0\|_{x^*} \leq \Delta$ satisfies

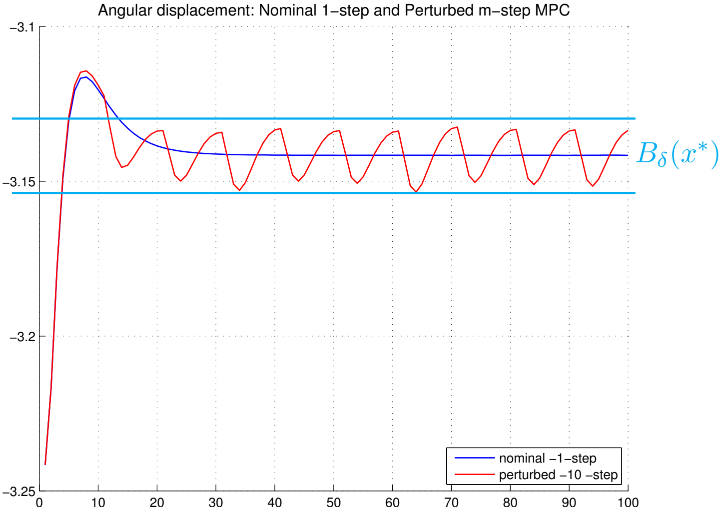
$$\tilde{x}_{\mu_N}(k, x_0) \in A \quad \text{and}$$

$$\|\tilde{x}_{\mu_N}(k, x_0)\|_{x^*} \leq \max\left\{\beta(\|x_0\|_{x^*}, k), \delta\right\} \quad \forall k \in \mathbb{N}_0.$$

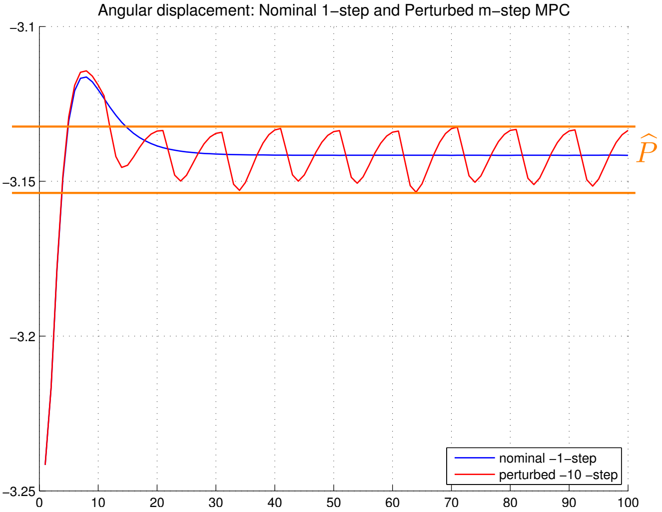
Robust Stability

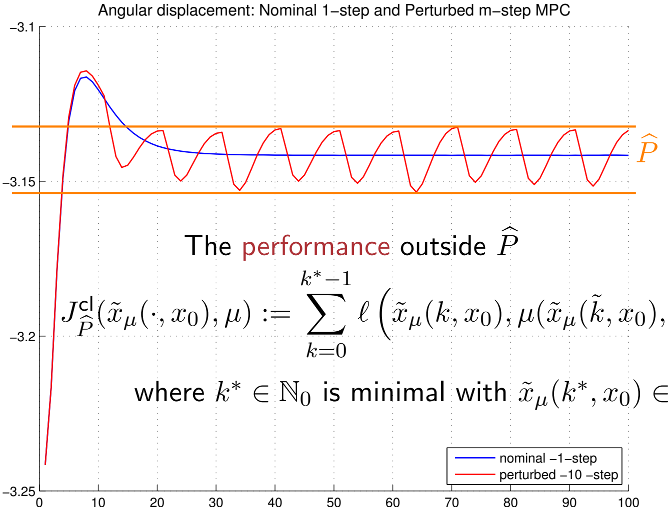


Robust Stability



Robust Stability





The performance outside $\widehat{P}$

$$J^{\mathsf{cl}}_{\widehat{P}}(\tilde{x}_\mu(\cdot, x_0), \mu) := \sum_{k=0}^{k^*-1} \ell\left(\tilde{x}_\mu(k, x_0), \mu(\tilde{x}_\mu(\tilde{k}, x_0), k)\right)$$

where $k^* \in \mathbb{N}_0$ is minimal with $\tilde{x}_\mu(k^*, x_0) \in \widehat{P}$

Counterpart of (P1)

Consider time-dependent $\mu : \mathbb{X} \times \mathbb{N} \to U$, function $V : X \to \mathbb{R}_0^+$ and sets $Y, P, \widehat{P}$ with appropriate invariance properties.

