Fast NMPC

Some Good News and Some Facts to Keep in Mind

Mazen Alamir

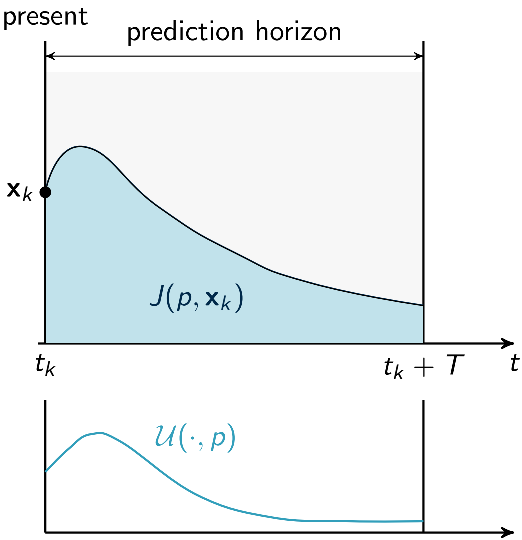
CNRS, University of Grenoble

Outline

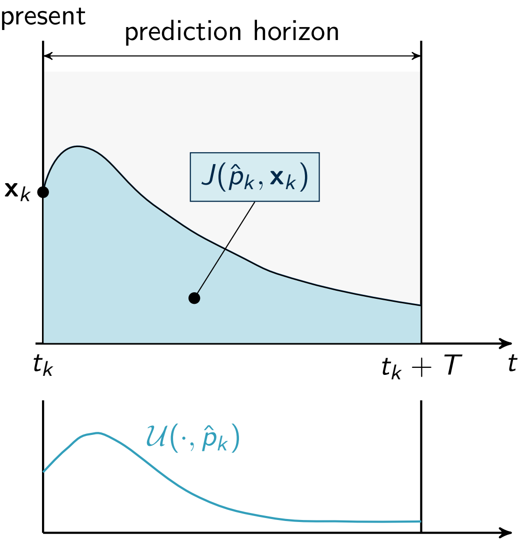
Outline

- Ideal NMPC
- Definition of Fast NMPC
- Intuitive Sufficient Conditions

Ideal Framework: Recalls & Basic Notations



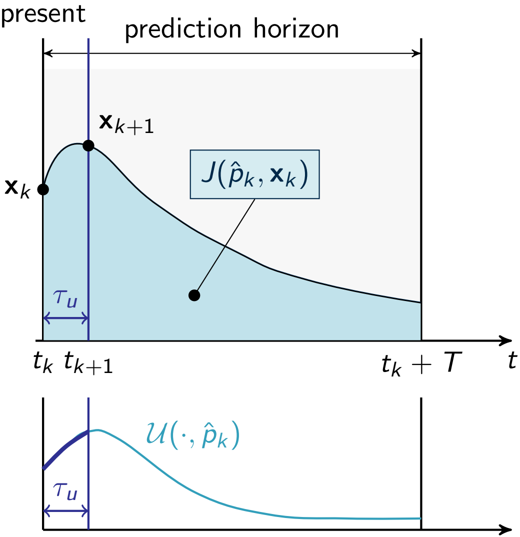
Ideal Framework: Recalls & Basic Notations



$$\min_p \; J(p, \mathbf{x}_k) \text{ s.t. } C(p, \mathbf{x}_k) \leq 0$$

$$\hat{p}_k = \hat{p}(\mathbf{x}_k)$$

Ideal Framework: Recalls & Basic Notations



$$\min_p \; J(p, \mathbf{x}_k) \text{ s.t. } C(p, \mathbf{x}_k) \leq 0$$

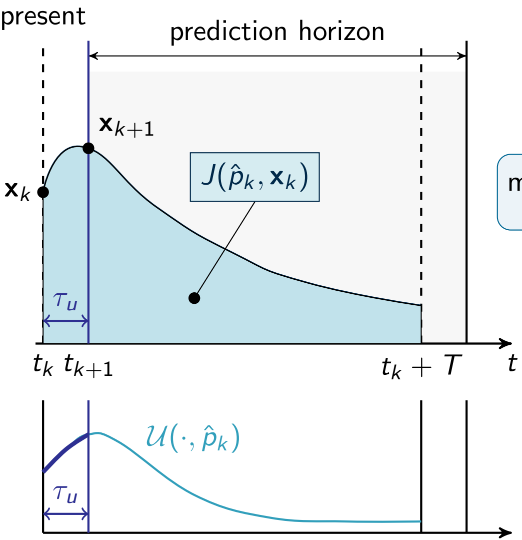
↓

$$\hat{p}_k = \hat{p}(\mathbf{x}_k)$$

↓

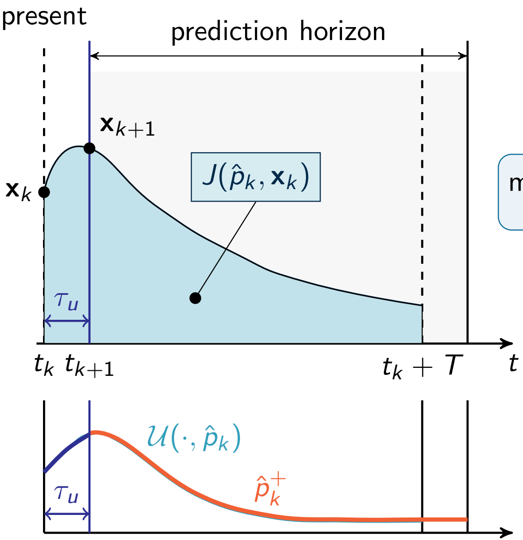
Apply $\mathcal{U}(\cdot, \hat{p}(\mathbf{x}_k))$ during τ_u

Ideal Framework: Recalls & Basic Notations



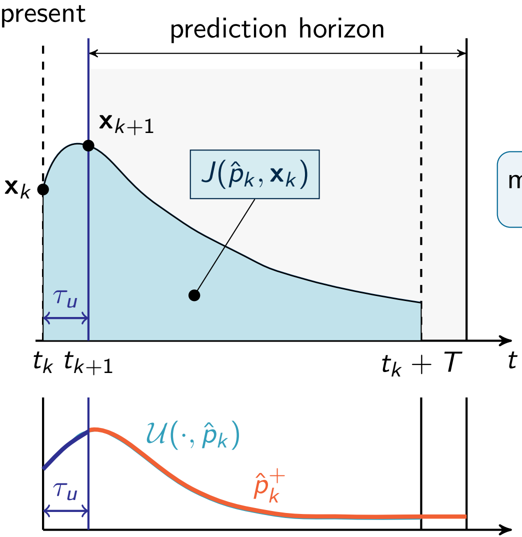
$$\min_p J(p, \mathbf{x}_{k+1}) \text{ s.t. } C(p, \mathbf{x}_{k+1}) \leq 0$$

Ideal Framework: Recalls & Basic Notations



$$\min_p \; J(p, \mathbf{x}_{k+1}) \text{ s.t. } C(p, \mathbf{x}_{k+1}) \leq 0$$

Ideal Framework: Recalls & Basic Notations

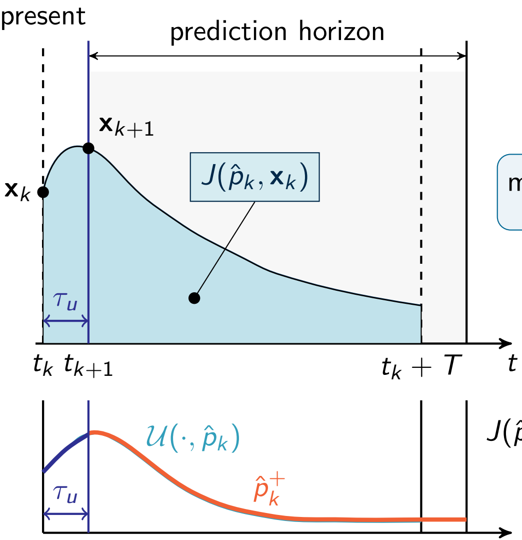


$$\min_p J(p, \mathbf{x}_{k+1}) \text{ s.t. } C(p, \mathbf{x}_{k+1}) \leq 0$$

Initialize at $\hat{p}_k^+$ (hot start)

$$\hat{p}_{k+1} = \hat{p}(\mathbf{x}_{k+1})$$

Ideal Framework: Recalls & Basic Notations



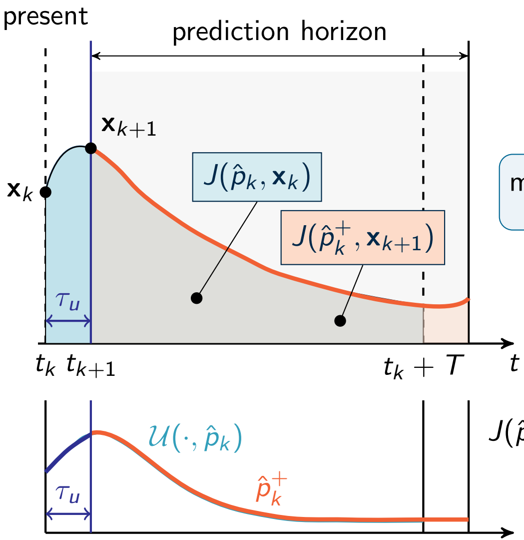
$$\min_{p} J(p, \mathbf{x}_{k+1}) \text{ s.t. } C(p, \mathbf{x}_{k+1}) \leq 0$$

Initialize at $\hat{p}_k^+$ (hot start)

$$\hat{p}_{k+1} = \hat{p}(\mathbf{x}_{k+1})$$

$$J(\hat{p}_{k+1}, \mathbf{x}_{k+1}) \leq J(\hat{p}_k^+, \mathbf{x}_{k+1})$$

Ideal Framework: Recalls & Basic Notations



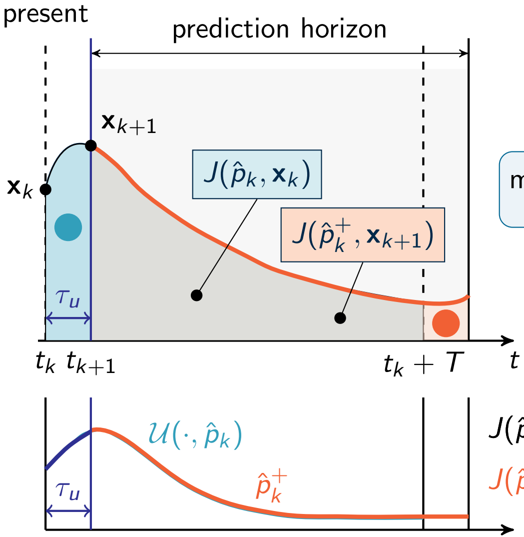
$$\min_p J(p, \mathbf{x}_{k+1}) \text{ s.t. } C(p, \mathbf{x}_{k+1}) \le 0$$

Initialize at $\hat{p}_k^+$ (hot start)

$$\hat{p}_{k+1} = \hat{p}(\mathbf{x}_{k+1})$$

$$J(\hat{p}_{k+1}, \mathbf{x}_{k+1}) \le J(\hat{p}_k^+, \mathbf{x}_{k+1})$$

Ideal Framework: Recalls & Basic Notations



$$\min_p \; J(p, \mathbf{x}_{k+1}) \text{ s.t. } C(p, \mathbf{x}_{k+1}) \leq 0$$

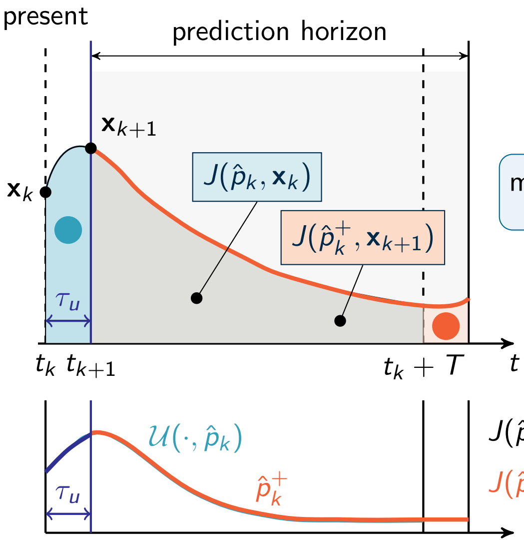
Initialize at $\hat{p}_k^+$ (hot start)

$$\hat{p}_{k+1} = \hat{p}(\mathbf{x}_{k+1})$$

$$J(\hat{p}_{k+1}, \mathbf{x}_{k+1}) \leq J(\hat{p}_k^+, \mathbf{x}_{k+1})$$

$$J(\hat{p}_k^+, \mathbf{x}_{k+1}) = J(\hat{p}_k, \mathbf{x}_k) - \bullet + \bullet$$

