Fast MPC: Fundamentals for breaking trough the speed barrier

Fast NMPC: A Reality-Steered Paradigm

Mazen Alamir

CNRS, University of Grenoble, France.

Overview of the tutorial session

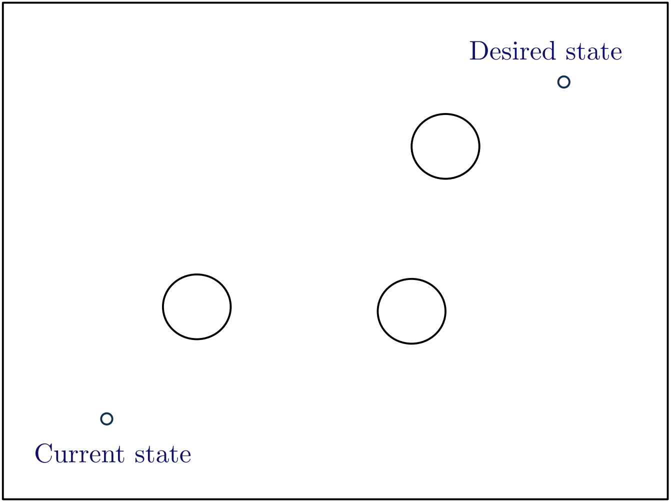
Fast MPC, a reality-steered paradigm	(M. Alamir)
Nonlinear MPC algorithms	(M. Diehl)
Exact penalty methods for fast MPC	(V. Zavala)
Splitting methods for embedded optimization and control	(C. Jones)
Co-Design of hardware and algorithms for embedded optimization	(E. Kerrigan)