Assume $\exists\ \alpha \in (0,1]$ s.t. the RDPI

$$V(x_0) \geq V(\tilde{x}_\mu(m, x_0)) + \alpha \sum_{k=0}^{m-1} \ell(\tilde{x}_\mu(k, x_0), \mu(\tilde{x}_\mu(\tilde{k}, x_0), k))$$

holds $\forall x_0 \in Y \setminus P$ and $\forall$ perturbed soln $\tilde{x}_\mu(\cdot, x_0)$ with $\|d(k)\|$.

Counterpart of (P1)

Consider time-dependent $\mu : \mathbb{X} \times \mathbb{N} \to U$, function $V : X \to \mathbb{R}_0^+$ and sets $Y, P, \widehat{P}$ with appropriate invariance properties.

Assume $\exists\ \alpha \in (0, 1]$ s.t. the RDPI

$$V(x_0) \geq V(\tilde{x}_\mu(m, x_0)) + \alpha \sum_{k=0}^{m-1} \ell(\tilde{x}_\mu(k, x_0), \mu(\tilde{x}_\mu(\tilde{k}, x_0), k))$$

holds $\forall x_0 \in Y \setminus P$ and $\forall$ perturbed soln $\tilde{x}_\mu(\cdot, x_0)$ with $\|d(k)\|$.

Then $\forall x_0 \in Y \setminus \widehat{P}$ and $\forall \tilde{x}_\mu(k, x_0)$,

$$J^{\mathsf{cl}}_{\widehat{P}}(\tilde{x}_\mu(k, x_0), \mu) \leq V(x_0)/\alpha$$

Counterpart of (P1)

Consider time-dependent $\mu : \mathbb{X} \times \mathbb{N} \to U$, function $V : X \to \mathbb{R}_0^+$ and sets $Y, P, \widehat{P}$ with appropriate invariance properties.

Assume $\exists\ \alpha \in (0, 1]$ s.t. the RDPI

$$V(x_0) \geq V(\tilde{x}_\mu(m, x_0)) + \alpha \sum_{k=0}^{m-1} \ell(\tilde{x}_\mu(k, x_0), \mu(\tilde{x}_\mu(\tilde{k}, x_0), k))$$

holds $\forall x_0 \in Y \setminus P$ and $\forall$ perturbed soln $\tilde{x}_\mu(\cdot, x_0)$ with $\|d(k)\|$.

Then $\forall x_0 \in Y \setminus \widehat{P}$ and $\forall \tilde{x}_\mu(k, x_0)$,

$$J^{\mathsf{cl}}_{\widehat{P}}(\tilde{x}_\mu(k, x_0), \mu) \leq V(x_0)/\alpha$$

If (A1) holds and $\exists\ \alpha_3, \alpha_4 \in \mathcal{K}_\infty$ s.t.

$$\alpha_3(\|x\|_{x_*}) \leq V(x) \leq \alpha_4(\|x\|_{x_*}),$$

then μ renders the closed-loop system robustly stable.

Counterpart of (P1)

- Aim: using the counterpart of (P1) for perturbed system, show that $\mu_{N,m}$ and $\hat{\mu}_{N,m}$ renders the perturbed system robustly stable
- We need the corresponding α's

Uniform Continuity

Consider vector spaces Z, Y, set $A \subset Z$ and arbitrary set W

- a function $\phi : Z \to Y$ is uniformly continuous on A if $\exists\ \omega \in \mathcal{K}$ s.t. $\forall\ z_1, z_2 \in A$

$$\|\phi(z_1) - \phi(z_2)\| \leq \omega\left(\|z_1 - z_2\|\right).$$

Uniform Continuity

Consider vector spaces Z, Y, set $A \subset Z$ and arbitrary set W

- a function $\phi : Z \to Y$ is uniformly continuous on A if $\exists\ \omega \in \mathcal{K}$ s.t. $\forall\ z_1, z_2 \in A$

$$\|\phi(z_1) - \phi(z_2)\| \leq \omega\left(\|z_1 - z_2\|\right).$$

- a function $\phi : Z \times W \to Y$ is uniformly continuous on A uniformly in $v \in W$ if $\exists\ \omega \in \mathcal{K}$ s.t. $\forall\ z_1, z_2 \in A$ and $\forall\ v \in W$