Ideal Framework: Recalls & Basic Notations



$$\min_p J(p, \mathbf{x}_{k+1}) \text{ s.t. } C(p, \mathbf{x}_{k+1}) \leq 0$$

Initialize at $\hat{p}_k^+$ (hot start)

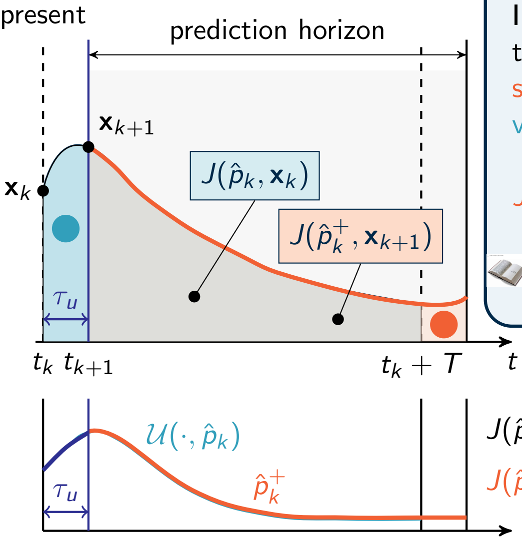
$$\hat{p}_{k+1} = \hat{p}(\mathbf{x}_{k+1})$$

$$J(\hat{p}_{k+1}, \mathbf{x}_{k+1}) \leq J(\hat{p}_k^+, \mathbf{x}_{k+1})$$

$$J(\hat{p}_k^+, \mathbf{x}_{k+1}) = J(\hat{p}_k, \mathbf{x}_k) - \bullet + \bullet$$

≤ 0 ?

Ideal Framework: Recalls & Basic Notations



Keep in mind

In the **ideal framework**, when the horizon moves, the hot start $\hat{p}_k^+$ computed from the previous optimal solution $\hat{p}_k$ satisfies:

$$J(\hat{p}_k^+, \mathbf{x}_{k+1}) \leq J(\hat{p}_k, \mathbf{x}_k)$$

Mayne et al. Automatica (2000)

$$J(\hat{p}_{k+1}, \mathbf{x}_{k+1}) \leq J(\hat{p}_k^+, \mathbf{x}_{k+1})$$

$$J(\hat{p}_k^+, \mathbf{x}_{k+1}) = J(\hat{p}_k, \mathbf{x}_k) - \bullet + \bullet$$

≤ 0 ?

Ideal MPC: The key assumptions

Keep in mind

In the **ideal framework**, when the horizon moves, the hot start $\hat{p}_k^+$ computed from the previous optimal solution $\hat{p}_k$ satisfies:

$$J(\hat{p}_k^+, \mathbf{x}_{k+1}) \le J(\hat{p}_k, \mathbf{x}_k)$$

Mayne et al. Automatica (2000).

- Formulation involving Final constraints
- (hot start)-compatible parametrization
- $\hat{p}_k$ sufficiently good

Ideal MPC: The key assumptions

Keep in mind

In a **realistic framework**, when the horizon moves, the hot start p_k^+ computed from the previous solution p_k satisfies:

$$J(p_k^+, \mathbf{x}_{k+1}) = J(p_k, \mathbf{x}_k) + D(\tau_u)$$

$D(0) = 0$
$D(\tau_u)$ is **not necessarily** ≤ 0.

- Formulation involving Final constraints
- (hot start)-compatible parametrization
- $\hat{p}_k$ sufficiently good

Ideal MPC: The key assumptions

Keep in mind

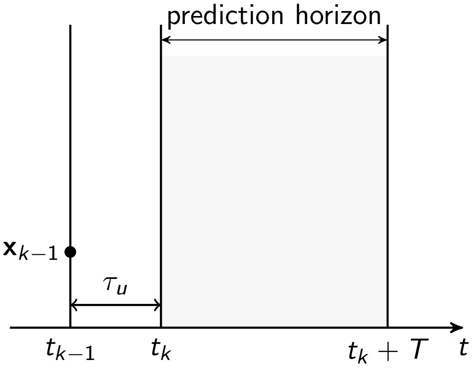
In a **realistic framework**, when the horizon moves, the hot start p_k^+ computed from the previous solution p_k satisfies:

$$J(p_k^+, \mathbf{x}_{k+1}) = J(p_k, \mathbf{x}_k) + D(\tau_u)$$

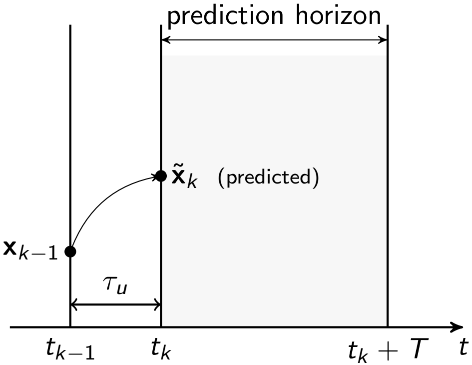
$D(0) = 0$
$D(\tau_u)$ is **not necessarily** ≤ 0.

Even with **perfect undisturbed** model

Preparation & Feedback Steps



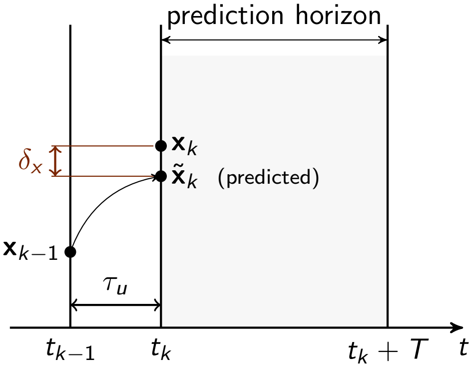
Preparation & Feedback Steps



1. Predict $\tilde{\mathbf{x}}_k$
2. During $[t_{k-1}, t_k]$

   Compute $\hat{p}(\tilde{\mathbf{x}}_k)$ [and $\frac{\partial \hat{p}_k}{\partial \mathbf{x}}$]

Preparation & Feedback Steps



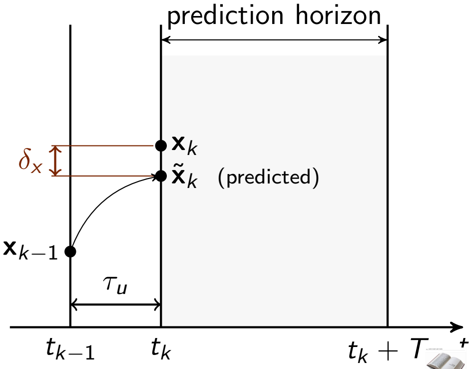
1. Predict $\tilde{\mathbf{x}}_k$
2. During $[t_{k-1}, t_k]$

   Compute $\hat{p}(\tilde{\mathbf{x}}_k)$ [and $\dfrac{\partial \hat{p}_k}{\partial \mathbf{x}}$]

3. Once $\mathbf{x}_k$ is available:

$$\hat{p}_k \leftarrow \hat{p}(\tilde{\mathbf{x}}_k) + \left[\frac{\partial \hat{p}_k}{\partial \mathbf{x}}\right] \cdot \delta_x$$

Preparation & Feedback Steps



1. Predict $\tilde{\mathbf{x}}_k$
2. During $[t_{k-1}, t_k]$

   Compute $\hat{p}(\tilde{\mathbf{x}}_k)$ [and $\dfrac{\partial \hat{p}_k}{\partial \mathbf{x}}$]

   (*preparation step*)
3. Once $\mathbf{x}_k$ is available:

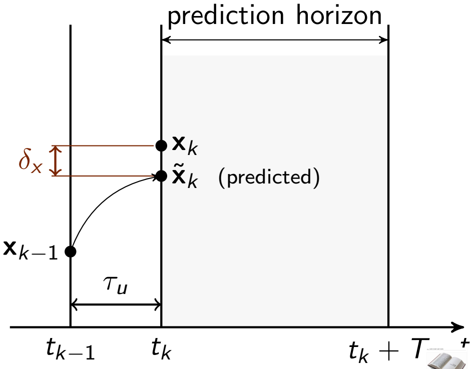
$$\hat{p}_k \leftarrow \hat{p}(\tilde{\mathbf{x}}_k) + \left[\frac{\partial \hat{p}_k}{\partial \mathbf{x}}\right] \cdot \delta_x$$

(*feedback step*)

Diehl et al. SIAM J. Ctrl and Opt. (2005)

Zavala and Biegler. Auomatica (2009)

Preparation & Feedback Steps



1. Predict $\tilde{\mathbf{x}}_k$
2. During $[t_{k-1}, t_k]$

   Compute $\hat{p}(\tilde{\mathbf{x}}_k)$ [and $\dfrac{\partial \hat{p}_k}{\partial \mathbf{x}}$]

   (*preparation step*)
3. Once $\mathbf{x}_k$ is available:

$$\hat{p}_k \leftarrow \hat{p}(\tilde{\mathbf{x}}_k) + \left[\frac{\partial \hat{p}_k}{\partial \mathbf{x}}\right] \cdot \delta_x$$

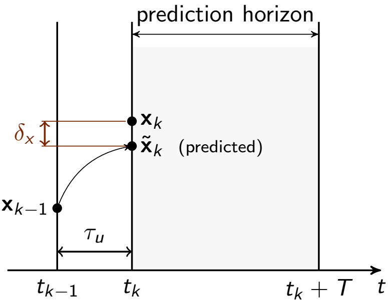
(*feedback step*)

Diehl et al. SIAM J. Ctrl and Opt. (2005)

Zavala and Biegler. Auomatica (2009)

NOTA: Even with feedback, uncertainty potentially increases $D(\tau_u)$

Definition of Fast NMPC Problems

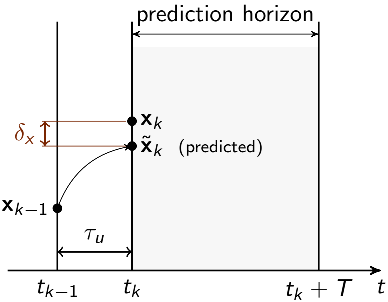


τ_u is the time between two control updating

τ_u is the time during which there is no feedback

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

Definition of Fast NMPC Problems



τ_u is the time between two control updating

τ_u is the time during which there is no feedback

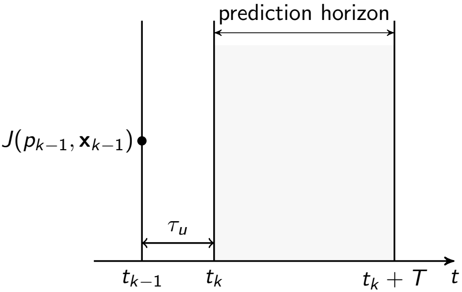
$$\Rightarrow \tau_u \leq \tau_u^{max}$$

Fast NMPC Problems

Fast NMPC problems are those for which

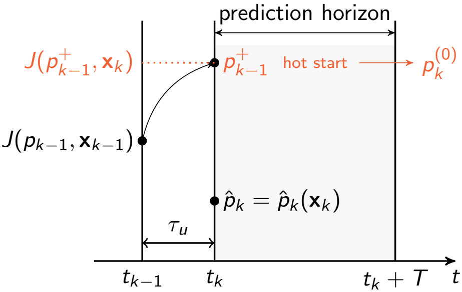
$$\tau_{solve}\Big(NLP(\tilde{\mathbf{x}}_k)\Big) \geq \tau_u^{max}$$