Overview of this talk

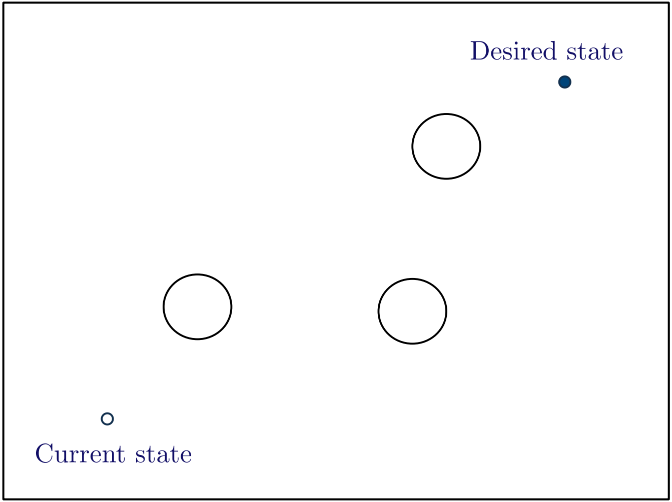
Agenda

1. Ideal NMPC
   - Definition & Stability
2. Fast NMPC
3. Advanced Topics

Ideal NMPC: Informal Definition

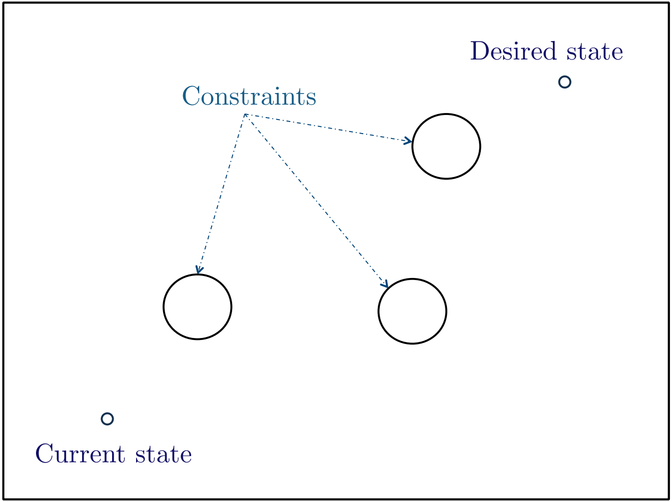


Ideal NMPC: Informal Definition



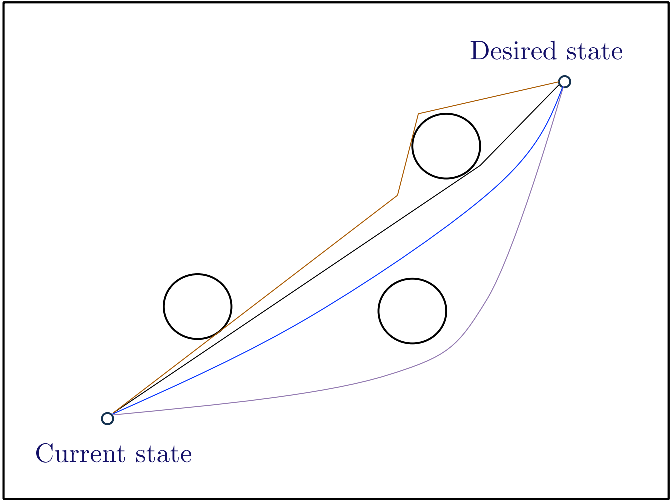
Ideal NMPC: Informal Definition

Constraints /



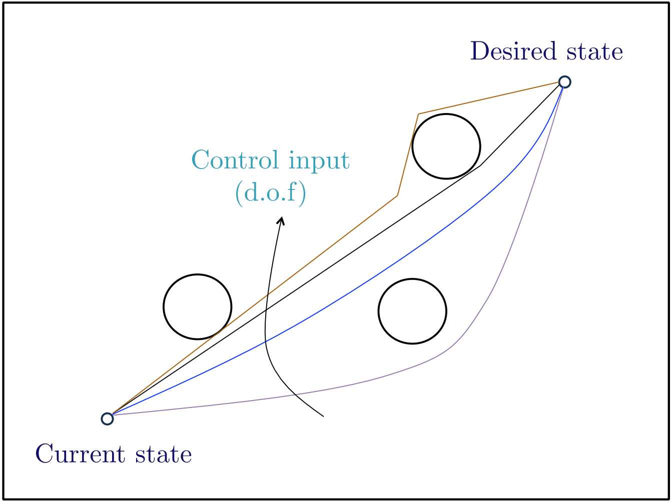
Ideal NMPC: Informal Definition

Constraints / Cost /



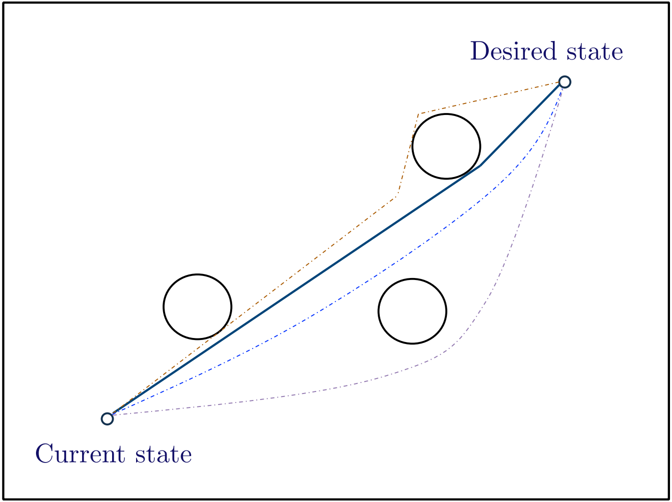
Ideal NMPC: Informal Definition

Constraints / Cost / Control /



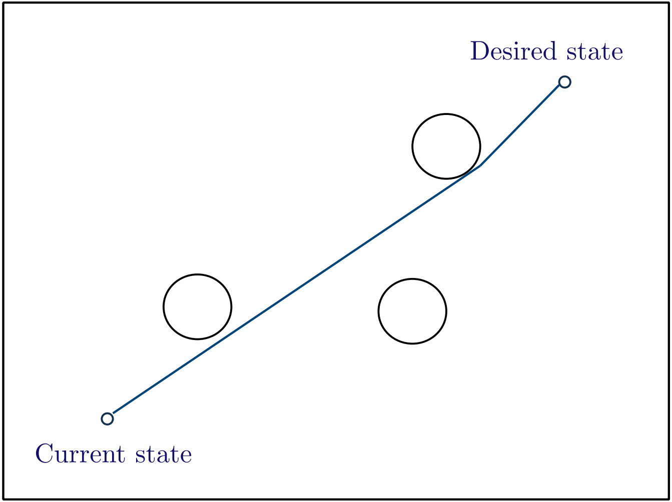
Ideal NMPC: Informal Definition

Constraints / Cost / Control /



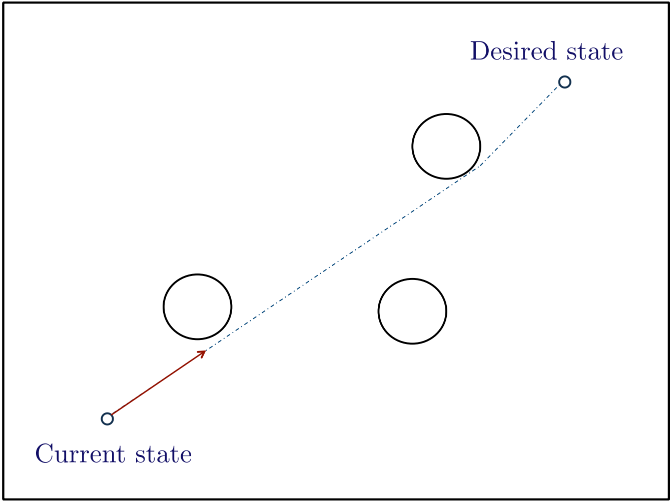
Ideal NMPC: Informal Definition

Constraints / Cost / Control /



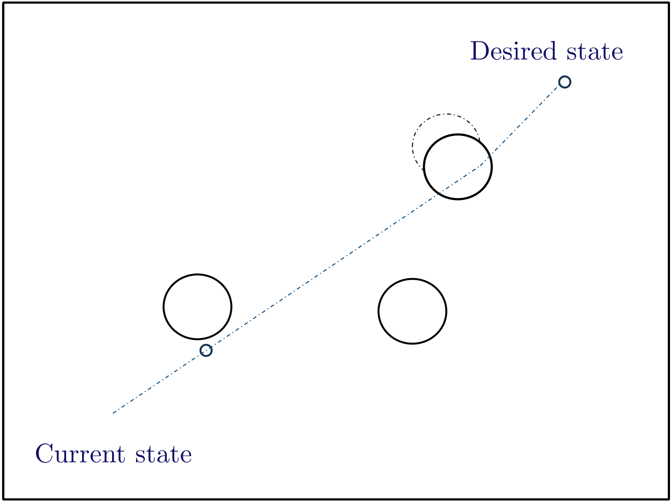
Ideal NMPC: Informal Definition

Constraints / Cost / Control /



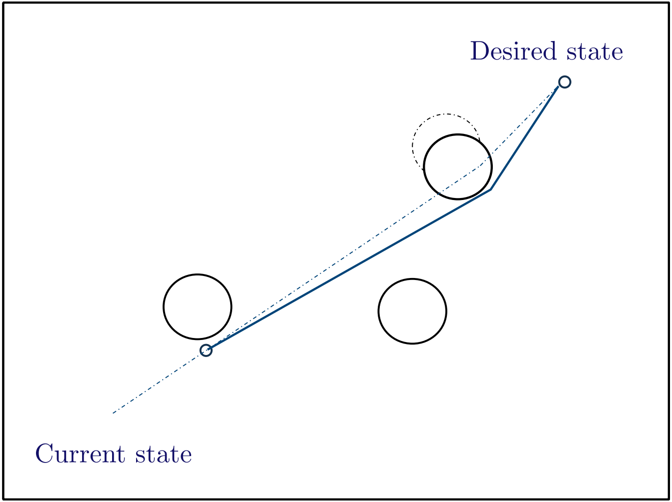
Ideal NMPC: Informal Definition

Constraints / Cost / Control / Disturbances



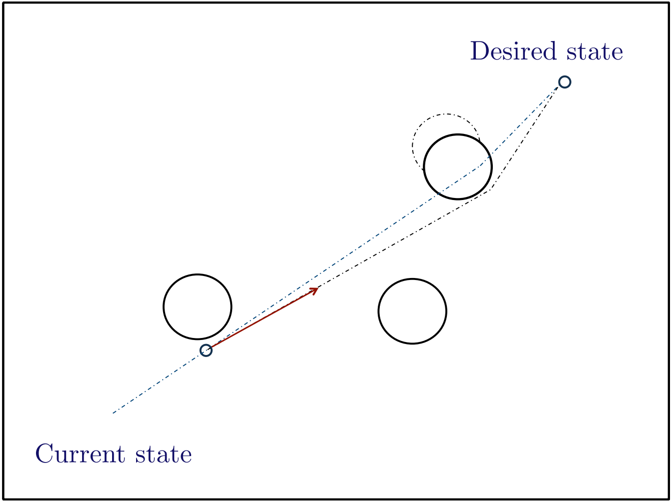
Ideal NMPC: Informal Definition

Constraints / Cost / Control / Disturbances



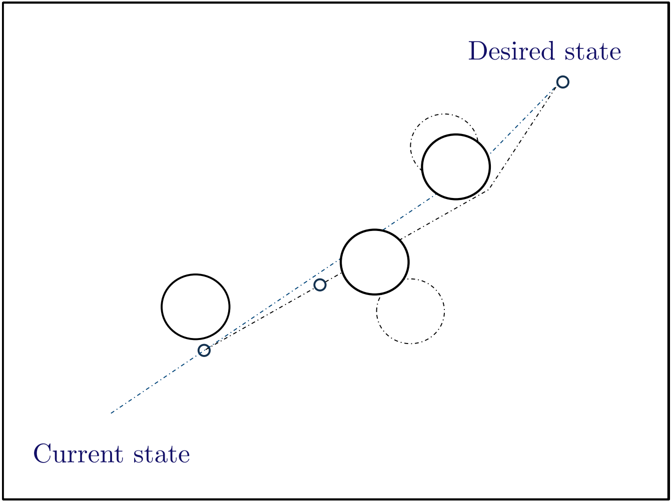
Ideal NMPC: Informal Definition

Constraints / Cost / Control / Disturbances



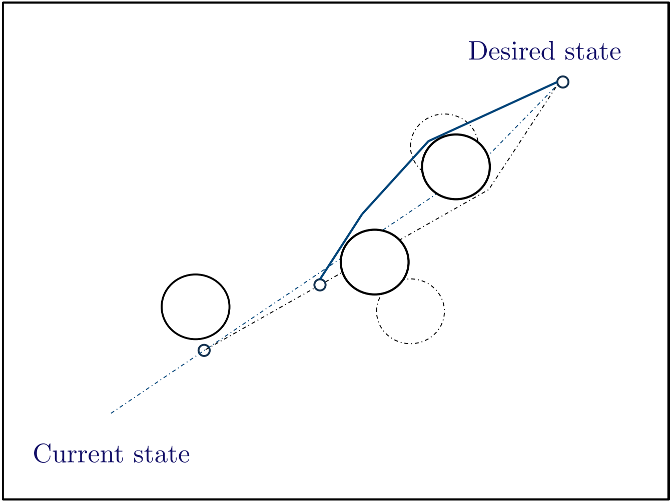
Ideal NMPC: Informal Definition

Constraints / Cost / Control / Disturbances



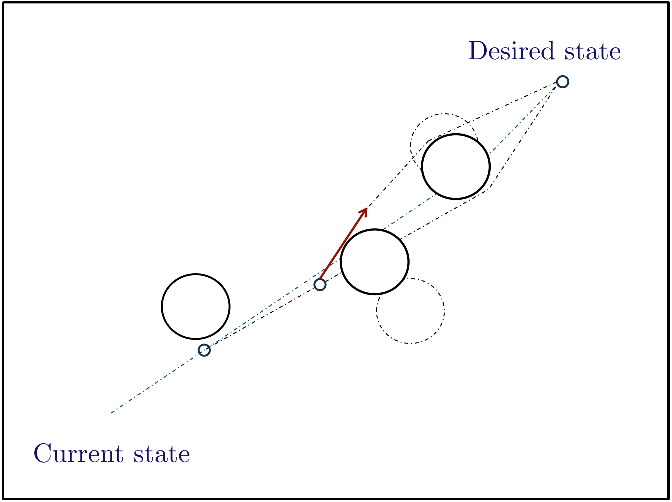
Ideal NMPC: Informal Definition

Constraints / Cost / Control / Disturbances



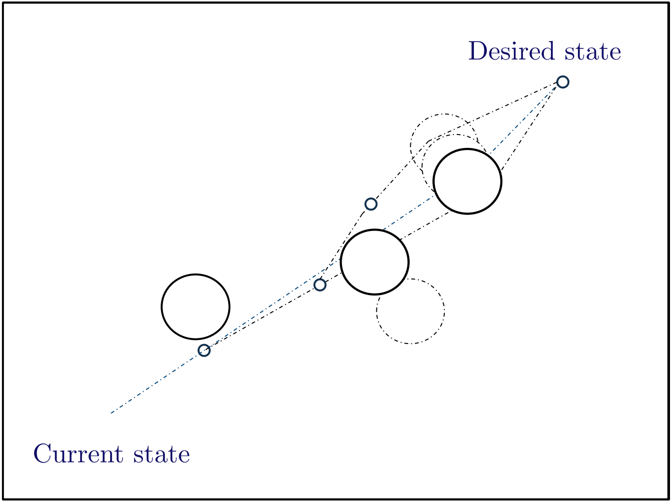
Ideal NMPC: Informal Definition

Constraints / Cost / Control / Disturbances



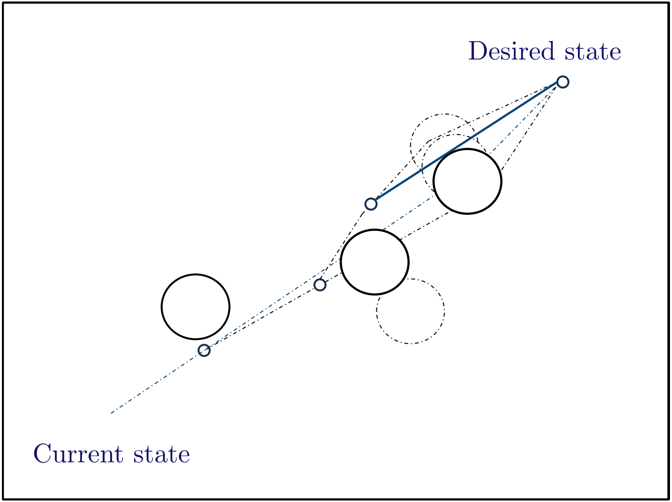
Ideal NMPC: Informal Definition

Constraints / Cost / Control / Disturbances



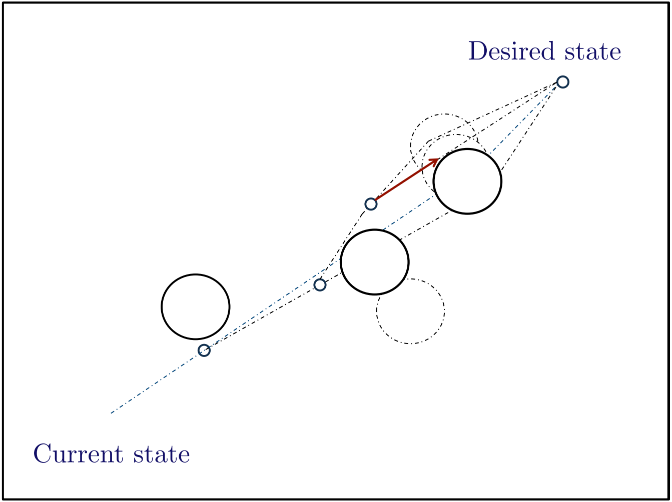
Ideal NMPC: Informal Definition

Constraints / Cost / Control / Disturbances



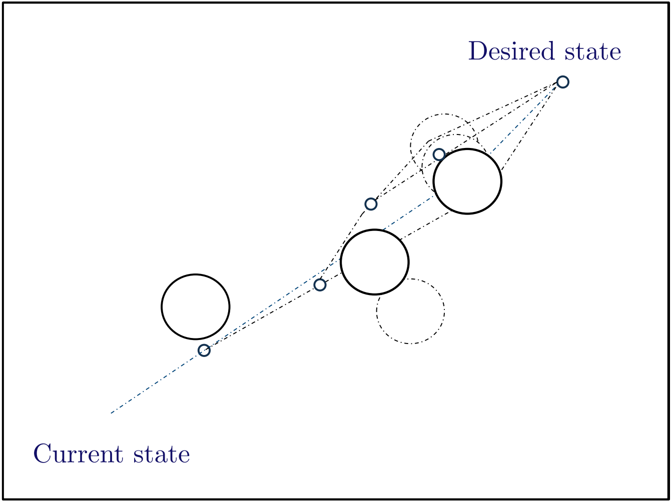
Ideal NMPC: Informal Definition

Constraints / Cost / Control / Disturbances



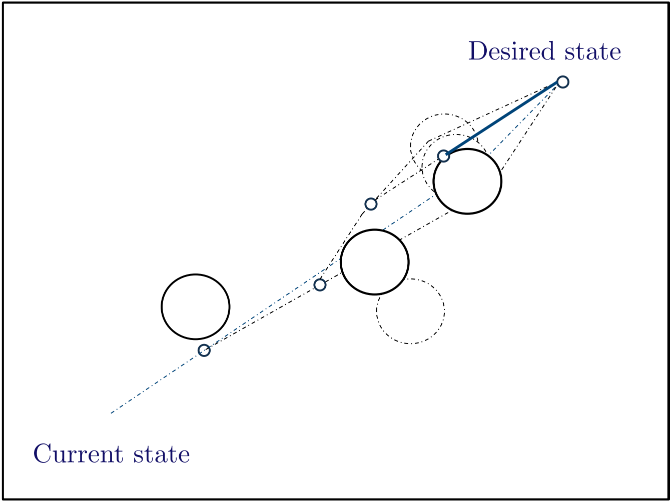
Ideal NMPC: Informal Definition

Constraints / Cost / Control / Disturbances



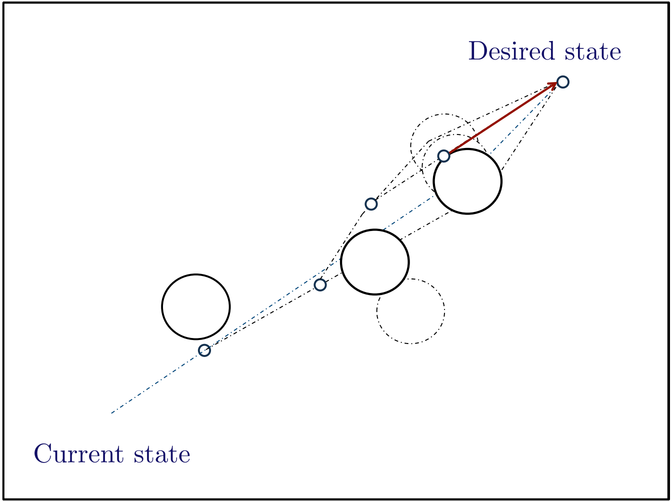
Ideal NMPC: Informal Definition

Constraints / Cost / Control / Disturbances



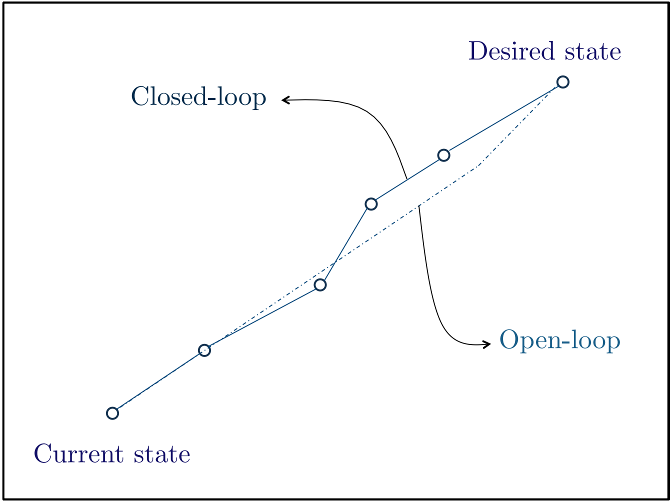
Ideal NMPC: Informal Definition

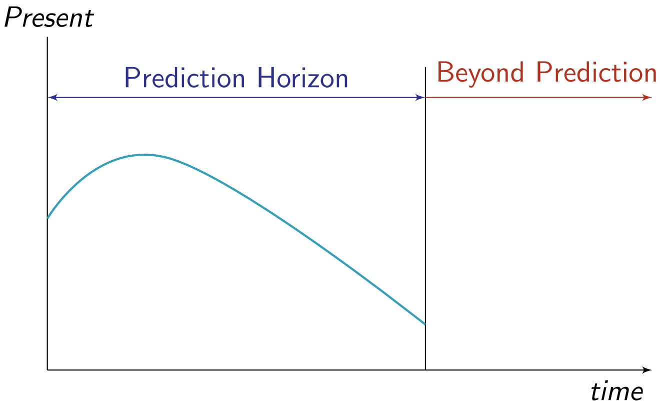
Constraints / Cost / Control / Disturbances