$$\|\phi(z_1, v) - \phi(z_2, v)\| \leq \omega\left(\|z_1 - z_2\|\right).$$

Uniform Continuity

Consider vector spaces Z, Y, set $A \subset Z$ and arbitrary set W

- a function $\phi : Z \to Y$ is uniformly continuous on A if $\exists\, \omega \in \mathcal{K}$ s.t. $\forall\, z_1, z_2 \in A$

$$\|\phi(z_1) - \phi(z_2)\| \leq \omega\left(\|z_1 - z_2\|\right).$$

- a function $\phi : Z \times W \to Y$ is uniformly continuous on A uniformly in $v \in W$ if $\exists\, \omega \in \mathcal{K}$ s.t. $\forall\, z_1, z_2 \in A$ and $\forall\, v \in W$

$$\|\phi(z_1, v) - \phi(z_2, v)\| \leq \omega\left(\|z_1 - z_2\|\right).$$

The function ω is called the modulus of continuity.

Uniform Continuity of $V_N(\cdot)$

$$\|J_N(x_1, u(\cdot)) - J_N(x_2, u(\cdot))\| \leq \omega_{J_N}(\|x_1 - x_2\|)$$

uniform continuity of J_N uniformly on $u(\cdot)$

Uniform Continuity of $V_N(\cdot)$

$$\|J_N(x_1, u(\cdot)) - J_N(x_2, u(\cdot))\| \leq \omega_{J_N}(\|x_1 - x_2\|)$$

uniform continuity of J_N uniformly on $u(\cdot)$

$$\Downarrow$$

$$\|V_N(x_1) - V_N(x_2)\| \leq \omega_{V_N}(\|x_1 - x_2\|)$$

uniform continuity of V_N

Uniform Continuity of $V_N(\cdot)$

$$\|J_N(x_1, u(\cdot)) - J_N(x_2, u(\cdot))\| \leq \omega_{J_N}(\|x_1 - x_2\|)$$

uniform continuity of J_N uniformly on $u(\cdot)$

$$\Downarrow$$

$$\|V_N(x_1) - V_N(x_2)\| \leq \omega_{V_N}(\|x_1 - x_2\|)$$

uniform continuity of V_N

- $\omega_{V_N} \leq \omega_{J_N}$
- $\omega_{V_N} \ll \omega_{J_N}$ for open-loop unstable and controllable

Obtaining analogous statements

Recall $\mathcal{P}_\alpha$

$$\alpha := \inf_{\lambda_0,\ldots,\lambda_{N-1},\nu} \frac{\sum_{n=0}^{N-1}\lambda_n - \nu}{\sum_{n=0}^{m-1}\lambda_n}$$

s.t. $$\sum_{n=k}^{N-1}\lambda_n \le B_{N-k}(\lambda_k) \qquad k = 0,\ldots,N-2$$

$$\nu \le \sum_{n=0}^{j-1}\lambda_{n+m} + B_{N-j}(\lambda_{j+m}) \qquad j = 0,..,N-m-1$$

$$\sum_{n=0}^{m-1}\lambda_n > 0,\ \lambda_0,\ldots,\lambda_{N-1},\nu \ge 0$$

Obtaining analogous statements

Define $\mathcal{P}_\alpha^{\mathsf{pmult}}$

$$\alpha^{\mathsf{pmult}} := \inf_{\lambda_0,\ldots,\lambda_{N-1},\nu} \frac{\sum_{n=0}^{N-1}\lambda_n - \nu}{\sum_{n=0}^{m-1}\lambda_n}$$

$$\text{s.t.} \quad \sum_{n=k}^{N-1}\lambda_n \leq B_{N-k}(\lambda_k) + \xi^{\mathsf{pmult}}, \qquad k = 0,\ldots,N-2$$

$$\nu \leq \sum_{n=0}^{j-1}\lambda_{n+m} + B_{N-j}(\lambda_{j+m}) + \xi^{\mathsf{pmult}}, \quad j = 0,..,N-m-1$$

$$\sum_{n=0}^{m-1}\lambda_n > \zeta,\ \lambda_0,\ldots,\lambda_{N-1},\nu \geq 0$$

where $\xi_k^{\mathsf{pmult}} = \sum_{j=1}^{N-k-1}\omega_{J_{N-k-j}}(\|d(k+j)\|) + \omega_{B_{N-k}}(\lambda_{k,k,0} - \lambda_{k,0,0}) + \omega_{J_{N-k}}(x_{k,k,0} - x_{k,0,0})$