The Iterative Process



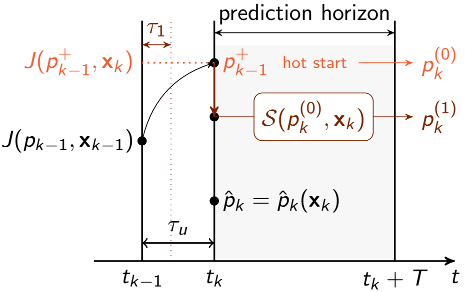
$$p^{(i+1)} \leftarrow \mathcal{S}(p^{(i)}, \mathbf{x})$$

The Iterative Process



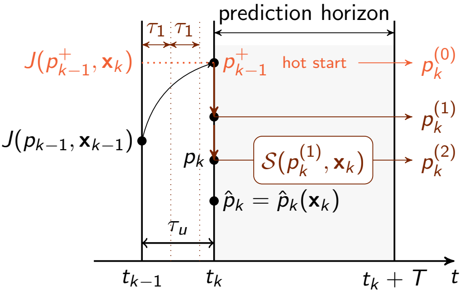
$$p^{(i+1)} \leftarrow \mathcal{S}(p^{(i)}, \mathbf{x})$$

The Iterative Process



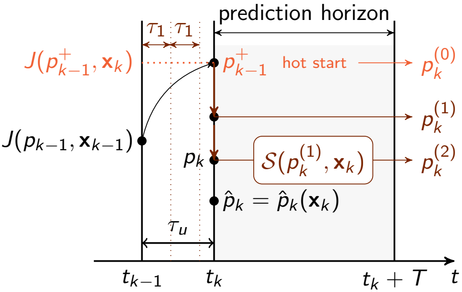
$$p^{(i+1)} \leftarrow \mathcal{S}(p^{(i)}, \mathbf{x})$$

The Iterative Process



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

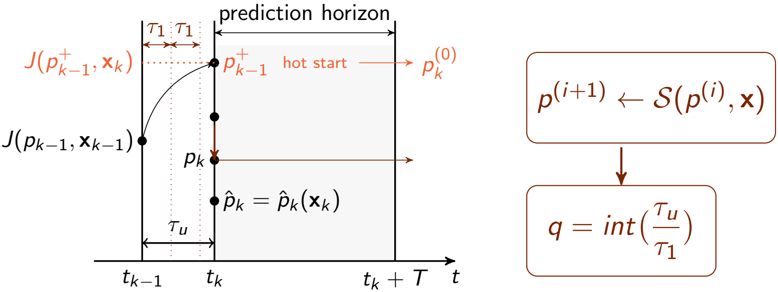
The Iterative Process



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

$$q = int\left(\frac{\tau_u}{\tau_1}\right)$$

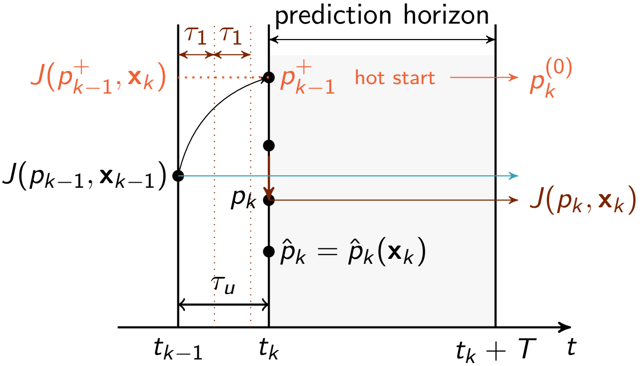
The Iterative Process



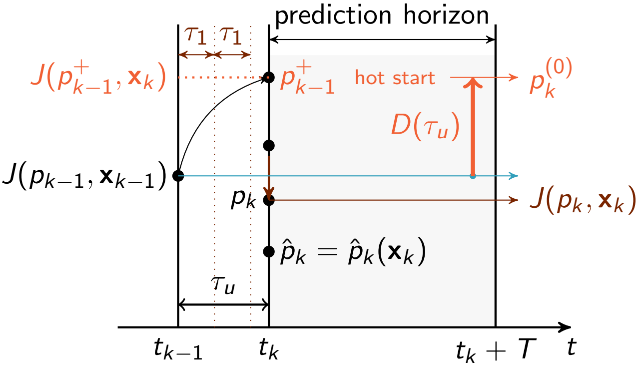
dynamic equation for p

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

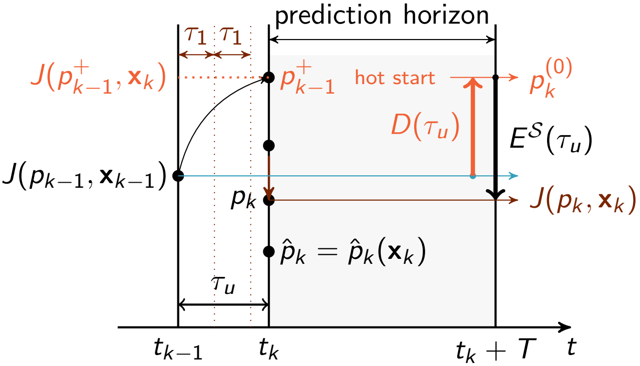
Sufficient Conditions of Success



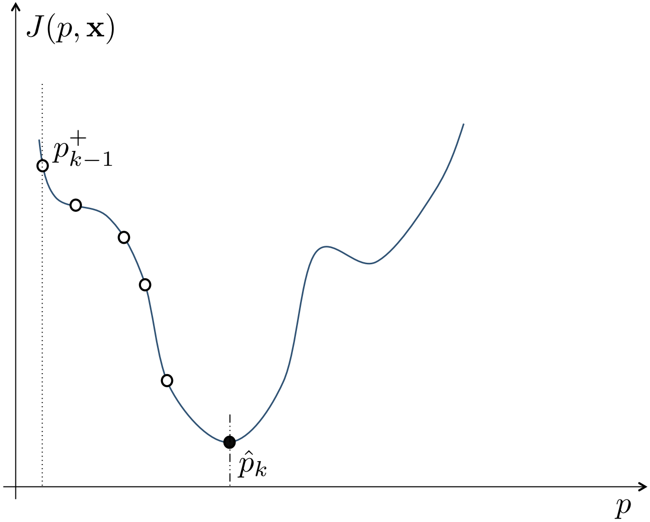
Sufficient Conditions of Success



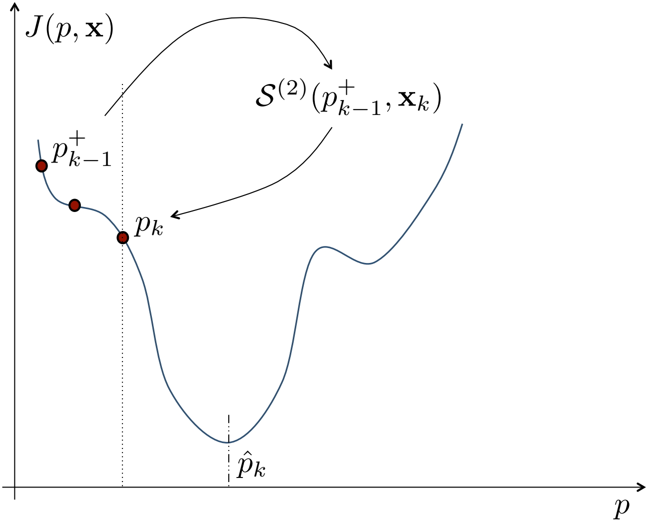
Sufficient Conditions of Success



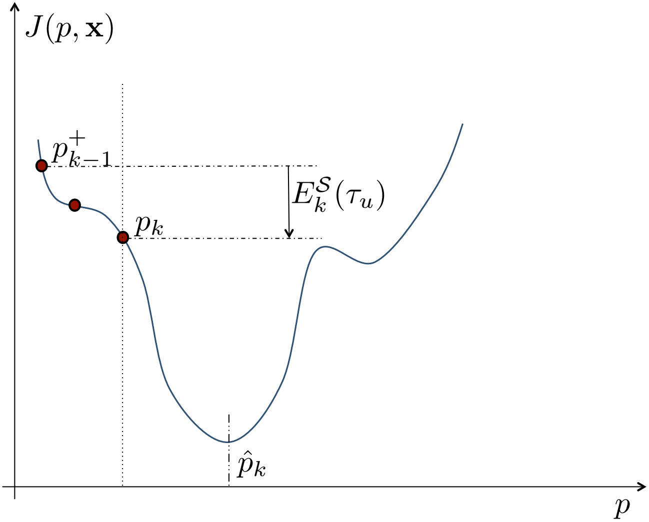
Closed-Loop Evolution of the Cost Function



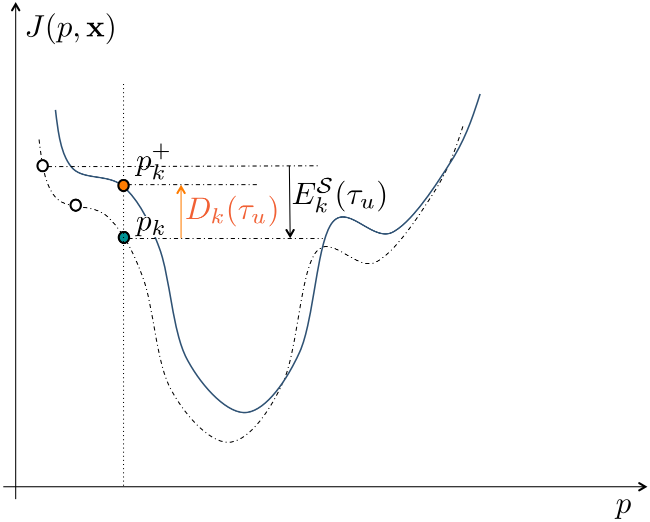
Closed-Loop Evolution of the Cost Function



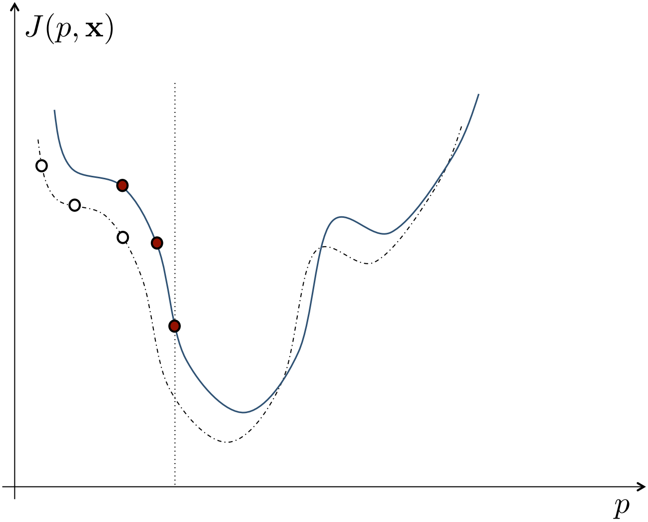
Closed-Loop Evolution of the Cost Function



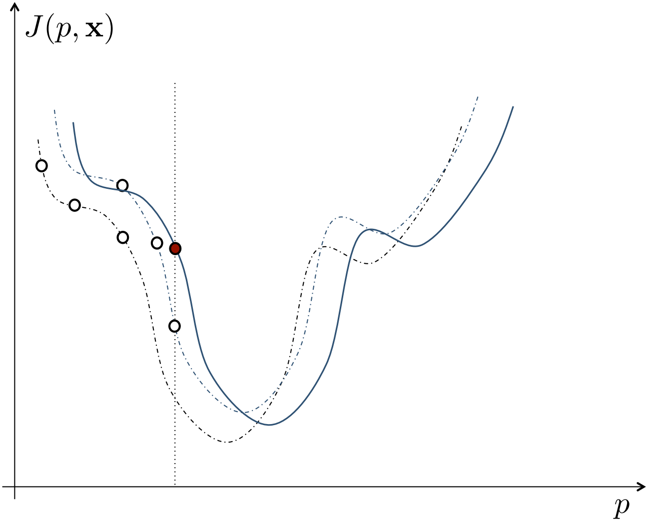
Closed-Loop Evolution of the Cost Function