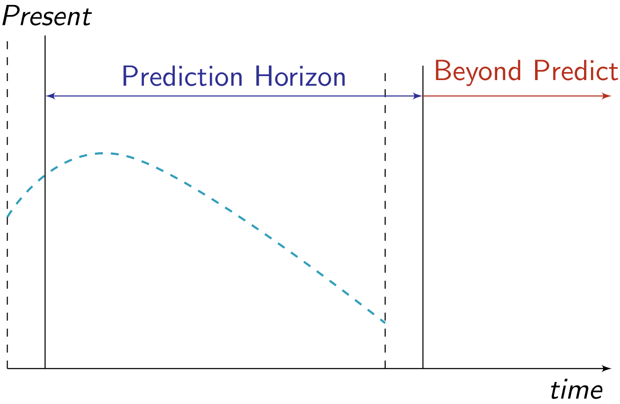


Ideal NMPC: Informal Definition

Constraints / Cost / Control / Disturbances




Present
Prediction Horizon
Beyond Prediction
time


Present
Prediction Horizon
Beyond Prediction
time

Ideal NMPC: Instability may arise

In general

Closed-loop trajectory $\neq$ Initial open-loop trajectory

and

Stability is not guaranteed for arbitrary choices of finite-horizon MPC ingredients

Ideal NMPC: Instability may arise

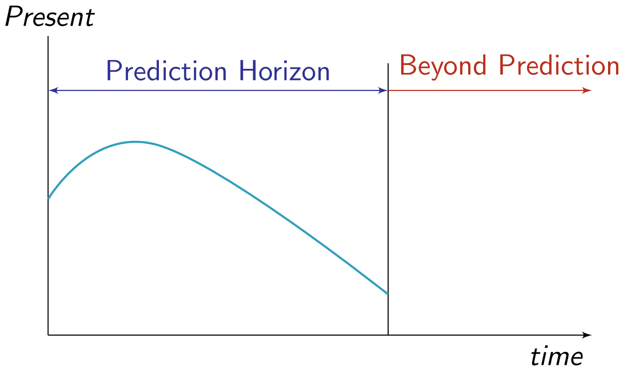
In general

Closed-loop trajectory $\neq$ Initial open-loop trajectory

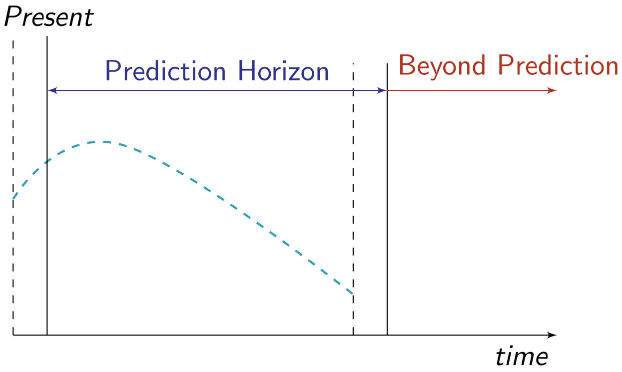
and

Stability is not guaranteed for arbitrary choices of finite-horizon MPC ingredients even with a perfect undisturbed model.

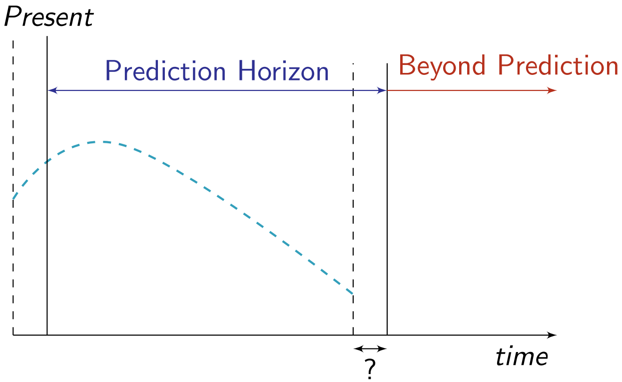
Ideal NMPC: Genesis of instability



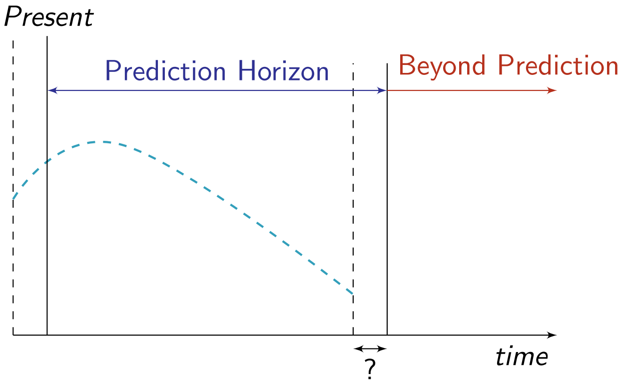
Ideal NMPC: Genesis of instability



Ideal NMPC: Genesis of instability



Ideal NMPC: Genesis of instability



Stability assessment is based on the characterization of the terminal region that must be reached at the end of the finite prediction horizon.

[Mayne et al. Automatica 2000]

Ideal NMPC: Formal Definition

At each decision instant k

Solve a NLP problem,

$$p^*(\mathbf{x}(k)) \quad \leftarrow \min_{p} J\big(p \mid \mathbf{x}(k)\big)$$

$$\text{under} \quad C\big(p \mid \mathbf{x}(k)\big) \leq 0$$

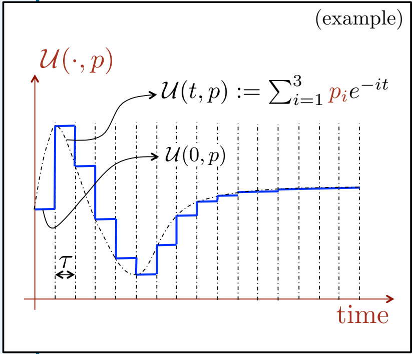
Apply the first control

$$\mathbf{u}(k) = \mathcal{U}\big(0, p^*(\mathbf{x}(k))\big)$$

over the updating period $[k, k+1]$.

Optimal cost

$$J^*(\mathbf{x}(k)) := J(p^*(\mathbf{x}(k)) \mid \mathbf{x}(k))$$



Ideal NMPC: Formal Definition

At each decision instant k

Solve a NLP problem,

$$p^*(\mathbf{x}(k)) \quad \leftarrow \min_p J(p \mid \mathbf{x}(k))$$

$$\text{under} \quad C(p \mid \mathbf{x}(k)) \le 0$$

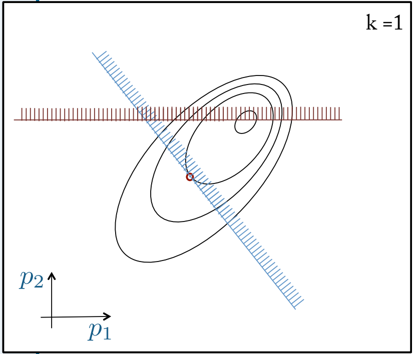
Apply the first control

$$\mathbf{u}(k) = \mathcal{U}(0, p^*(\mathbf{x}(k)))$$

over the updating period $[k, k+1]$.

Optimal cost

$$J^*(\mathbf{x}(k)) := J(p^*(\mathbf{x}(k)) \mid \mathbf{x}(k))$$



Ideal NMPC: Formal Definition

At each decision instant k

Solve a NLP problem,

$$p^*(\mathbf{x}(k)) \quad \leftarrow \min_p J(p \mid \mathbf{x}(k))$$
$$\text{under} \quad C(p \mid \mathbf{x}(k)) \leq 0$$

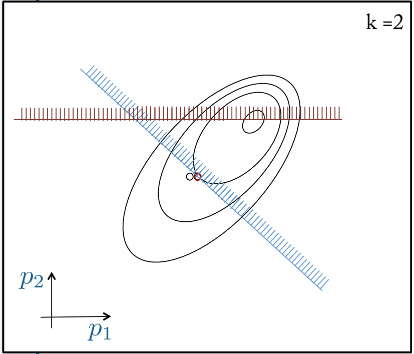
Apply the first control

$$\mathbf{u}(k) = \mathcal{U}(0, p^*(\mathbf{x}(k)))$$

over the updating period $[k, k+1]$.

Optimal cost

$$J^*(\mathbf{x}(k)) := J(p^*(\mathbf{x}(k)) \mid \mathbf{x}(k))$$



Ideal NMPC: Formal Definition

At each decision instant k

Solve a NLP problem,

$$p^*(\mathbf{x}(k)) \quad \leftarrow \min_p J(p \mid \mathbf{x}(k))$$
$$\text{under} \quad C(p \mid \mathbf{x}(k)) \leq 0$$

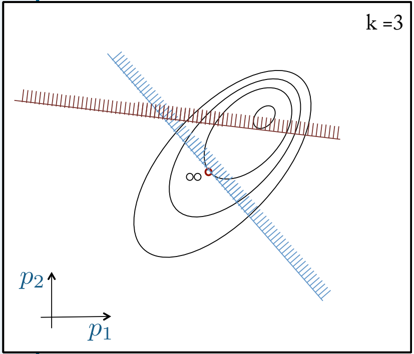
Apply the first control

$$\mathbf{u}(k) = \mathcal{U}(0, p^*(\mathbf{x}(k)))$$

over the updating period $[k, k+1]$.

Optimal cost

$$J^*(\mathbf{x}(k)) := J(p^*(\mathbf{x}(k)) \mid \mathbf{x}(k))$$



Ideal NMPC: Formal Definition

At each decision instant k

Solve a NLP problem,

$$p^*(\mathbf{x}(k)) \quad \leftarrow \min_p J(p \mid \mathbf{x}(k))$$
$$\text{under} \quad C(p \mid \mathbf{x}(k)) \leq 0$$

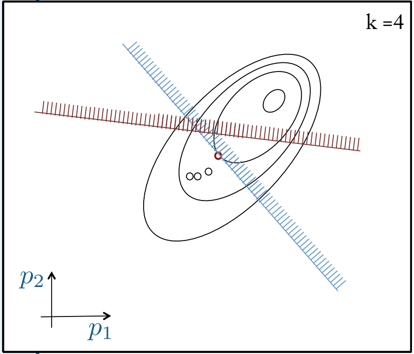
Apply the first control

$$\mathbf{u}(k) = \mathcal{U}(0, p^*(\mathbf{x}(k)))$$

over the updating period $[k, k+1]$.