Obtaining analogous statements

Define $\mathcal{P}_\alpha^{\mathsf{upd}}$

$$\alpha^{\mathsf{upd}} := \inf_{\lambda_0,\ldots,\lambda_{N-1},\nu} \frac{\sum_{n=0}^{N-1}\lambda_n - \nu}{\sum_{n=0}^{m-1}\lambda_n}$$

s.t.

$$\sum_{n=k}^{N-1}\lambda_n \leq B_{N-k}(\lambda_k) + \xi^{\mathsf{upd}}, \qquad k = 0,\ldots,N-2$$

$$\nu \leq \sum_{n=0}^{j-1}\lambda_{n+m} + B_{N-j}(\lambda_{j+m}) + \xi^{\mathsf{upd}}, \quad j = 0,..,N-m-1$$

$$\sum_{n=0}^{m-1}\lambda_n > \zeta,\ \lambda_0,\ldots,\lambda_{N-1},\nu \geq 0$$

where $\xi_k^{\mathsf{upd}} = \sum_{j=1}^{N-k-1}\omega_{V_{N-k-j}}(\|d(k+j)\|)$

Comparing $\alpha, \alpha^{\text{pmult}}, \alpha^{\text{upd}}$

- $\mathcal{P}_\alpha^{\text{pmult}}$ and $\mathcal{P}_\alpha^{\text{upd}}$ are 'perturbed' versions of $\mathcal{P}_\alpha$

Comparing $\alpha, \alpha^{\text{pmult}}, \alpha^{\text{upd}}$

- $\mathcal{P}_\alpha^{\text{pmult}}$ and $\mathcal{P}_\alpha^{\text{upd}}$ are 'perturbed' versions of $\mathcal{P}_\alpha$

Theorem

Consider problems $\mathcal{P}_\alpha$, $\mathcal{P}_\alpha^{pmult}$ and $\mathcal{P}_\alpha^{upd}$.
Assume that the B_k, $k \in \mathbb{N}$ from (A2) are linear functions.
Then

$$\alpha^{pmult} \geq \alpha - \frac{B_{m+1}(\xi^{pmult}) + \xi^{pmult}}{\zeta}$$

$$\alpha^{upd} \geq \alpha - \frac{B_{m+1}(\xi^{upd}) + \xi^{upd}}{\zeta}$$

Comparing $\alpha, \alpha^{\text{pmult}}, \alpha^{\text{upd}}$

- $\mathcal{P}_\alpha^{\text{pmult}}$ and $\mathcal{P}_\alpha^{\text{upd}}$ are 'perturbed' versions of $\mathcal{P}_\alpha$

Theorem

Consider problems $\mathcal{P}_\alpha$, $\mathcal{P}_\alpha^{pmult}$ and $\mathcal{P}_\alpha^{upd}$.
Assume that the B_k, $k \in \mathbb{N}$ from (A2) are linear functions.
Then

$$\alpha^{pmult} \geq \alpha - \frac{B_{m+1}(\xi^{pmult}) + \xi^{pmult}}{\zeta}$$

$$\alpha^{upd} \geq \alpha - \frac{B_{m+1}(\xi^{upd}) + \xi^{upd}}{\zeta}$$

- lower bounds for α^{pmult} and α^{upd} in terms of α

Robust Stability rendered by $\mu_{N,m}$ and $\hat{\mu}_{N,m}$

Assume some technical assumptions.
Let α^{pmult} be the solution of $\mathcal{P}_\alpha^{\mathsf{pmult}}$ for $d(\cdot)$ and some $\zeta > 0$.

Then the RDPI

$$V_N(x_0) \geq V_N(x_{\mu_{N,m}}(m, x_0)) + \tilde{\alpha}^{\mathsf{pmult}} \sum_{k=0}^{m-1} \ell(\tilde{x}_{\mu_{N,m}}(k, x_0), \mu_{N,m}(x_0, k))$$

holds for

$$\tilde{\alpha}^{\mathsf{pmult}} = \alpha^{\mathsf{pmult}} - \frac{\sigma}{\zeta} \quad \text{where } \sigma = \sum_{j=1}^{m-1} \omega_{J_{N-j}}(\|d(j)\|)$$

Robust Stability rendered by $\mu_{N,m}$ and $\hat{\mu}_{N,m}$

Assume some technical assumptions.
Let α^{upd} be the solution of $\mathcal{P}_\alpha^{\mathsf{upd}}$ for $d(\cdot)$ and some $\zeta > 0$.