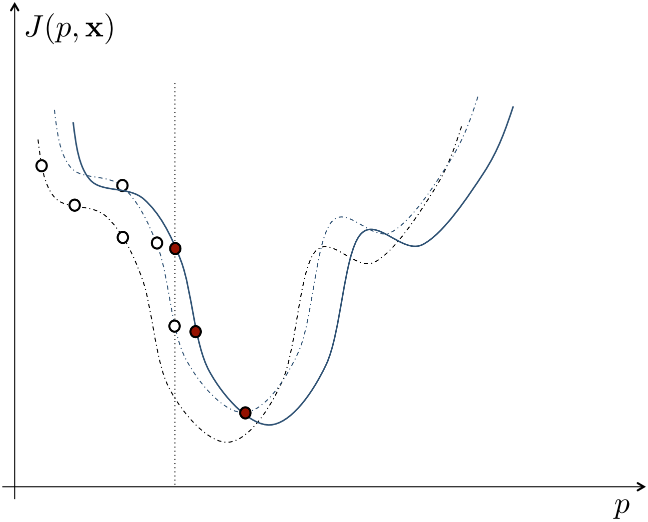
Closed-Loop Evolution of the Cost Function



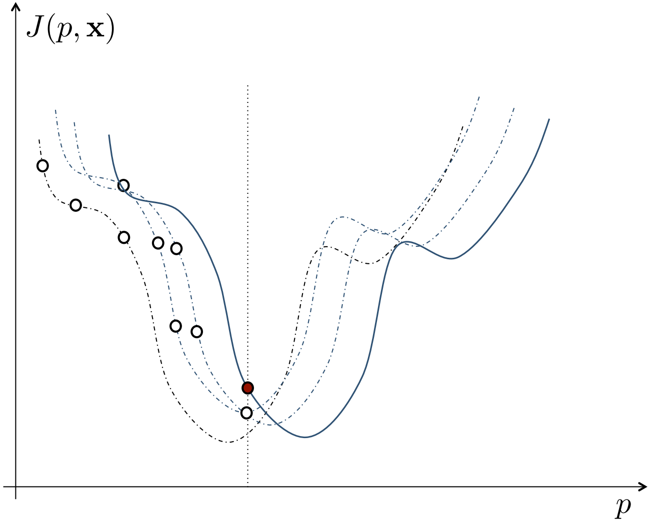
Closed-Loop Evolution of the Cost Function



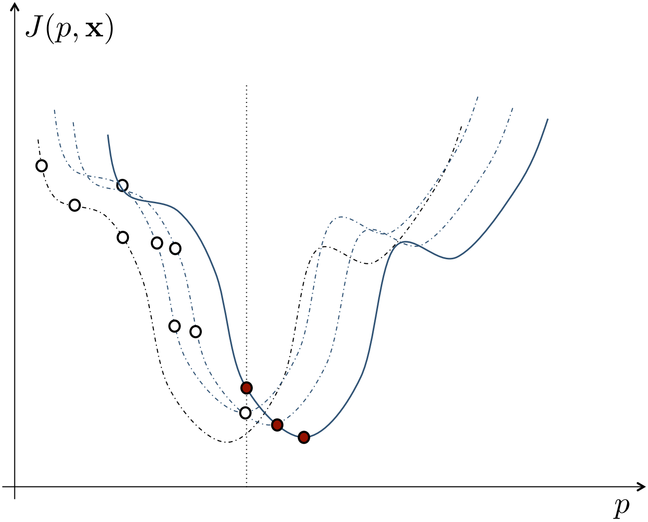
Closed-Loop Evolution of the Cost Function



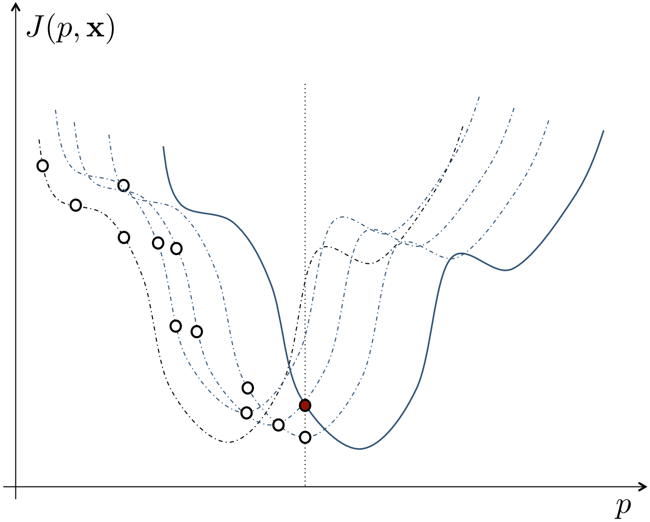
Closed-Loop Evolution of the Cost Function



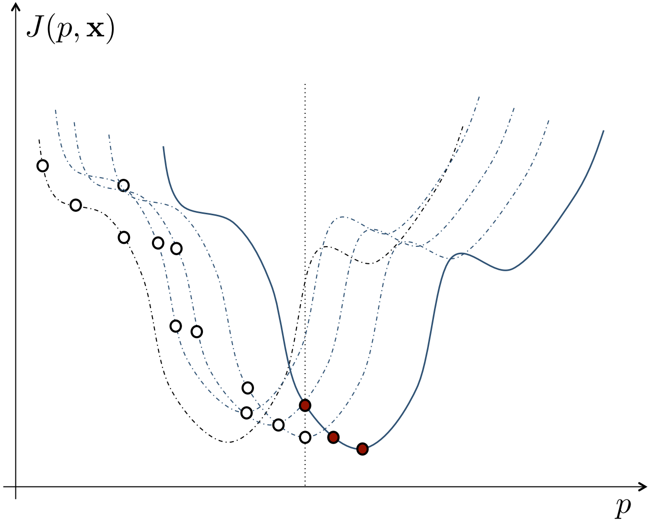
Closed-Loop Evolution of the Cost Function



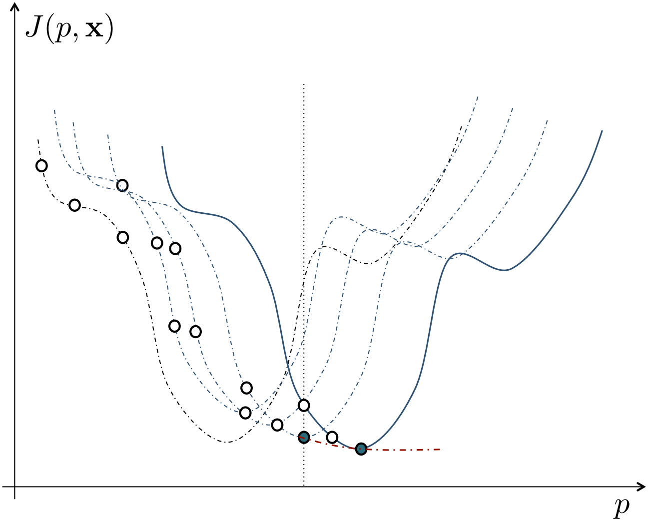
Closed-Loop Evolution of the Cost Function



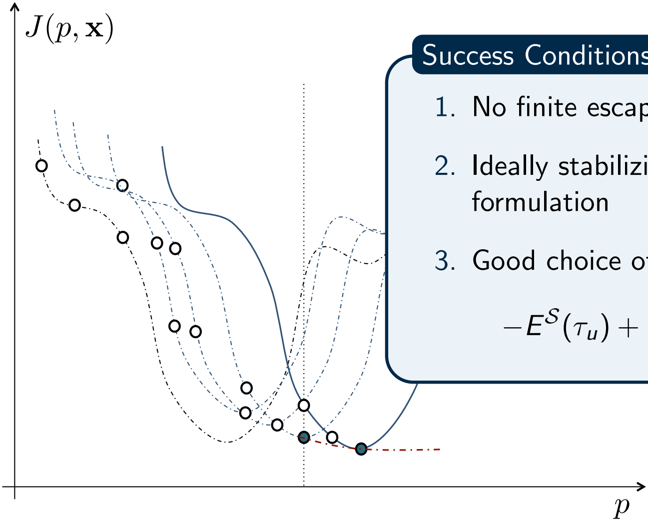
Closed-Loop Evolution of the Cost Function



Closed-Loop Evolution of the Cost Function



Closed-Loop Evolution of the Cost Function

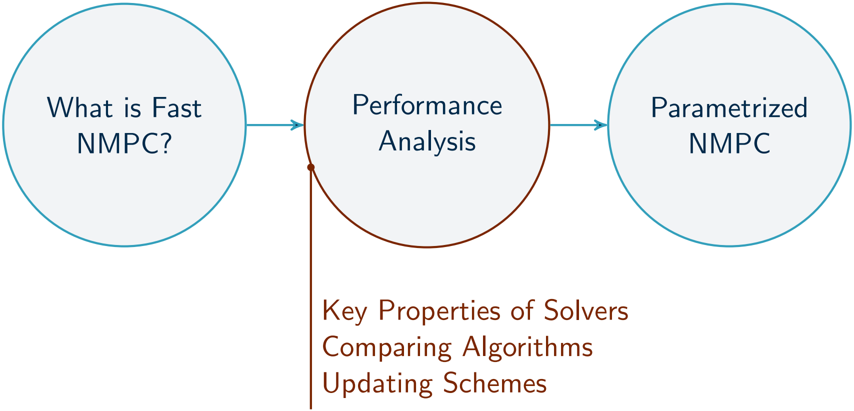


Success Conditions

1. No finite escape time
2. Ideally stabilizing NMPC formulation
3. Good choice of $(\mathcal{S}, \tau_u)$:

$$-E^{\mathcal{S}}(\tau_u) + D(\tau_u) < 0$$

Outline


What is Fast NMPC?
Performance Analysis
Parametrized NMPC
Key Properties of Solvers
Comparing Algorithms
Updating Schemes

Reminder

$$D(\tau_u) - E^{\mathcal{S}}(\tau_u) < 0$$

Reminder

$$D(\tau_u) - E^{\mathcal{S}}(\tau_u) < 0$$

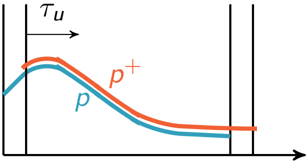
$D(\tau_u)$

$-E^{\mathcal{S}}(\tau_u)$

Reminder

$$D(\tau_u) - E^{\mathcal{S}}(\tau_u) < 0$$

$D(\tau_u)$



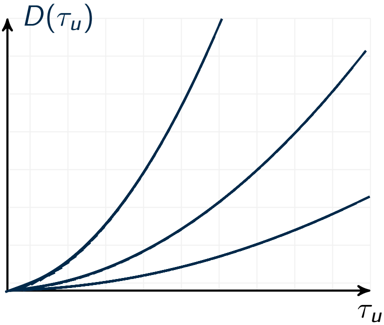
- $D(\tau_u) := J(p^+, \mathbf{x}^+) - J(p, \mathbf{x})$
- $D(0) = 0$, $D(\tau_u)$ can be ≥ 0
- $\tau_u \in [0, \tau_u^{max}]$
- Independent of the solver $\mathcal{S}$

$-E^{\mathcal{S}}(\tau_u)$

Reminder

$$D(\tau_u) - E^{\mathcal{S}}(\tau_u) < 0$$

$D(\tau_u)$



$-E^{\mathcal{S}}(\tau_u)$

Reminder

$$D(\tau_u) - E^{\mathcal{S}}(\tau_u) < 0$$

$D(\tau_u)$

$-E^{\mathcal{S}}(\tau_u)$

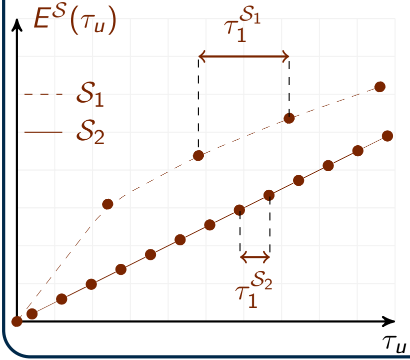
- $E^{\mathcal{S}}(\tau_u) := J(p^{(0)}, \mathbf{x}) - J(p^{(q^{\mathcal{S}})}, \mathbf{x})$
- $q^{\mathcal{S}} = int(\tau_u/\tau_1^{\mathcal{S}})$
- $\tau_1^{\mathcal{S}}$ time for a single iteration
- $\tau_u \in \{1, 2, \dots\} \times \tau_1^{\mathcal{S}}$
- $E^{\mathcal{S}}(0) = 0$