Optimal cost

$$J^*(\mathbf{x}(k)) := J(p^*(\mathbf{x}(k)) \mid \mathbf{x}(k))$$



Ideal NMPC: Formal Definition

At each decision instant k

Solve a NLP problem,

$$p^*(\mathbf{x}(k)) \quad \leftarrow \min_p J(p \mid \mathbf{x}(k))$$
$$\text{under} \quad C(p \mid \mathbf{x}(k)) \leq 0$$

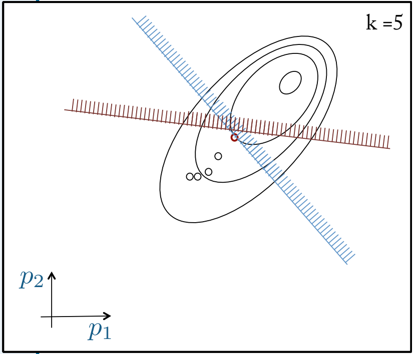
Apply the first control

$$\mathbf{u}(k) = \mathcal{U}(0, p^*(\mathbf{x}(k)))$$

over the updating period $[k, k+1]$.

Optimal cost

$$J^*(\mathbf{x}(k)) := J(p^*(\mathbf{x}(k)) \mid \mathbf{x}(k))$$



Ideal NMPC: Formal Definition

At each decision instant k

Solve a NLP problem,

$$p^*(\mathbf{x}(k)) \quad \leftarrow \min_p J(p \mid \mathbf{x}(k))$$
$$\text{under} \quad C(p \mid \mathbf{x}(k)) \leq 0$$

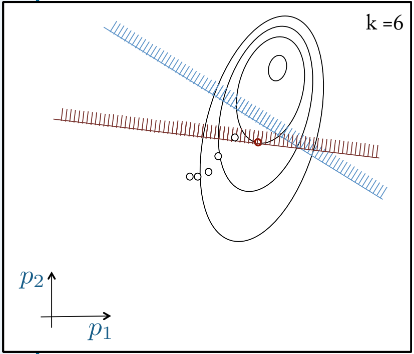
Apply the first control

$$\mathbf{u}(k) = \mathcal{U}(0, p^*(\mathbf{x}(k)))$$

over the updating period $[k, k+1]$.

Optimal cost

$$J^*(\mathbf{x}(k)) := J(p^*(\mathbf{x}(k)) \mid \mathbf{x}(k))$$



Ideal NMPC: Formal Definition

At each decision instant k

Solve a NLP problem,

$$p^*(\mathbf{x}(k)) \quad \leftarrow \min_p J(p \mid \mathbf{x}(k))$$
$$\text{under} \quad C(p \mid \mathbf{x}(k)) \leq 0$$

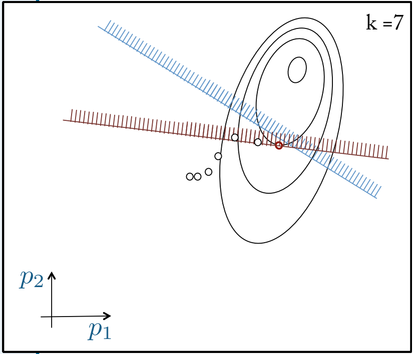
Apply the first control

$$\mathbf{u}(k) = \mathcal{U}(0, p^*(\mathbf{x}(k)))$$

over the updating period $[k, k+1]$.

Optimal cost

$$J^*(\mathbf{x}(k)) := J(p^*(\mathbf{x}(k)) \mid \mathbf{x}(k))$$



Ideal NMPC: Formal Definition

At each decision instant k

Solve a NLP problem,

$$p^*(\mathbf{x}(k)) \quad \leftarrow \min_p J(p \mid \mathbf{x}(k))$$
$$\text{under} \quad C(p \mid \mathbf{x}(k)) \leq 0$$

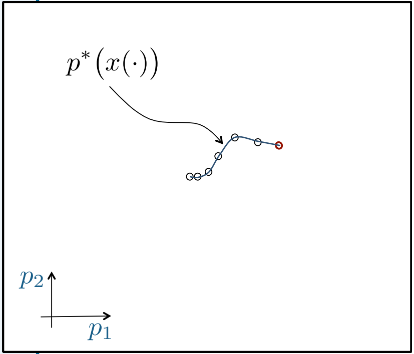
Apply the first control

$$\mathbf{u}(k) = \mathcal{U}(0, p^*(\mathbf{x}(k)))$$

over the updating period $[k, k+1]$.

Optimal cost

$$J^*(\mathbf{x}(k)) := J(p^*(\mathbf{x}(k)) \mid \mathbf{x}(k))$$



Ideal NMPC: Formal Definition

At each decision instant k

Solve a NLP problem,

$$p^*(\mathbf{x}(k)) \quad \leftarrow \min_p J(p \mid \mathbf{x}(k))$$
$$\text{under} \quad C(p \mid \mathbf{x}(k)) \le 0$$

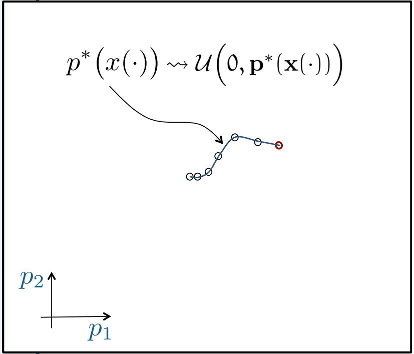
Apply the first control

$$\mathbf{u}(k) = \mathcal{U}\big(0, p^*(\mathbf{x}(k))\big)$$

over the updating period $[k, k+1]$.

Optimal cost

$$J^*(\mathbf{x}(k)) := J(p^*(\mathbf{x}(k)) \mid \mathbf{x}(k))$$



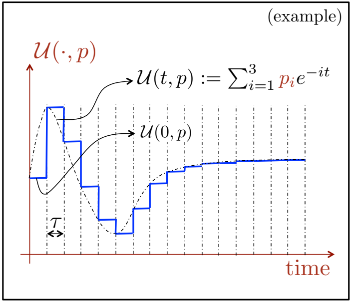
Ideal MPC: Common Control Parametrization

Trivial $\rightarrow p(k) := \tilde{\mathbf{u}}(k) := (\mathbf{u}(k), \dots, \mathbf{u}(k+N-1))^T \in \mathbb{R}^{Nn_u}$

Ideal MPC: Common Control Parametrization

Reduced $\rightarrow \tilde{\mathbf{u}}(k) = \mathcal{U}(k, p(k)),$ $[n_p < Nn_u]$

Trivial $\rightarrow p(k) := \tilde{\mathbf{u}}(k) := (\mathbf{u}(k), \dots, \mathbf{u}(k+N-1))^T \in \mathbb{R}^{Nn_u}$



Ideal MPC: Common Control Parametrization

Reduced $\rightarrow \tilde{\mathbf{u}}(k) = \mathcal{U}(k, p(k)), \qquad [n_p < Nn_u]$

Trivial $\rightarrow p(k) := \tilde{\mathbf{u}}(k) := (\mathbf{u}(k), \dots, \mathbf{u}(k+N-1))^T \in \mathbb{R}^{Nn_u}$

$$\text{Extended} \rightarrow p(k) = \begin{pmatrix} \tilde{\mathbf{u}}(k) \\ \tilde{\mathbf{x}}(k) \\ \tilde{\mu}(k) \end{pmatrix}, \qquad [n_p = N(n_u + n_x + n_c)]$$

$\tilde{\mu}(k)$: KKT conditions-related multiplier's profile

Ideal MPC: Common Control Parametrization

Reduced $\rightarrow \tilde{\mathbf{u}}(k) = \mathcal{U}(k, p(k)), \qquad [n_p < Nn_u]$

Trivial $\rightarrow p(k) := \tilde{\mathbf{u}}(k) := (\mathbf{u}(k), \dots, \mathbf{u}(k+N-1))^T \in \mathbb{R}^{Nn_u}$

Extended $\rightarrow p(k) = \begin{pmatrix} \tilde{\mathbf{u}}(k) \\ \tilde{\mathbf{x}}(k) \\ \tilde{\mu}(k) \end{pmatrix}, \qquad [n_p = N(n_u + n_x + n_c)]$

Dimension ↓ Structure

Ideal NMPC: Formal Definition

At each decision instant k

Solve a NLP problem,

$$p^*(\mathbf{x}(k)) \quad \leftarrow \min_p J(p \mid \mathbf{x}(k))$$
$$\text{under} \quad C(p \mid \mathbf{x}(k)) \le 0$$

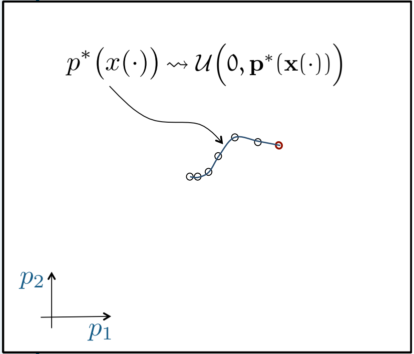
Apply the first control

$$\mathbf{u}(k) = \mathcal{U}\big(0, p^*(\mathbf{x}(k))\big)$$

over the updating period $[k, k+1]$.

Optimal cost

$$J^*(\mathbf{x}(k)) := J(p^*(\mathbf{x}(k)) \mid \mathbf{x}(k))$$



Ideal NMPC: Closed-Loop Stability

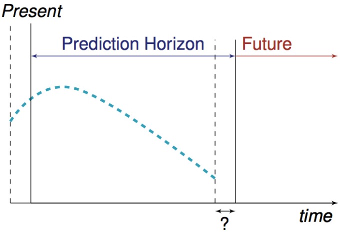
Provided that:

1. The formulation includes appropriate terminal assessment
2. The **optimal** solution $p^*(x(k))$ is obtained
3. The corresponding MPC control $\mathbf{u}(k) = \mathcal{U}\big(0, p^*(\mathbf{x}(k))\big)$ is applied

Stability can be proved based on:

$$J^*(\mathbf{x}(k+1)) \leq \quad J^*(\mathbf{x}(k)) - \ell(\mathbf{x}(k))$$

[Mayne et al. Automatica 2000]



Ideal NMPC: Closed-Loop Stability

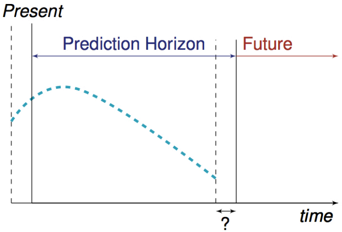
Provided that:

1. The formulation includes appropriate terminal assessment
2. The **optimal** solution $p^*(x(k))$ is obtained
3. The corresponding MPC control $\mathbf{u}(k) = \mathcal{U}\big(0, p^*(\mathbf{x}(k))\big)$ is applied

Stability can be proved based on:

$$J^*(\mathbf{x}(k+1)) - J^*(\mathbf{x}(k)) \leq -\ell(\mathbf{x}(k))$$

[Mayne et al. Automatica 2000]



Ideal NMPC: Closed-Loop Stability

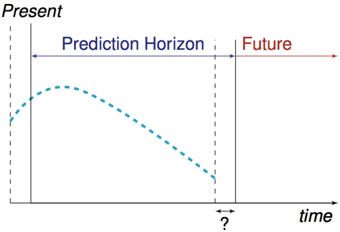
Provided that:

1. The formulation includes appropriate terminal assessment
2. The **optimal** solution $p^*(x(k))$ is obtained $\Leftarrow$ IDEAL !!!
3. The corresponding MPC control $\mathbf{u}(k) = \mathcal{U}\big(0, p^*(\mathbf{x}(k))\big)$ is applied