Then the RDPI

$$V_N(x_0) \geq V_N(x_{\hat{\mu}_{N,m}}(m, x_0)) + \tilde{\alpha}^{\mathsf{upd}} \sum_{k=0}^{m-1} \ell(\tilde{x}_{\hat{\mu}_{N,m}}(k, x_0), \hat{\mu}_{N,m}(x_0, k))$$

holds for

$$\tilde{\alpha}^{\mathsf{upd}} = \alpha^{\mathsf{upd}} - \frac{\sigma}{\zeta} \quad \text{where } \sigma = \sum_{j=1}^{m-1} \omega_{V_{N-j}}(\|d(j)\|)$$

Robust Stability rendered by $\mu_{N,m}$ and $\hat{\mu}_{N,m}$

- the decisive difference bet. the 2 cases: error terms
- for $\mu_{N,m}$, dependence on ω_{J_k}
  for $\hat{\mu}_{N,m}$, dependence on ω_{V_k}
- these error terms determine the bound for $\overline{d}$
  and the suboptimality index α

Numerical Example

Table: Comparison of time requirements in CPU time

m	multistep	updated
1	11.0447	11.0967
2	5.6484	10.4687
3	3.6762	10.3646
4	2.5522	10.1046
5	2.1921	9.3766
6	1.8241	8.6125
7	1.5801	7.7765
8	1.2321	7.7845
9	1.0881	7.2405
10	1.0641	6.5404
11	0.9521	6.1124
12	0.8601	5.7884
13	0.8681	5.2243

Numerical Example

Table: Suboptimality index α of the schemes for various m and iterations

m	nominal multistep	perturbed multistep	updated multistep
1	0.9908	0.8667	0.8667
2	0.9911	0.8678	0.8681
3	0.9915	0.7936	0.7955
4	0.9917	0.7672	0.7729
5	0.9916	0.7632	0.7734
6	0.9913	0.7724	0.7868
7	0.9908	0.7404	0.7629
8	0.9902	0.7103	0.7414
9	0.9895	0.7066	0.7423
10	0.9888	0.6988	0.7379
11	0.9883	0.6477	0.6953
12	0.9880	0.6183	0.6688
13	0.9879	0.6133	0.6609

Summary and Outlook

- statements analogous to those of the nominal MPC closed-loop are obtained for the perturbed closed-loop setting

Summary and Outlook

- statements analogous to those of the nominal MPC closed-loop are obtained for the perturbed closed-loop setting
- a RDPI holds for feedbacks $\mu_{N,m}$ and $\hat{\mu}_{N,m}$ along perturbed trajectories

Summary and Outlook

- statements analogous to those of the nominal MPC closed-loop are obtained for the perturbed closed-loop setting
- a RDPI holds for feedbacks $\mu_{N,m}$ and $\hat{\mu}_{N,m}$ along perturbed trajectories $\Longrightarrow$ robust stability and performance

Summary and Outlook

- statements analogous to those of the nominal MPC closed-loop are obtained for the perturbed closed-loop setting
- a RDPI holds for feedbacks $\mu_{N,m}$ and $\hat{\mu}_{N,m}$ along perturbed trajectories $\Longrightarrow$ robust stability and performance
- enhanced robustness induced by the updates in $\hat{\mu}_{N,m}$ manifest via the moduli of continuity ω_{V_k}, which are often considerably smaller than their counterparts ω_{J_k}

Summary and Outlook

- statements analogous to those of the nominal MPC closed-loop are obtained for the perturbed closed-loop setting
- a RDPI holds for feedbacks $\mu_{N,m}$ and $\hat{\mu}_{N,m}$ along perturbed trajectories $\Longrightarrow$ robust stability and performance
- enhanced robustness induced by the updates in $\hat{\mu}_{N,m}$ manifest via the moduli of continuity ω_{V_k}, which are often considerably smaller than their counterparts ω_{J_k}

Where do we use the results?

- to be applied to a sensitivity-based m-step MPC where the sensitivity-based updates can be viewed as a less costly approximation of the updates of the updated m-step MPC

Summary and Outlook

- statements analogous to those of the nominal MPC closed-loop are obtained for the perturbed closed-loop setting
- a RDPI holds for feedbacks $\mu_{N,m}$ and $\hat{\mu}_{N,m}$ along perturbed trajectories $\Longrightarrow$ robust stability and performance
- enhanced robustness induced by the updates in $\hat{\mu}_{N,m}$ manifest via the moduli of continuity ω_{V_k}, which are often considerably smaller than their counterparts ω_{J_k}

Where do we use the results?

- to be applied to a sensitivity-based m-step MPC where the sensitivity-based updates can be viewed as a less costly approximation of the updates of the updated m-step MPC

Thank you for your attention! :)