Reminder

$$D(\tau_u) - E^{\mathcal{S}}(\tau_u) < 0$$

$D(\tau_u)$

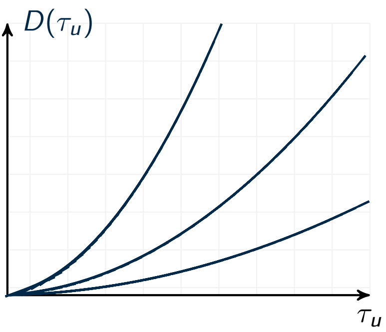
$-E^{\mathcal{S}}(\tau_u)$



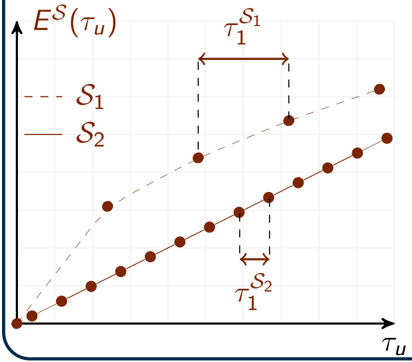
Reminder

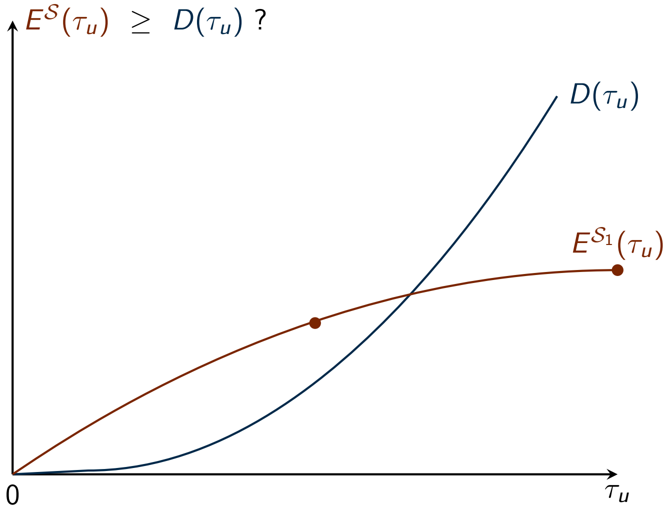
$$D(\tau_u) - E^{\mathcal{S}}(\tau_u) < 0$$

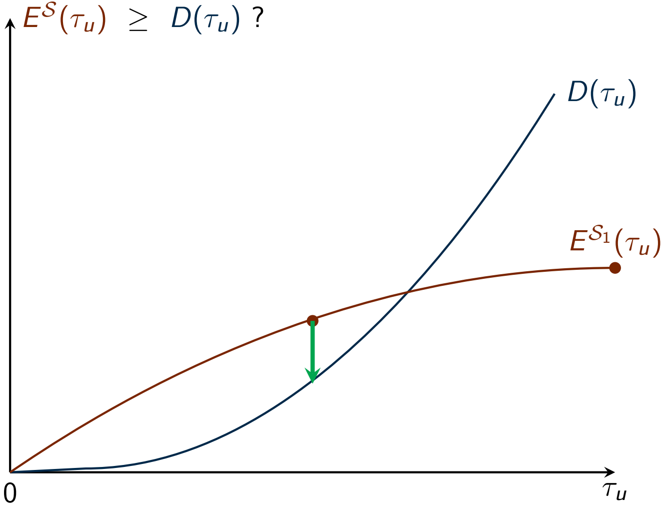
$D(\tau_u)$

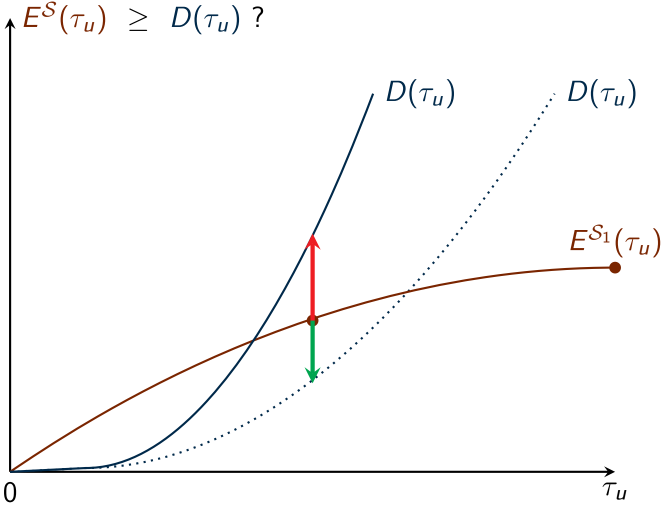


$-E^{\mathcal{S}}(\tau_u)$



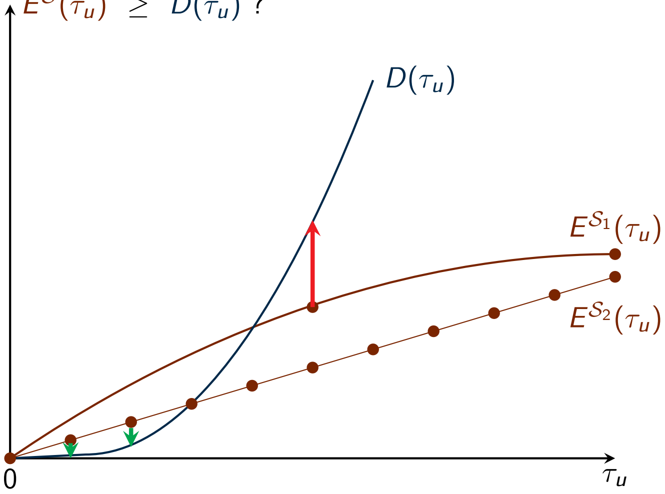

$E^{\mathcal{S}}(\tau_u) \geq D(\tau_u)$?
$D(\tau_u)$
$E^{\mathcal{S}_1}(\tau_u)$
0
τ_u


$E^{\mathcal{S}}(\tau_u) \geq D(\tau_u)$?
$D(\tau_u)$
$E^{\mathcal{S}_1}(\tau_u)$
0
τ_u

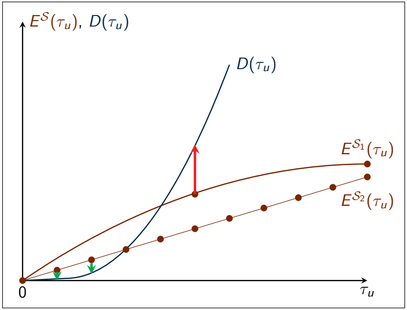


$E^{\mathcal{S}}(\tau_u) \geq D(\tau_u)$?
$D(\tau_u)$
$D(\tau_u)$
$E^{\mathcal{S}_1}(\tau_u)$
0
τ_u

$E^{\mathcal{S}}(\tau_u) \geq D(\tau_u)$?



Key properties of a solver

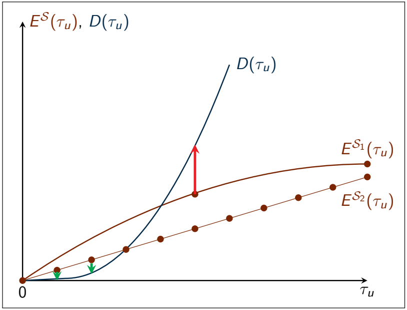


Keep in Mind

It is sometimes better to choose a **less efficient**[a] solver with **shorter preparation step** duration τ_1.

[a] per iteration

Key properties of a solver



Keep in Mind

It is sometimes better to choose a **less efficient**[a] solver with **shorter preparation step** duration τ_1.

[a] per iteration

Gradient-based studies

Bemporad and Patrinos (2012), Jones et al. (2012), MA (2013).

Heuristics for second order methods

Bock et al. SIAM (2007)

Comparing Algorithms: An Example

Houska et al. Automatica (2011)

Model

$$
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= u \qquad u \in [-1, +1] \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= -g \sin x_3 - u \cos x_3 - b x_4
\end{aligned}
$$

Comparing Algorithms: An Example

Houska et al. Automatica (2011)

Model

$$
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= u \qquad u \in [-1, +1] \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= -g \sin x_3 - u \cos x_3 - b x_4
\end{aligned}
$$

Cost

$$
\sum_{i=1}^{N_p} \|\mathbf{x}_{k+i}^T\|_Q^2 + \|\mathbf{u}_{k+i-1}^T\|_R^2
$$

Comparing Algorithms: An Example

Houska et al. Automatica (2011)

Model

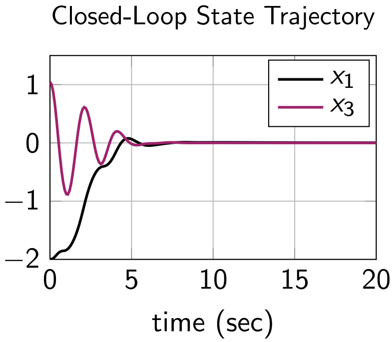
$$
\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= u \qquad u \in [-1, +1] \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= -g \sin x_3 - u \cos x_3 - b x_4
\end{aligned}
$$

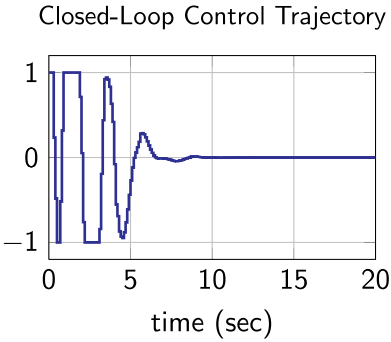
Cost

$$
\sum_{i=1}^{N_p} \|\mathbf{x}_{k+i}^T\|_Q^2 + \|\mathbf{u}_{k+i-1}^T\|_R^2
$$

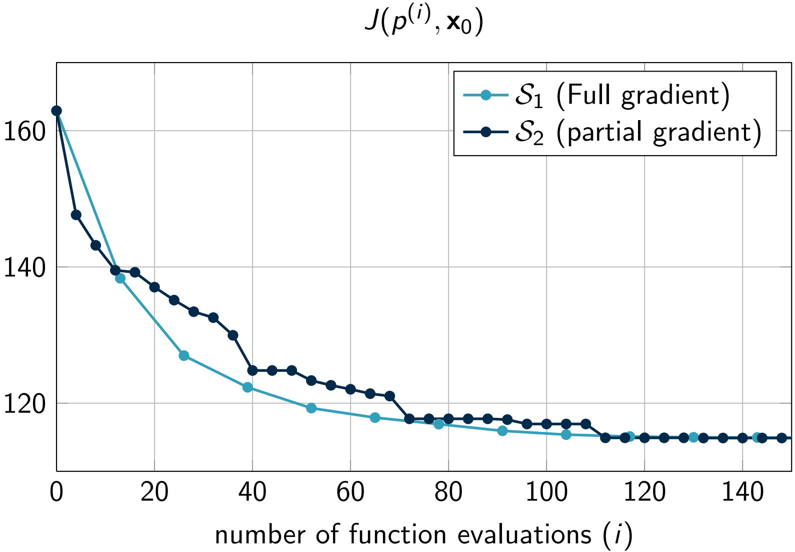
- $N_p = 10$ and $T_p = 3\ sec$
- Two solvers:
  - $\mathcal{S}_1$ Full gradient
  - $\mathcal{S}_2$ Partial gradient
- $\tau_1^{\mathcal{S}_2} = [\frac{4}{13}] \times \tau_1^{\mathcal{S}_1}$
- $\tau_u(\mathcal{S}_2) = [\frac{4}{13}] \times \tau_u(\mathcal{S}_1)$
- $\tau_u(\mathcal{S}_1) \in \{5, 50, 100\}\ ms$