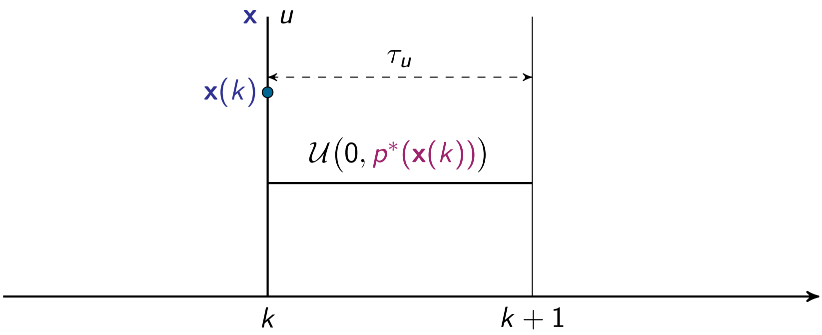
Stability can be proved based on:

$$J^*(\mathbf{x}(k+1)) \leq \quad J^*(\mathbf{x}(k)) - \ell(\mathbf{x}(k))$$

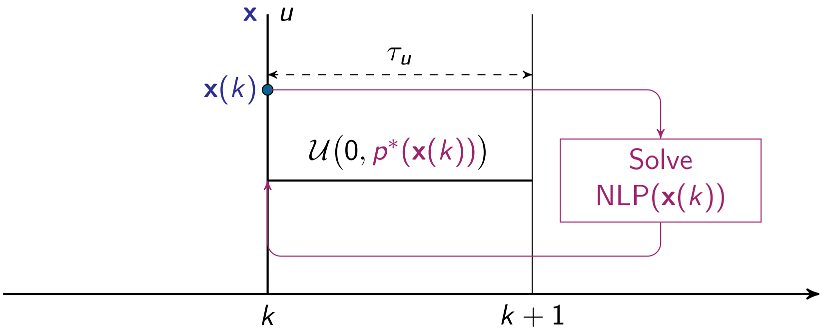
[Mayne et al. Automatica 2000]



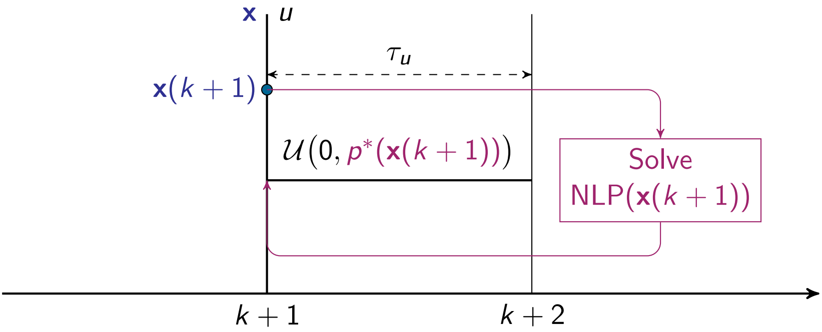
Ideal MPC: key assumption



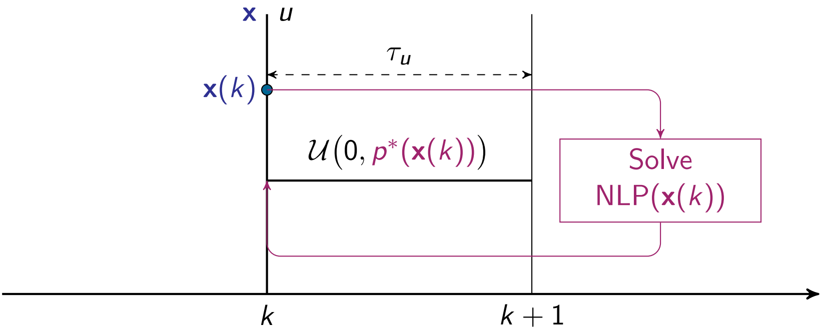
Ideal MPC: key assumption



Ideal MPC: key assumption



Ideal MPC: key assumption



Implicit assumption of ideal MPC

The time needed to get $p^*(\mathbf{x}(k))$ by solving NLP($\mathbf{x}(k)$) is negligible w.r.t the updating period τ_u.

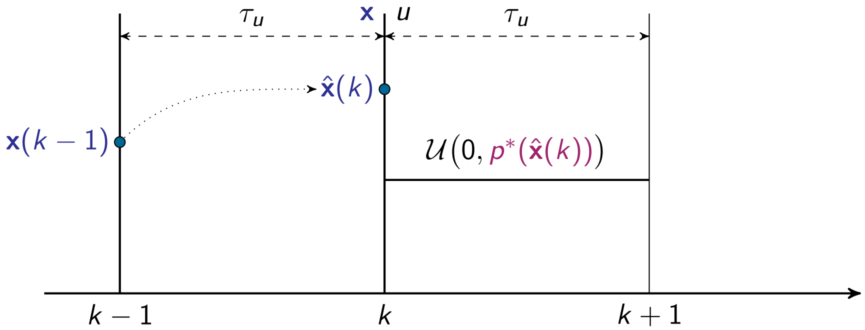
Agenda

1 Ideal NMPC

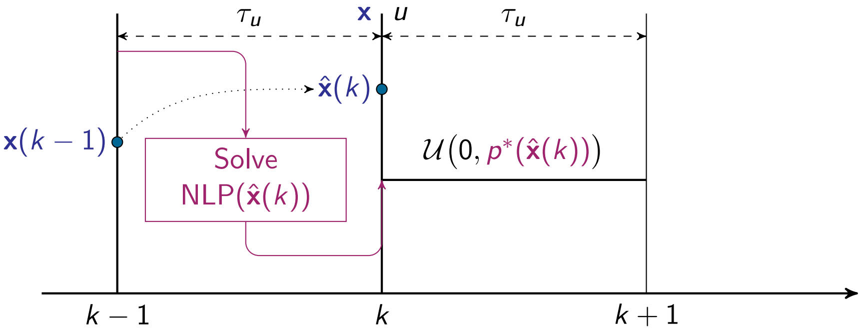
2 Fast NMPC
   What is it about?

3 Advanced Topics

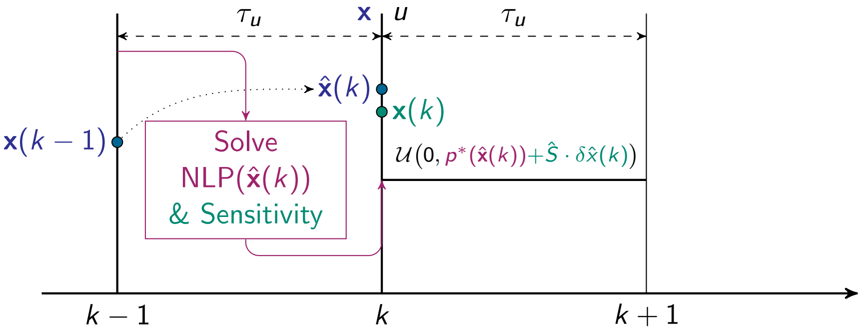
Quasi-Ideal MPC



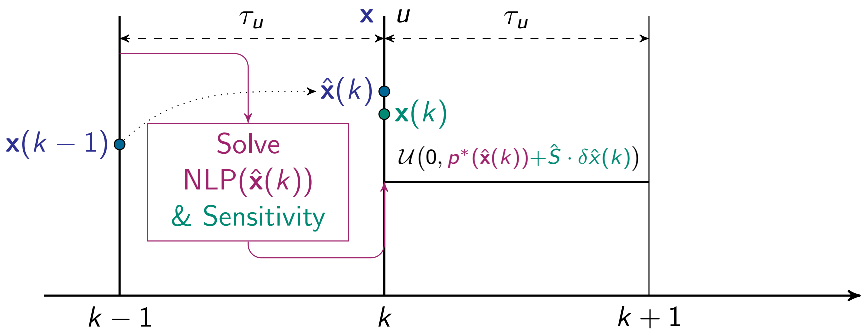
Quasi-Ideal MPC



Quasi-Ideal MPC



Quasi-Ideal MPC



Implicit assumption of quasi-ideal MPC

The time needed to get $p^*(\hat{\mathbf{x}}(k))$ by solving NLP($\hat{\mathbf{x}}(k)$) is lower than the updating period τ_u.

Definition of a fast system

- τ_u is the time between two updating of the strategy

Definition of a fast system

- τ_u is the time between two updating of the strategy
- τ_u it is the time during which there is NO FEEDBACK

Definition of a fast system

- τ_u is the time between two updating of the strategy
- τ_u it is the time during which there is NO FEEDBACK
- $\Rightarrow$ there is a system-dependent τ_u^{max} s.t. τ must satisfy

$$\tau_u \quad \leq \quad \tau_u^{max}$$

Definition of a fast system

- τ_u is the time between two updating of the strategy
- τ_u it is the time during which there is NO FEEDBACK
- $\Rightarrow$ there is a system-dependent τ_u^{max} s.t. τ must satisfy

$$\text{Quasi-Ideal MPC} \Rightarrow \quad \tau_{solve}(NLP(x)) \quad \leq \quad \tau_u \quad \leq \quad \tau_u^{max}$$

Definition of a fast system

- τ_u is the time between two updating of the strategy
- τ_u it is the time during which there is NO FEEDBACK
- $\Rightarrow$ there is a system-dependent τ_u^{max} s.t. τ must satisfy

$$\text{Quasi-Ideal MPC} \Rightarrow \quad \tau_{solve}(NLP(x)) \quad \leq \quad \tau_u \quad \leq \quad \tau_u^{max}$$

Fast Systems

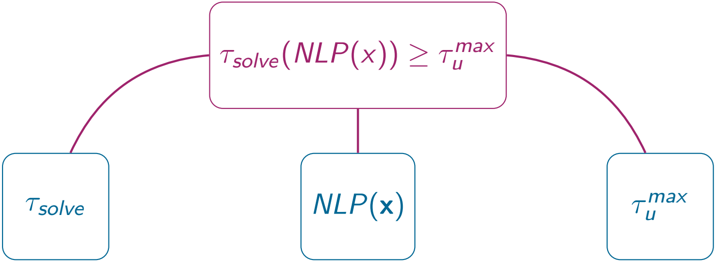
Fast systems are those for which:

$\tau_{solve}\big(NLP(x)\big)$ is greater than τ_u^{max}

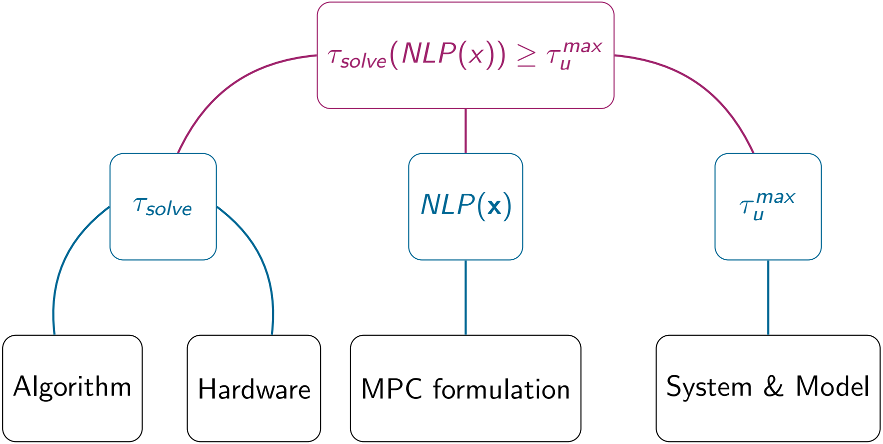
Did you say *Fast systems*?

$$\tau_{solve}(NLP(x)) \geq \tau_u^{max}$$

Did you say *Fast systems*?



Did you say *Fast systems*?



Fast MPC problems

Fast MPC problems are those for which:

$\tau_{solve}\big(NLP(x)\big)$ is greater than τ_u^{max}

Fast MPC problems

Fast MPC problems are those for which:

$\tau_{solve}\big(NLP(x)\big)$ is greater than τ_u^{max}

What happens in this case?

1. Updating is still necessary ($\leq \tau_u^{max}$)
2. *Optimal* solution $p^*(\mathbf{x}(k))$ cannot be obtained.

Implication of Fast MPC: $p(k)$ is a dynamic state

- $p^*(\mathbf{x}(k))$ is obtained through an iterative process:

$$p^{(i)} \leftarrow \mathcal{S}\big(p^{(i-1)}, \mathbf{x}(k)\big)$$

Implication of Fast MPC: $p(k)$ is a dynamic state

- $p^*(\mathbf{x}(k))$ is obtained through an iterative process:

$$p^{(i)} \leftarrow \mathcal{S}\big(p^{(i-1)}, \mathbf{x}(k)\big) \qquad \Rightarrow \qquad [p^{(q)} \leftarrow \mathcal{S}^{(q)}(p^{(0)}, \mathbf{x}(k))]$$

Implication of Fast MPC: $p(k)$ is a dynamic state

- $p^*(\mathbf{x}(k))$ is obtained through an iterative process:

$$p^{(i)} \leftarrow \mathcal{S}\big(p^{(i-1)}, \mathbf{x}(k)\big) \qquad \Rightarrow \qquad [p^{(q)} \leftarrow \mathcal{S}^{(q)}(p^{(0)}, \mathbf{x}(k))]$$

- Assume that a single iteration needs τ_1 time unit.

Implication of Fast MPC: $p(k)$ is a dynamic state

- $p^*(\mathbf{x}(k))$ is obtained through an iterative process:

$$p^{(i)} \leftarrow \mathcal{S}\Big(p^{(i-1)}, \mathbf{x}(k)\Big) \qquad \Rightarrow \qquad [p^{(q)} \leftarrow \mathcal{S}^{(q)}(p^{(0)}, \mathbf{x}(k))]$$

- Assume that a single iteration needs τ_1 time unit.

$$q \leq q^{max} := E\Big(\frac{\tau_u^{max}}{\tau_1}\Big)$$

Implication of Fast MPC: $p(k)$ is a dynamic state

- $p^*(\mathbf{x}(k))$ is obtained through an iterative process:

$$p^{(i)} \leftarrow \mathcal{S}\big(p^{(i-1)}, \mathbf{x}(k)\big) \quad \Rightarrow \quad [p^{(q)} \leftarrow \mathcal{S}^{(q)}(p^{(0)}, \mathbf{x}(k))]$$

- Assume that a single iteration needs τ_1 time unit.

$$q \leq q^{max} := E\left(\frac{\tau_u^{max}}{\tau_1}\right)$$

$$p(k) = \mathcal{S}^{(q)}(p^+(k-1), \mathbf{x}(k))$$

Implication of Fast MPC: $p(k)$ is a dynamic state

- $p^*(\mathbf{x}(k))$ is obtained through an iterative process:

$$p^{(i)} \leftarrow \mathcal{S}\big(p^{(i-1)}, \mathbf{x}(k)\big) \qquad \Rightarrow \qquad [p^{(q)} \leftarrow \mathcal{S}^{(q)}(p^{(0)}, \mathbf{x}(k))]$$

- Assume that a single iteration needs τ_1 time unit.

$$q \leq q^{max} := E\left(\frac{\tau_u^{max}}{\tau_1}\right) \rightarrow p(k) = \mathcal{S}^{(q)}(p^+(k-1), \mathbf{x}(k))$$

- $p(k) \neq p^*(\mathbf{x}(k))$ but rather an **dynamic internal state of the controller**.

Implication of Fast MPC: The Extended System

Fast MPC Extended System

At updating instants, the system is described by:

$$\mathbf{x}(k+1) = f\Big(\mathbf{x}(k), \overbrace{\mathcal{U}(0, p(k))}^{\mathbf{u}(k)}\Big)$$

Implication of Fast MPC: The Extended System

Fast MPC Extended System

At updating instants, the system is described by:

$$\begin{aligned}
\mathbf{x}(k+1) &= f\big(\mathbf{x}(k), \overbrace{\mathcal{U}(0, p(k))}^{\mathbf{u}(k)}\big) \\
p(k+1) &= \mathcal{S}^{(q)}\big(p^{+}(k), \mathbf{x}(k)\big)
\end{aligned}$$

Ideal NMPC: Formal Definition

At each decision instant k

Solve a NLP problem,

$$p^*(\mathbf{x}(k)) \quad \leftarrow \min_p J\big(p \mid \mathbf{x}(k)\big)$$
$$\text{under} \quad C(p \mid \mathbf{x}(k)) \leq 0$$

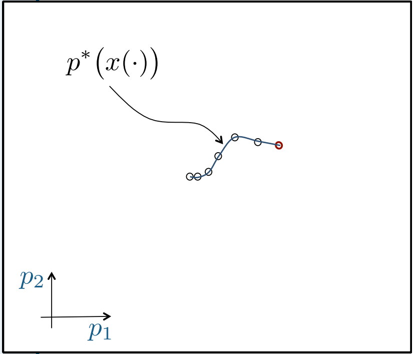
Apply the first control

$$\mathbf{u}(k) = \mathcal{U}\big(0, p^*(\mathbf{x}(k))\big)$$

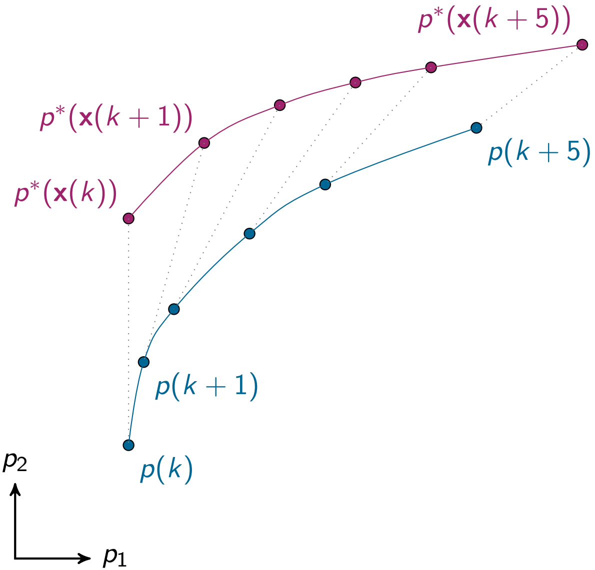
over the updating period $[k, k+1]$.

Optimal cost

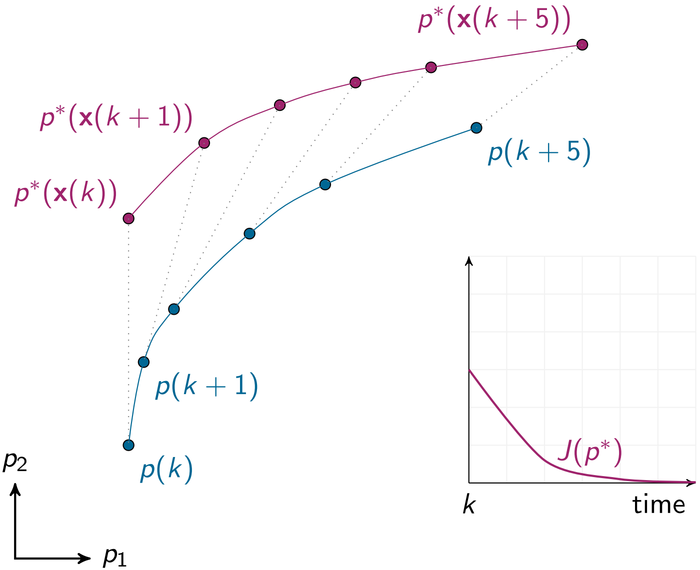
$$J^*(\mathbf{x}(k)) := J(p^*(\mathbf{x}(k)) \mid \mathbf{x}(k))$$



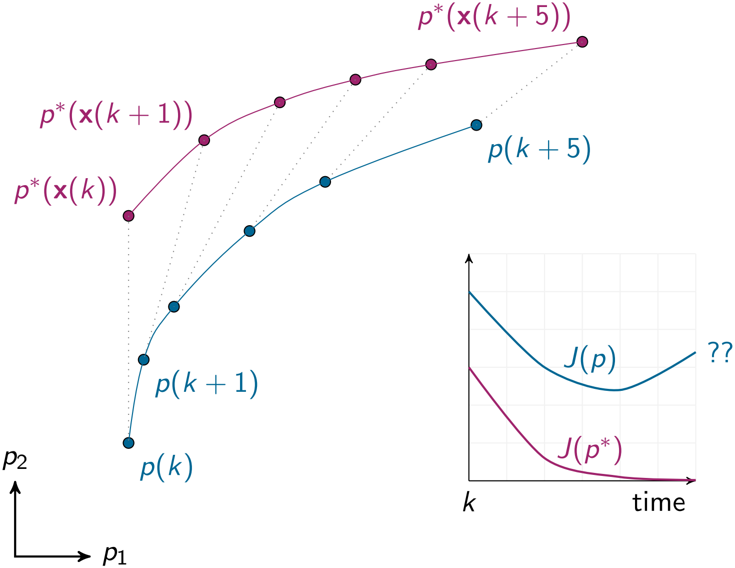
Optimal $p^*(\mathbf{x}(k))$ vs achievable $p(k)$



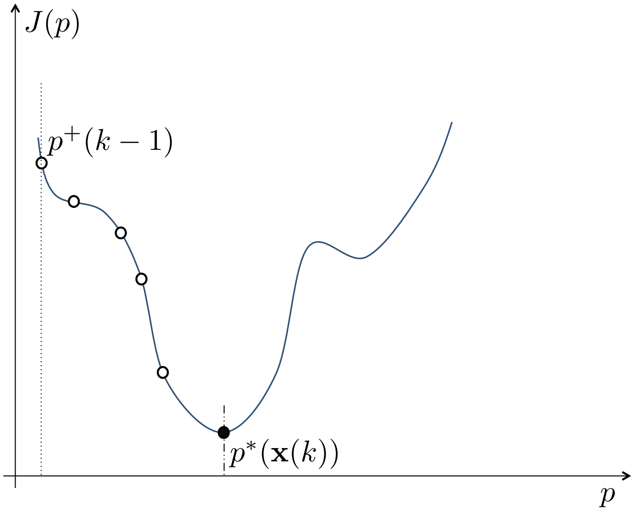
Optimal $p^*(\mathbf{x}(k))$ vs achievable $p(k)$



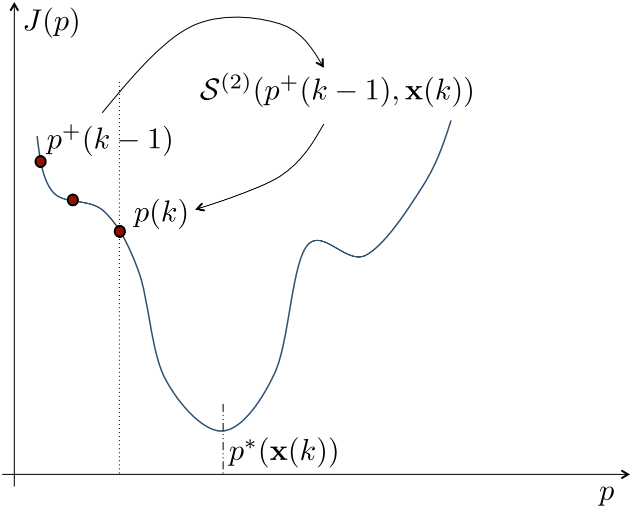
Optimal $p^*(\mathbf{x}(k))$ vs achievable $p(k)$