Typical Closed-Loop Behavior

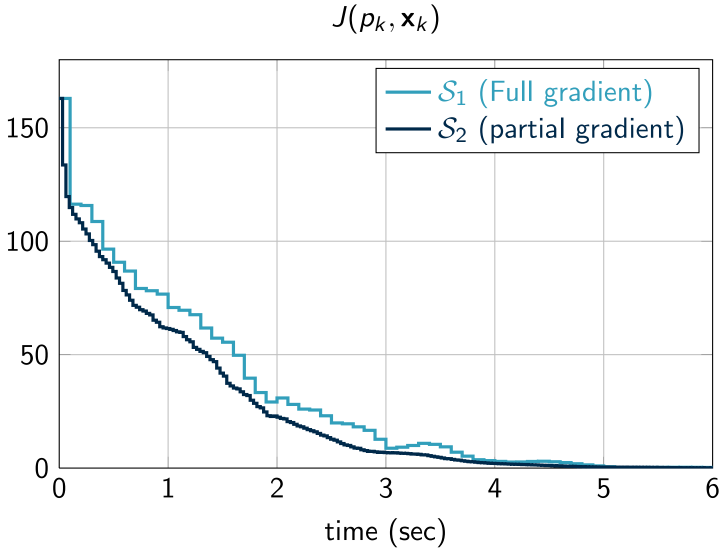




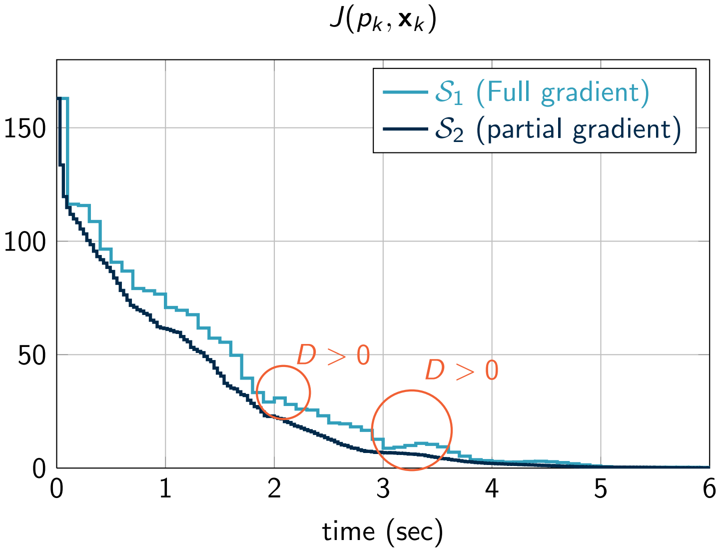
$\mathcal{S}_1$ is more efficient than $\mathcal{S}_2$ ... **per iteration** !



Closed-Loop Evolutions of the Predicted Cost



Closed-Loop Evolutions of the Predicted Cost



Comparison of Effective Closed-Loop Cost

$$J_{cl} := \frac{1}{N_{sim}} \sum_{k=1}^{N_{sim}} \left[\|\mathbf{x}_k\|_Q^2 + \|u_{k-1}\|_R^2 \right]$$

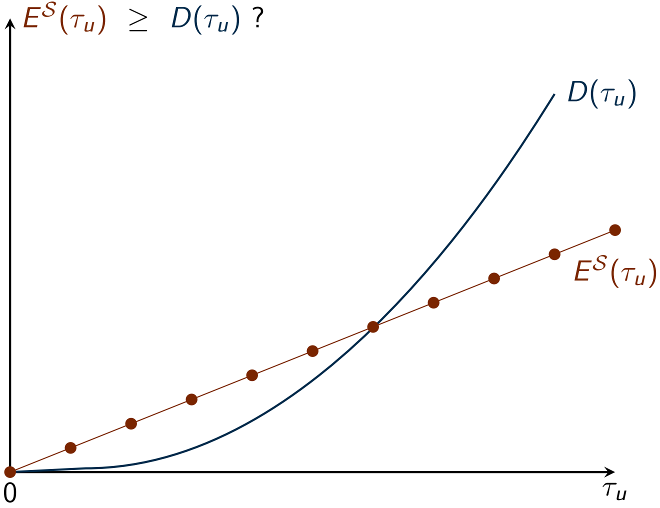
$\tau_u(\mathcal{S}_1)$	100 ms	50 ms	5 ms
Full Gradient	0.728	0.430	0.027
Partial Gradient	0.300	0.110	0.008
Gain %	59%	74%	70%

Updating Scheme For a Given Solver

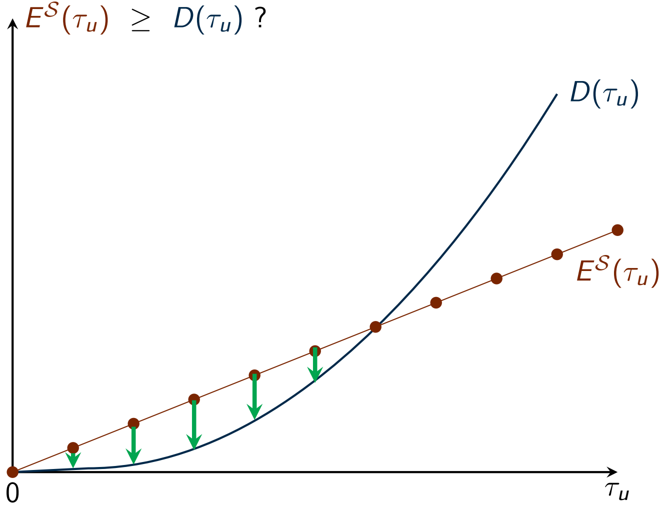
Assume that a solver $\mathcal{S}$ has been chosen . . .

Is there any remaining choice ?

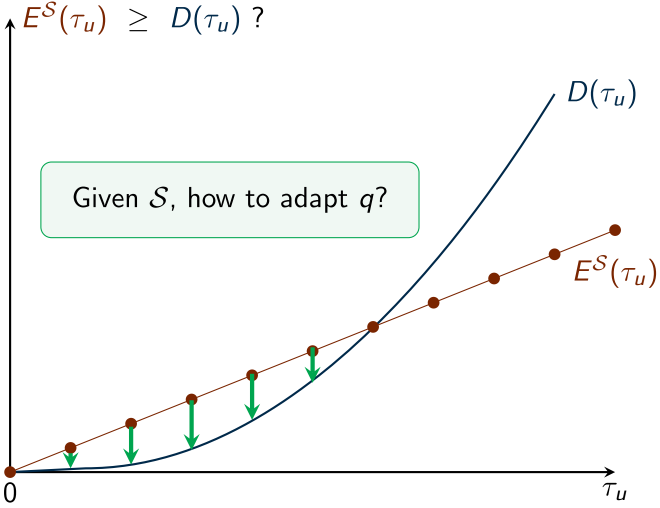
What is the optimal τ_u for a given solver?



What is the optimal τ_u for a given solver?



What is the optimal τ_u for a given solver?



Updating τ_u is a control problem . . . !

$$p_{k+1} \quad = \quad \mathcal{S}^{(q(\tau_u))}\left(p_k{}^{+}, \mathbf{x}_k\right)\Big)$$

Updating τ_u is a control problem ...!

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

Updating τ_u is a control problem . . . !

$$\begin{pmatrix} p_{k+1} \\ \mathbf{x}_{k+1} \end{pmatrix} = F\left(\begin{pmatrix} p_k \\ \mathbf{x}_k \end{pmatrix}, \tau_u\right)$$

Updating τ_u is a control problem ...!

$$\begin{pmatrix} p_{k+1} \\ \mathbf{x}_{k+1} \end{pmatrix} = F\left(\begin{pmatrix} p_k \\ \mathbf{x}_k \end{pmatrix}, \tau_u\right)$$

$$y = J(p_k, \mathbf{x}_k)$$

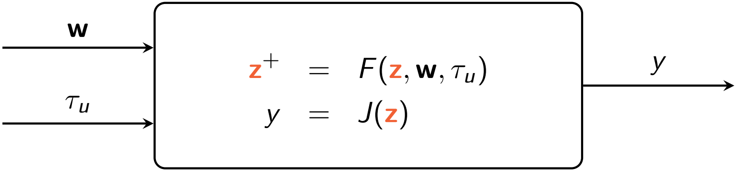
Updating τ_u is a control problem ...!

$$
\begin{aligned}
\mathbf{z}^{+} &= F(\mathbf{z}, \tau_u) \\
y &= J(\mathbf{z})
\end{aligned}
$$

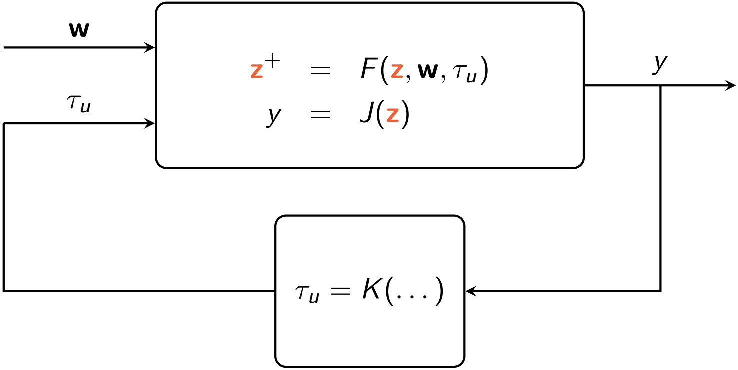
Updating τ_u is a control problem . . . !

$$\begin{aligned} \mathbf{z}^+ &= F(\mathbf{z}, \mathbf{w}, \tau_u) \\ y &= J(\mathbf{z}) \end{aligned}$$

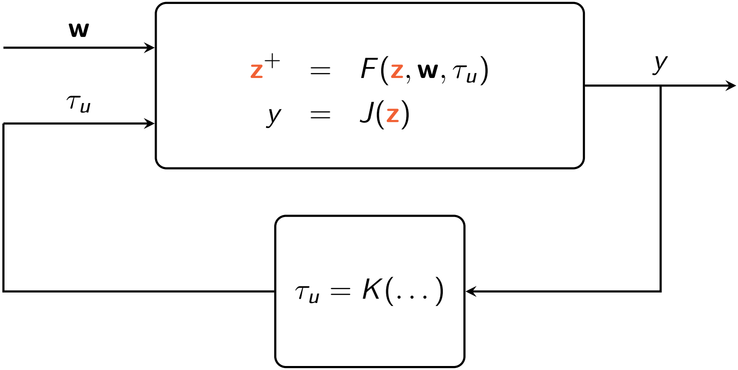
Updating τ_u is a control problem . . . !



Updating τ_u is a control problem ...!

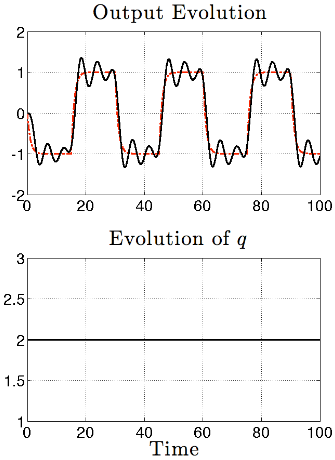


Updating τ_u is a control problem . . . !