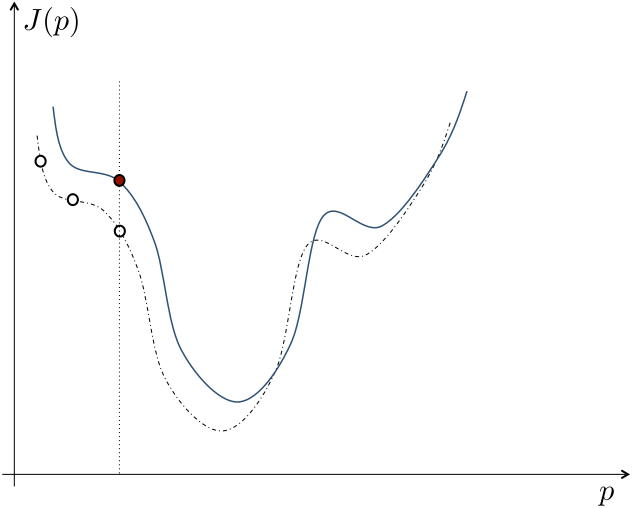
Closed-Loop Evolution of the Cost Function



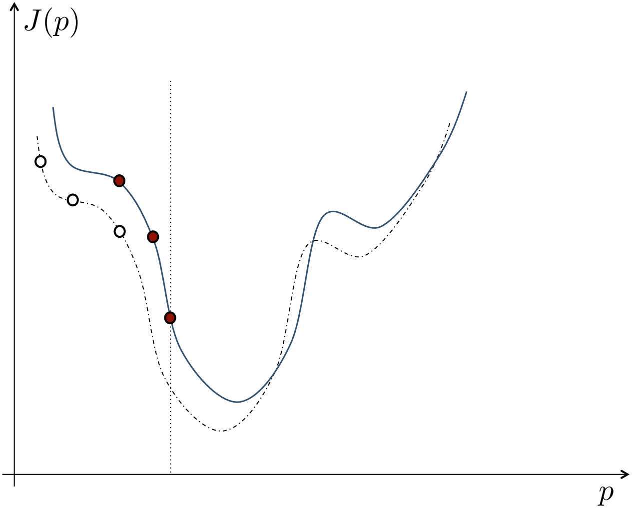
Closed-Loop Evolution of the Cost Function



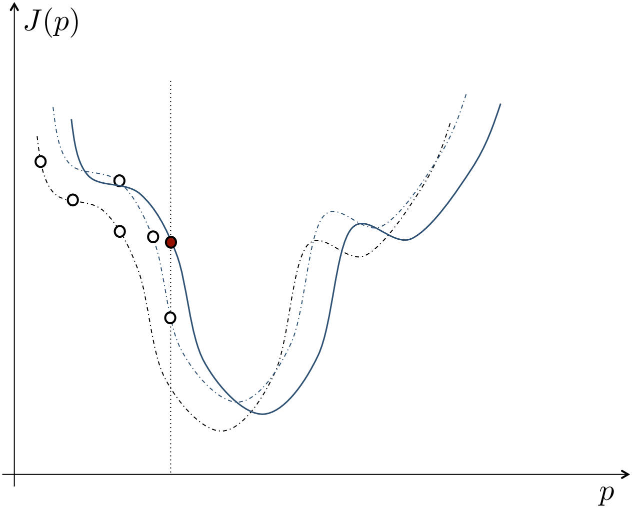
Closed-Loop Evolution of the Cost Function



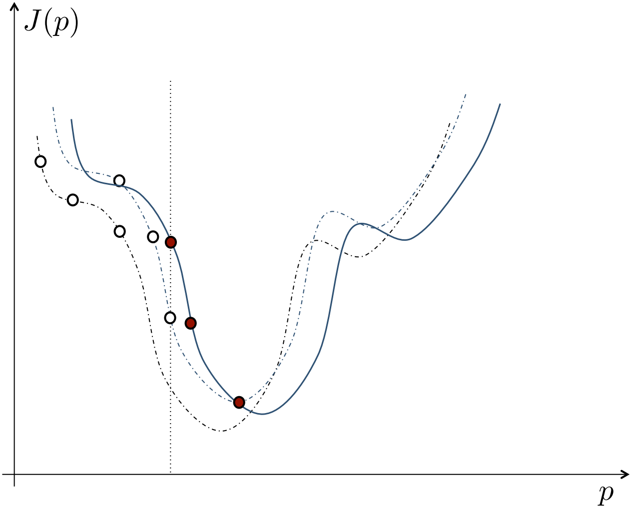
Closed-Loop Evolution of the Cost Function



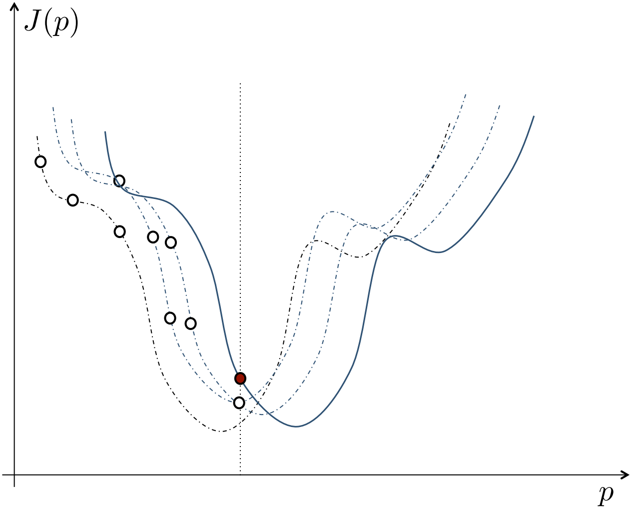
Closed-Loop Evolution of the Cost Function



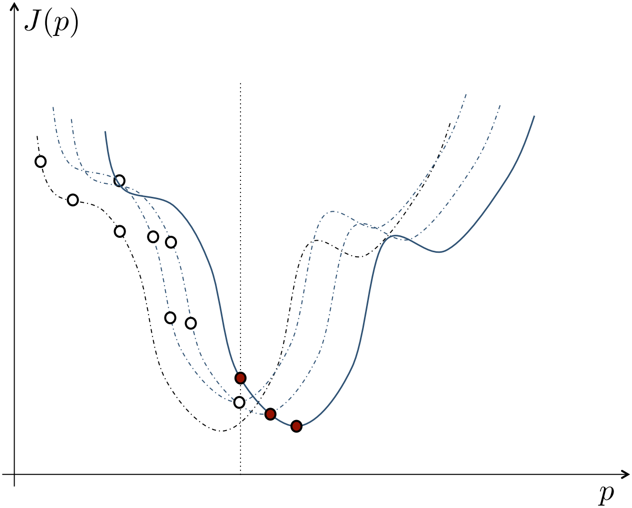
Closed-Loop Evolution of the Cost Function



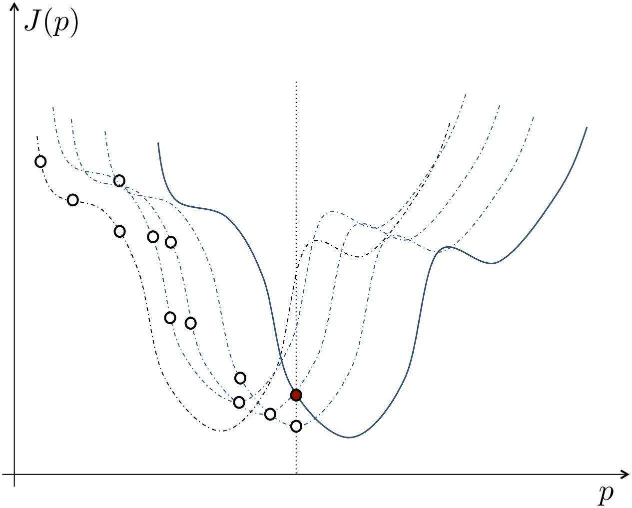
Closed-Loop Evolution of the Cost Function



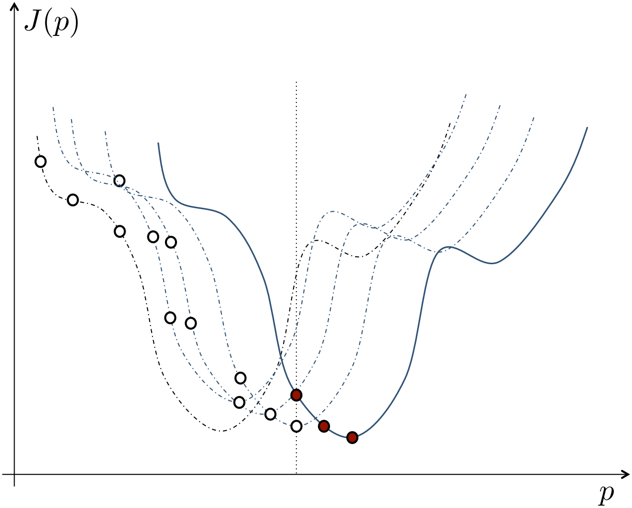
Closed-Loop Evolution of the Cost Function



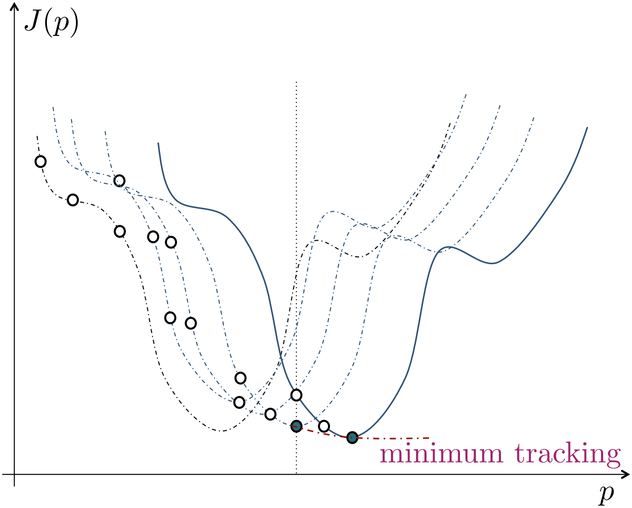
Closed-Loop Evolution of the Cost Function



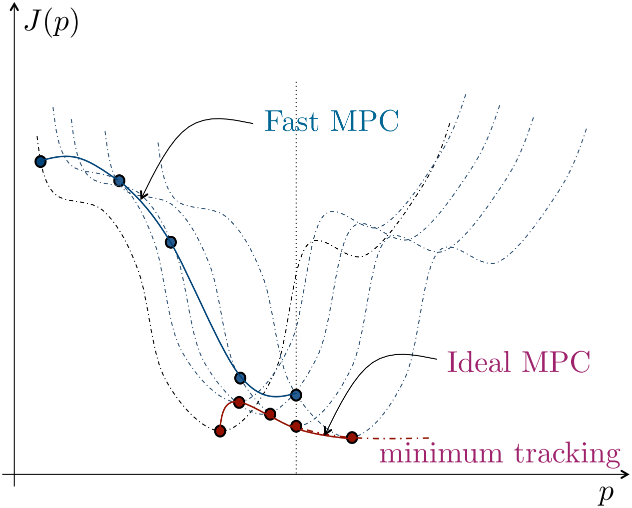
Closed-Loop Evolution of the Cost Function



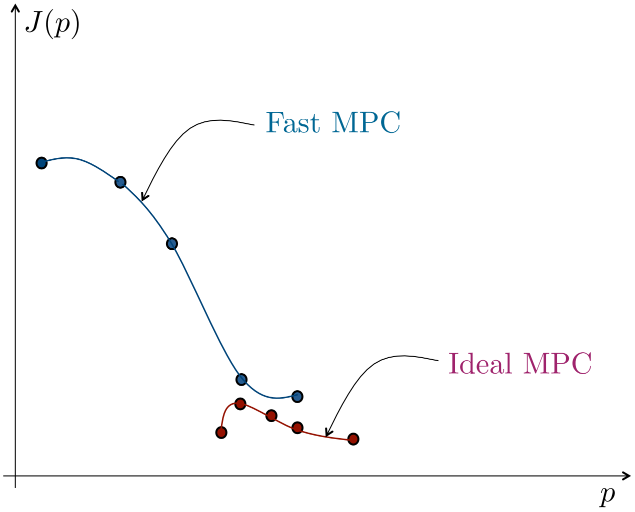
Closed-Loop Evolution of the Cost Function



Closed-Loop Evolution of the Cost Function



Closed-Loop Evolution of the Cost Function

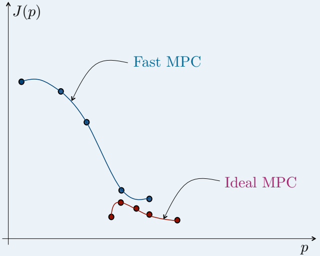


Fast MPC: A Competition

Game view of Fast MPC

Fast MPC: a competition:

Cost	$J(p, x)$
Good guy	Optimizer managing p
Bad guy	Reality [a] managing x

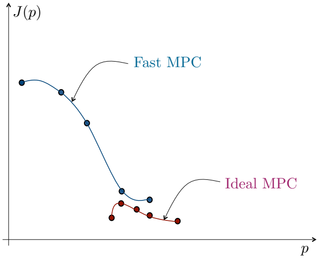


[a] fast dynamic, model mismatch, disturbance, set-point changes, control parametrization, bad MPC formulations, etc.