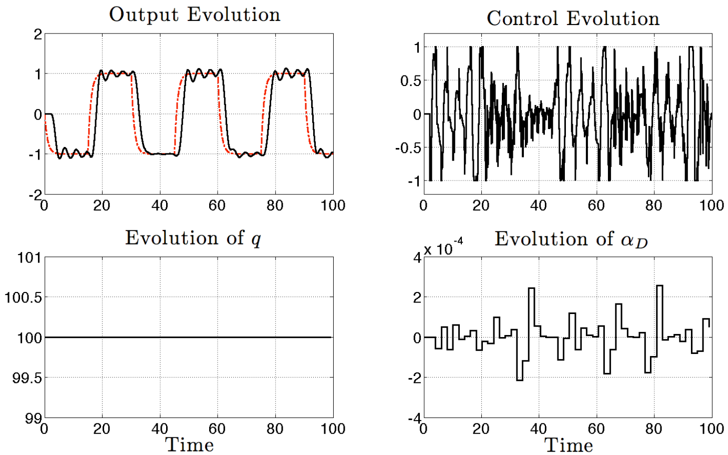
M.A. ECC (2013)

$q = 2$ without adaptation



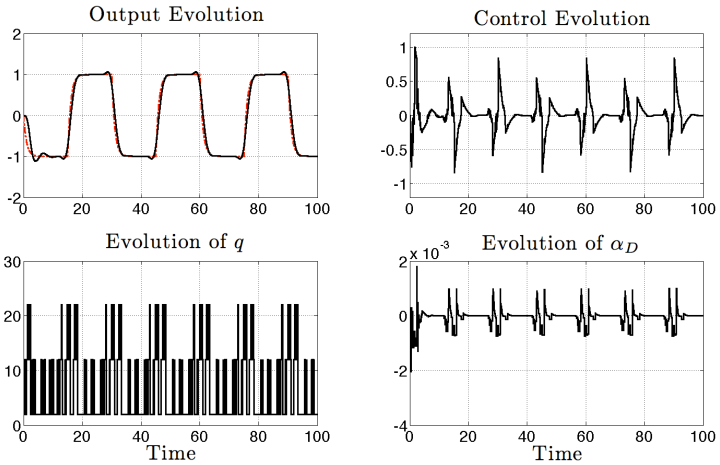
MA, ECC (2013)

$q = 100$ without adaptation



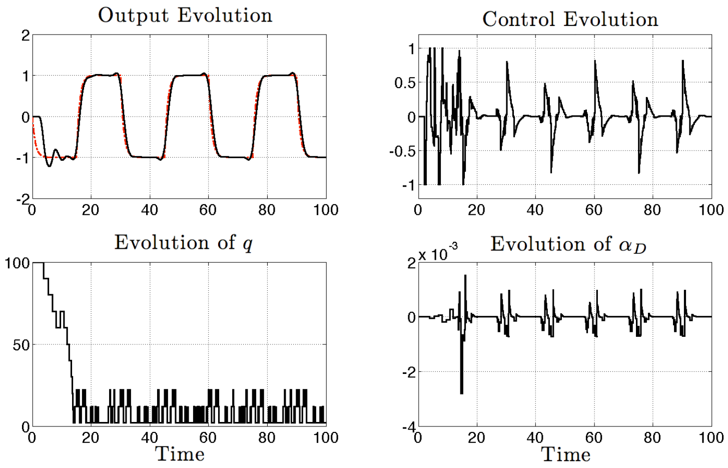
MA, ECC (2013)

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



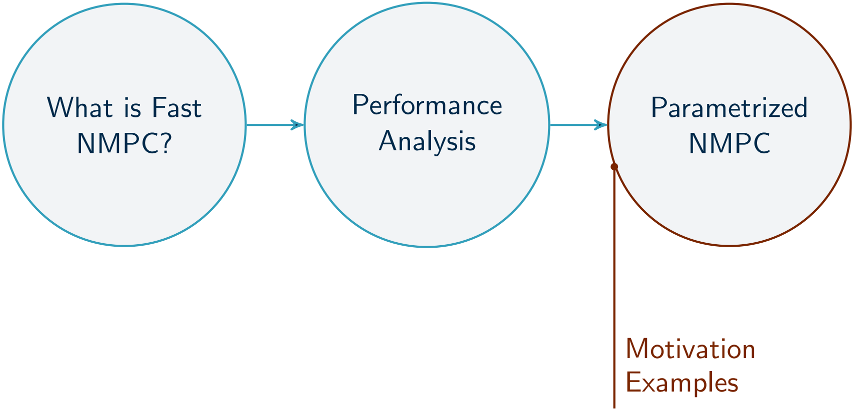
MA, ECC (2013)

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

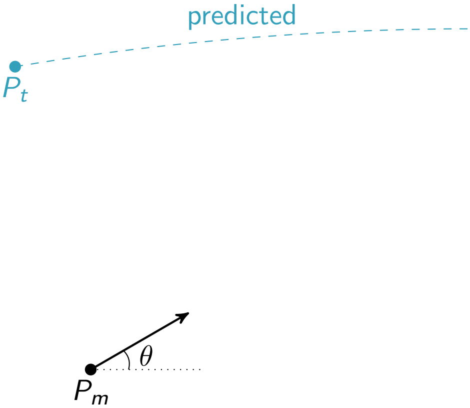


MA, ECC (2013)

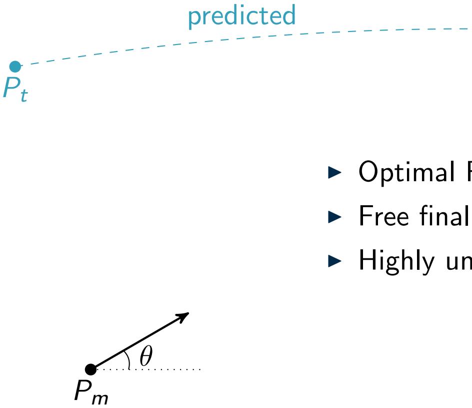
Outline


What is Fast NMPC?
Performance Analysis
Parametrized NMPC
Motivation
Examples

Introductory Example: Interception of moving target

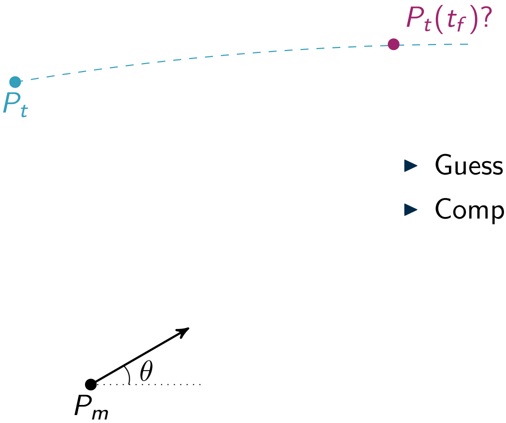


Introductory Example: Interception of moving target



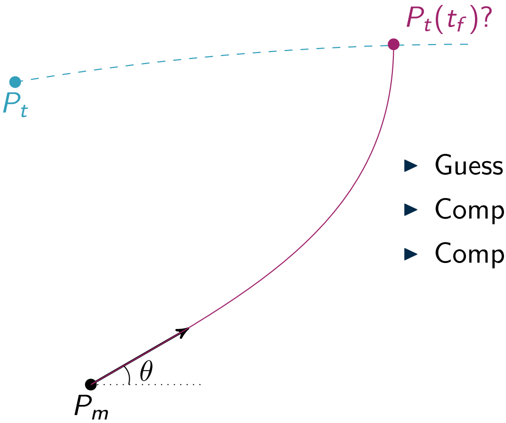
- Optimal Rendez-vous problem
- Free final time
- Highly uncertain

Introductory Example: Interception of moving target



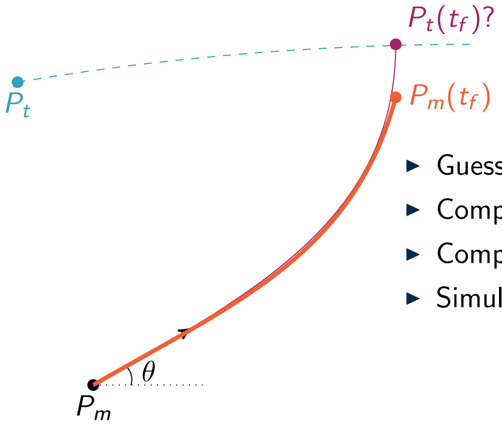
- Guess some t_f.
- Compute the corresponding $P_t(t_f)$

Introductory Example: Interception of moving target



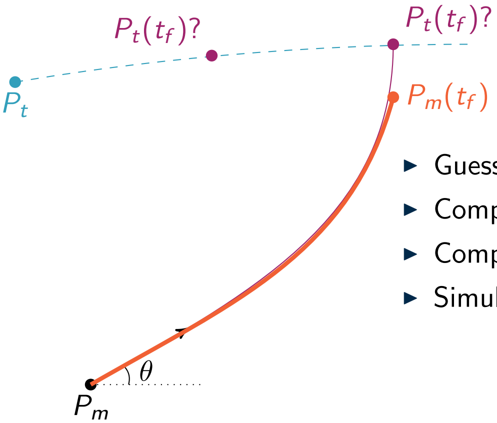
- Guess some t_f.
- Compute the corresponding $P_t(t_f)$
- Compute the parabola

Introductory Example: Interception of moving target



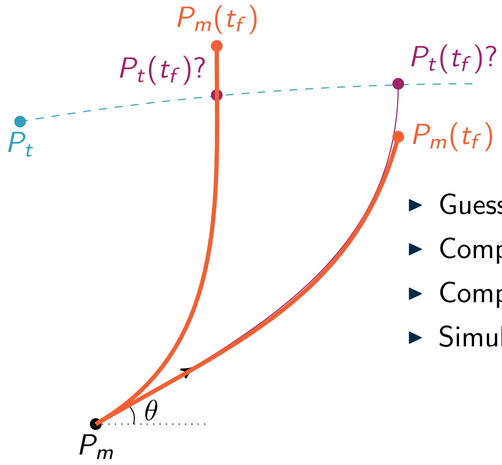
- Guess some t_f.
- Compute the corresponding $P_t(t_f)$
- Compute the parabola
- Simulate parabola tracking

Introductory Example: Interception of moving target



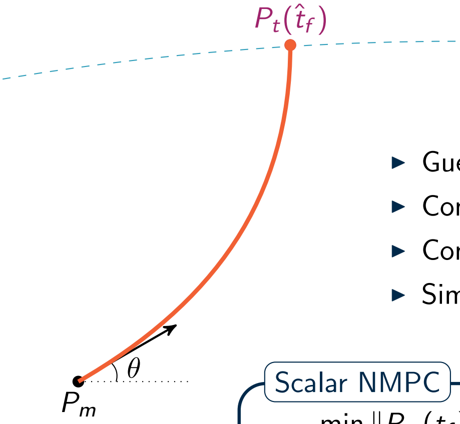
- Guess some t_f.
- Compute the corresponding $P_t(t_f)$
- Compute the parabola
- Simulate parabola tracking

Introductory Example: Interception of moving target



- Guess some t_f.
- Compute the corresponding $P_t(t_f)$
- Compute the parabola
- Simulate parabola tracking

Introductory Example: Interception of moving target

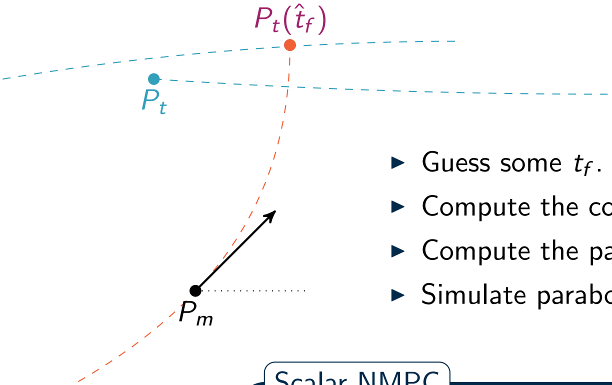


- Guess some t_f.
- Compute the corresponding $P_t(t_f)$
- Compute the parabola
- Simulate parabola tracking

Scalar NMPC

$$\min_{t_f} \|P_m(t_f) - P_t(t_f)\|$$

Introductory Example: Interception of moving target

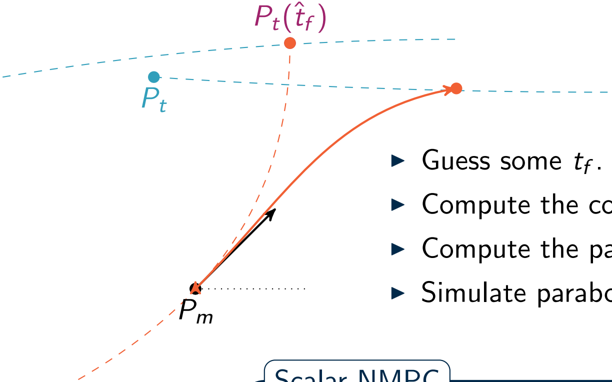


- Guess some t_f.
- Compute the corresponding $P_t(t_f)$
- Compute the parabola
- Simulate parabola tracking