Fast MPC: Intuitive Set of Success Conditions

Success Conditions

1. No finite escape time.
2. Efficient iteration vs not so fast dynamics
3. Stabilizing NMPC formulation (in the ideal case)



[Diehl et al. IEE Proc. Control Theory Appl. (2005)]

Agenda

1. Ideal NMPC
2. Fast NMPC
3. Advanced Topics
   - Monitoring the updating period
   - Quantum of Computation
   - Miscellaneous

The Controlled Extended System

Fast MPC Extended System

At updating instants, the system is described by:

$$
\begin{aligned}
\mathbf{x}(k+1) &= f\big(\mathbf{x}(k), \mathcal{U}(0, p(k))\big) \\
p(k+1) &= \mathcal{S}^{(q)}\big(p^{+}(k), \mathbf{x}(k)\big)
\end{aligned}
$$

The Controlled Extended System

Fast MPC Extended System

At updating instants, the system is described by:

$$\mathbf{z}(k+1) = F(\mathbf{z}(k), q)$$

$$\mathbf{z} := (p, \mathbf{x})$$

The Controlled Extended System

Fast MPC Extended System

At updating instants, the system is described by:

$$\mathbf{z}(k+1) = F(\mathbf{z}(k), q)$$
$$y(k) = J(\mathbf{z}(k))$$
$$\mathbf{z} := (p, \mathbf{x})$$

At each decision instant k

Solve a NLP problem,

$$p^*(\mathbf{x}(k)) \quad \leftarrow \min_p J\big(p \mid \mathbf{x}(k)\big)$$
$$\text{under} \quad C\big(p \mid \mathbf{x}(k)\big) \leq 0$$

The Controlled Extended System

Fast MPC Extended System

At updating instants, the system is described by:

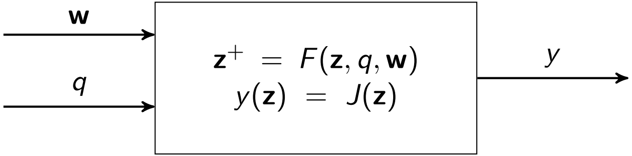
$$\begin{aligned} \mathbf{z}(k+1) &= F(\mathbf{z}(k), q, \mathbf{w}) \\ y(k) &= J(\mathbf{z}(k)) \\ \mathbf{z} &:= (p, \mathbf{x}) \end{aligned}$$

The Controlled Extended System

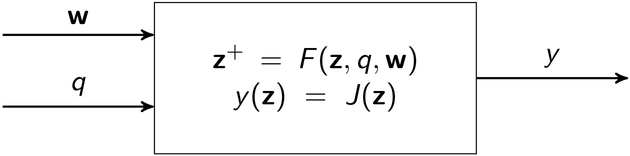
Fast MPC Extended System

At updating instants, the system is described by:

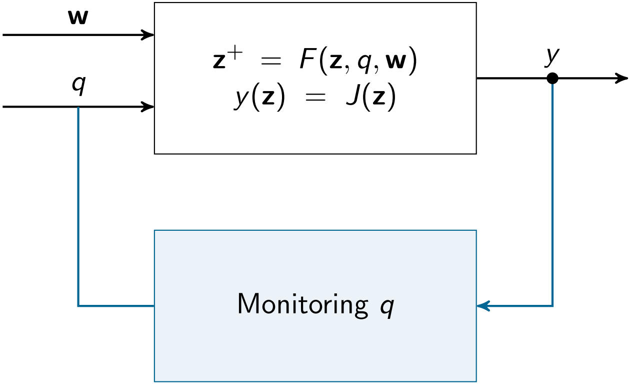
$$\mathbf{z}(k+1) = F(\mathbf{z}(k), q, \mathbf{w})$$
$$y(k) = J(\mathbf{z}(k))$$
$$\mathbf{z} := (p, \mathbf{x})$$



Updating the Number of Iterations



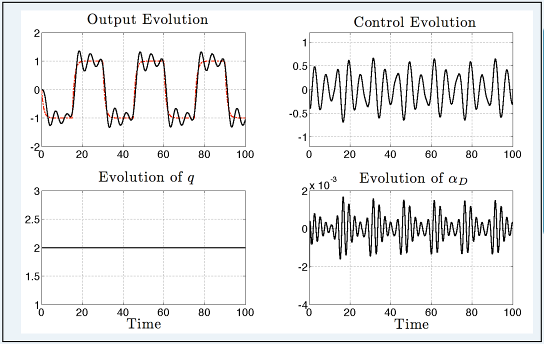
Updating the Number of Iterations

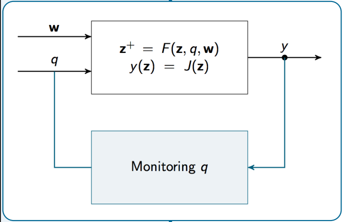


[MA, ECC2013, Zurich.]

Example

$q = 2$ without adaptation

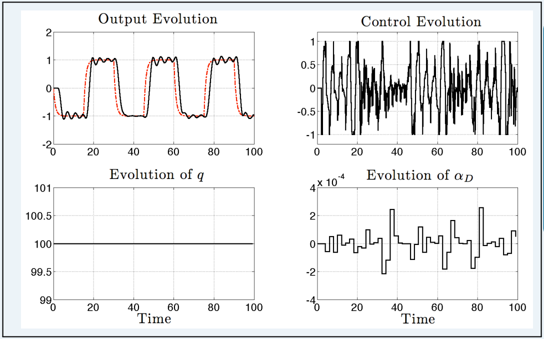


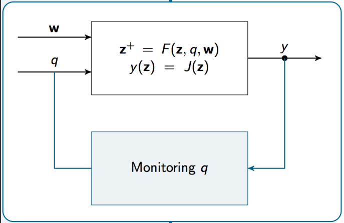


[MA, ECC2013, Zurich.]

Example

$q = 100$ without adaptation

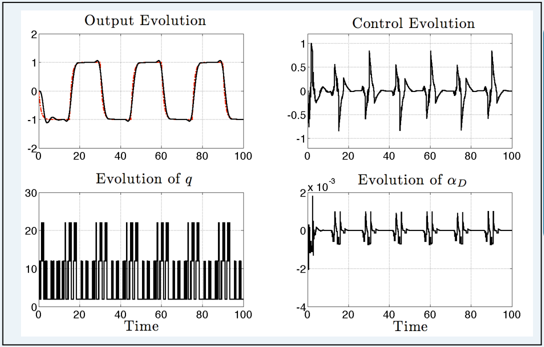


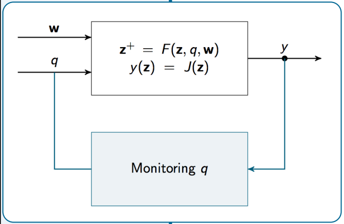


[MA, ECC2013, Zurich.]

Example

$q^{(0)} = 2$ with adaptation

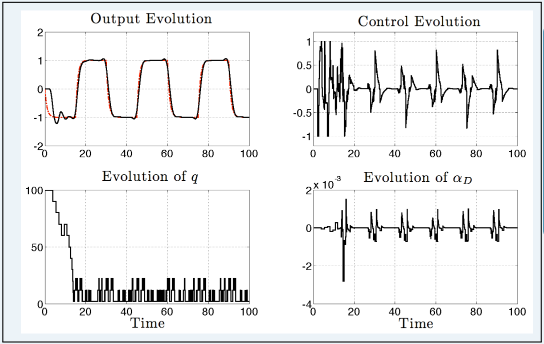


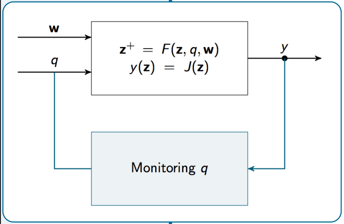


[MA, ECC2013, Zurich.]

Example

$q^{(0)} = 100$ with adaptation





[MA, ECC2013, Zurich.]

Agenda

1 Ideal NMPC

2 Fast NMPC

3 Advanced Topics
- Monitoring the updating period
- Quantum of Computation
- Miscellaneous

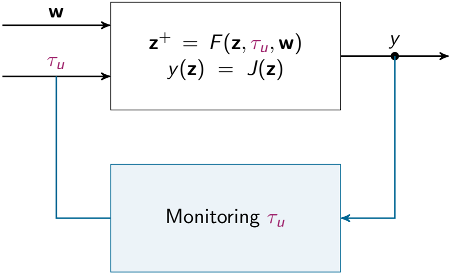
The Quantum amount of Interruptible Computation

$$\tau_u = q \times \tau_1$$

τ_u	Updating period
q	Number of iterations
τ_1	Time for a single iteration.

The Quantum amount of Interruptible Computation

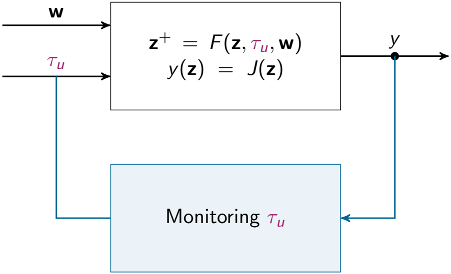
$$\tau_u = q \times \tau_1$$



τ_u	Updating period
q	Number of iterations
τ_1	Time for a single iteration.

The Quantum amount of Interruptible Computation

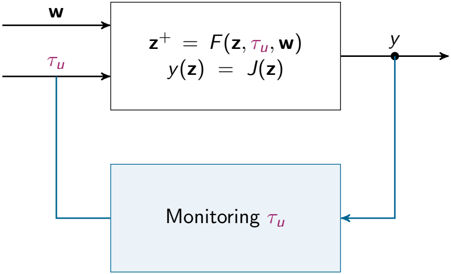
$$\tau_u = q \times \tau_1$$



τ_u	Updating period
q	Number of iterations
τ_1	**Time for a single iteration**.

The Quantum amount of Interruptible Computation

$$\tau_u = q \times \tau_1$$



τ_u Updating period

q Number of iterations

τ_1 **Time for a single iteration**.

(Newton-like, gradient, partial gradient, parametrization, etc.) does not show the same τ_1 and hence enable different updating periods.

[Semi-Plenary, ECC2014]

Agenda

1 Ideal NMPC

2 Fast NMPC

3 Advanced Topics
- Monitoring the updating period
- Quantum of Computation
- Miscellaneous

Other Topics . . .

Some Facts

- Many ready-to-use and user friendly software are available.
- Parallel computing on multi-core processors coming soon.
- Transposition to Moving-Horizon-Observers implementation

Some open issues

- Explicit NMPC vs on-line Fast MPC?
- Fast robust min/max frameworks?
- Automatic tuning?