Scalar NMPC

$$\min_{t_f} \|P_m(t_f) - P_t(t_f)\|$$

Introductory Example: Interception of moving target



- Guess some t_f.
- Compute the corresponding $P_t(t_f)$
- Compute the parabola
- Simulate parabola tracking

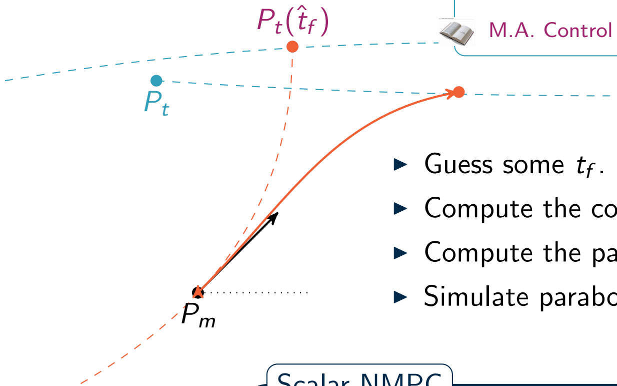
Scalar NMPC

$$\min_{t_f} \|P_m(t_f) - P_t(t_f)\|$$

Introductory Example: Interception of moving target

Full solution	40	ms
Fast solution	0.4	ms

M.A. Control Eng. Practice (2001)



- Guess some t_f.
- Compute the corresponding $P_t(t_f)$
- Compute the parabola
- Simulate parabola tracking

Scalar NMPC

$$\min_{t_f} \|P_m(t_f) - P_t(t_f)\|$$

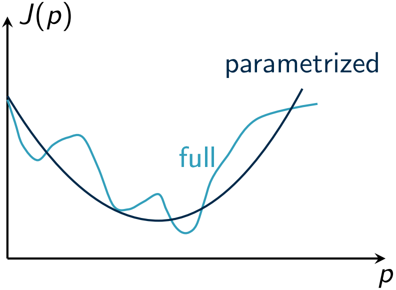
General Aspects of Parametrized NMPC

- Reduced number of parameters
- Problem conditioning
- Sub-optimality
- Problem dependent

1. Reduce τ_1 !
2. Qualify specific algorithms (Simplex, Torczon, etc.)
3. Easier certification

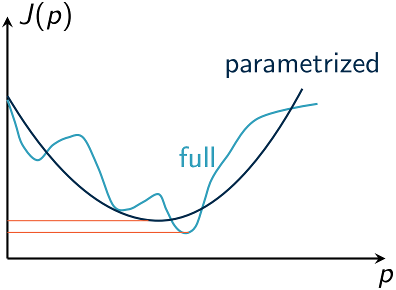
General Aspects of Parametrized NMPC

- Reduced number of parameters
- Problem conditioning
- Sub-optimality
- Problem dependent



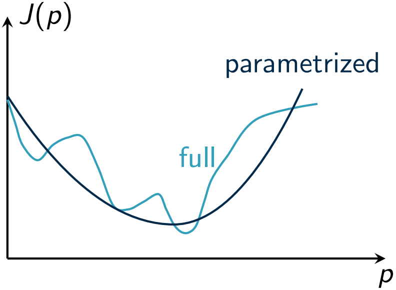
General Aspects of Parametrized NMPC

- Reduced number of parameters
- Problem conditioning
- Sub-optimality
- Problem dependent



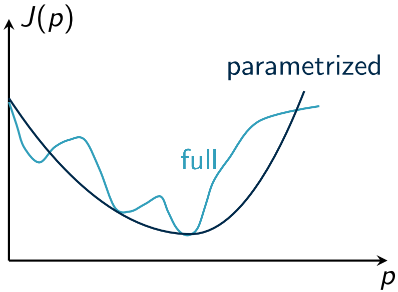
General Aspects of Parametrized NMPC

- Reduced number of parameters
- Problem conditioning
- Sub-optimality
- Problem dependent



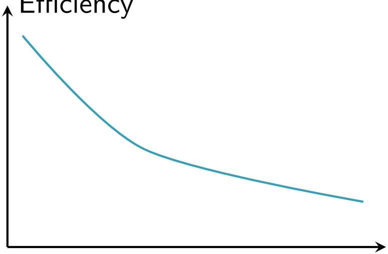
General Aspects of Parametrized NMPC

- Reduced number of parameters
- Problem conditioning
- Sub-optimality
- Problem dependent

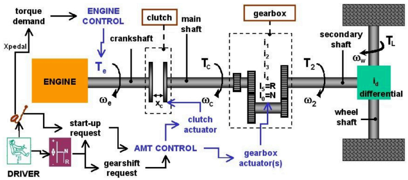


General Aspects of Parametrized NMPC

- Reduced number of parameters
- Problem conditioning
- Sub-optimality
- Problem dependent



Automated Manual Transmission



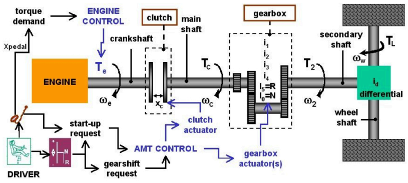
R. Amari

P. Tona (IFP)

Amari et al. IFAC World Congress (2008)

Automated Manual Transmission


R. Amari | P. Tona (IFP)



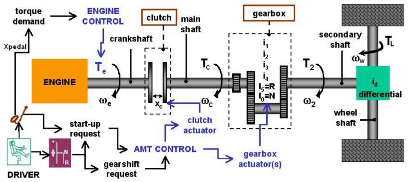
Control objective

- Smooth ($\omega_e - \omega_c \rightarrow 0$)
- Transparency (pedal $\rightarrow T_e$)

Amari et al. IFAC World Congress (2008)

Automated Manual Transmission

R. Amari P. Tona (IFP)



Control objective

- Smooth $(\omega_e - \omega_c \to 0)$
- Transparency (pedal $\to T_e$)

Constraints

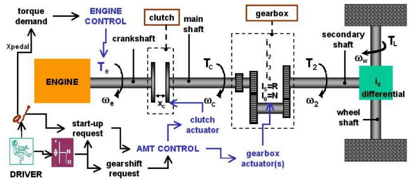
- Control level saturation
- Control derivative saturation
- Embedded control

Amari et al. IFAC World Congress (2008)

Automated Manual Transmission

R. Amari

P. Tona (IFP)



Control objective

- Smooth $(\omega_e - \omega_c \to 0)$
- Transparency (pedal $\to T_e$)

Constraints

- Control level saturation
- Control derivative saturation
- Embedded control

Solution

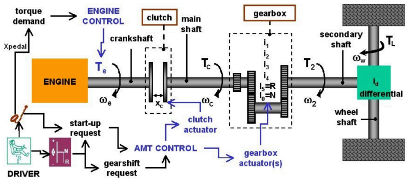
Replace NL groups by observed terms.

Parametrized MPC with $p = T_p$

Solve the NLP by dichotomy

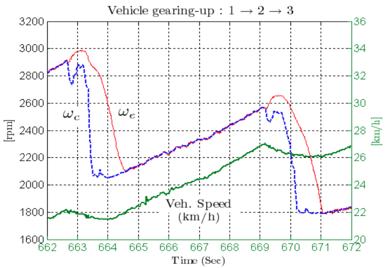
Amari et al. IFAC World Congress (2008)

Automated Manual Transmission



R. Amari

P. Tona (IFP)



Control objective

- Smooth ($\omega_e - \omega_c \to 0$)
- Transparency (pedal $\to T_e$)

Constraints

- Control level saturation
- Control derivative saturation
- Embedded control

Solution

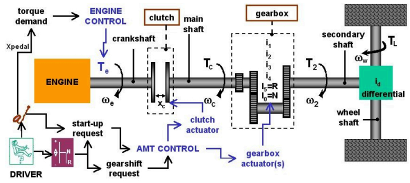
Replace NL groups by observed terms.

Parametrized MPC with $p = T_p$

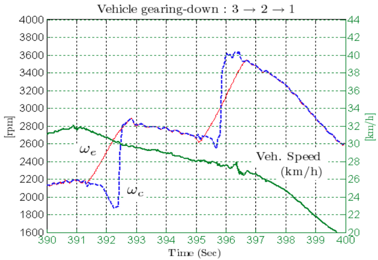
Solve the NLP by dichotomy

Amari et al. IFAC World Congress (2008)

Automated Manual Transmission




R. Amari P. Tona (IFP)



Control objective

- Smooth ($\omega_e - \omega_c \to 0$)
- Transparency (pedal $\to T_e$)

Constraints

- Control level saturation
- Control derivative saturation
- Embedded control

Solution

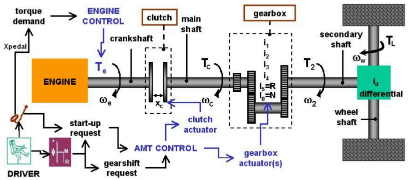
Replace NL groups by observed terms.

Parametrized MPC with $p = T_p$

Solve the NLP by dichotomy

Amari et al. IFAC World Congress (2008)

Automated Manual Transmission

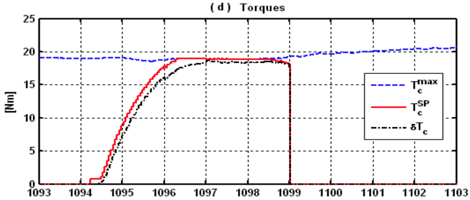


Control objective

- Smooth ($\omega_e - \omega_c \to 0$)
- Transparency (pedal $\to T_e$)

R. Amari

P. Tona (IFP)



Constraints

- Control level saturation
- Control derivative saturation
- Embedded control

Solution

Replace NL groups by observed terms.

Parametrized MPC with $p = T_p$

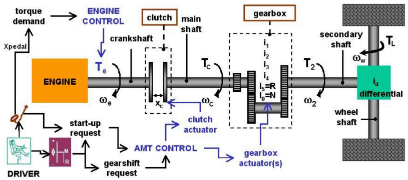
Solve the NLP by dichotomy

Amari et al. IFAC World Congress (2008)

Automated Manual Transmission

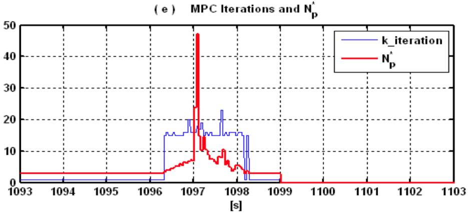
R. Amari

P. Tona (IFP)



Control objective

- Smooth ($\omega_e - \omega_c \to 0$)
- Transparency (pedal $\to T_e$)



Constraints

- Control level saturation
- Control derivative saturation
- Embedded control

Solution

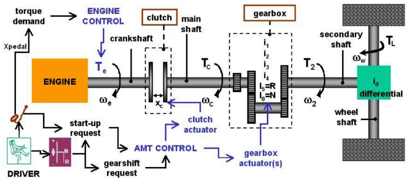
Replace NL groups by observed terms.

Parametrized MPC with $p = T_p$

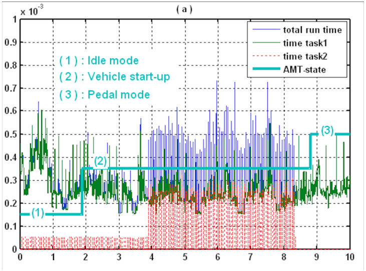
Solve the NLP by dichotomy

Amari et al. IFAC World Congress (2008)

Automated Manual Transmission




R. Amari P. Tona (IFP)



Control objective

- Smooth $(\omega_e - \omega_c \to 0)$
- Transparency (pedal $\to T_e$)

Constraints

- Control level saturation
- Control derivative saturation
- Embedded control

Solution

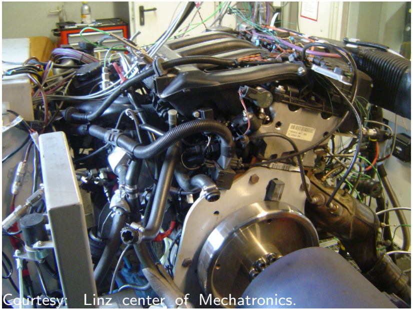
Replace NL groups by observed terms.

Parametrized MPC with $p = T_p$

Solve the NLP by dichotomy

Amari et al. IFAC World Congress (2008)

Diesel Engine AirPath Control



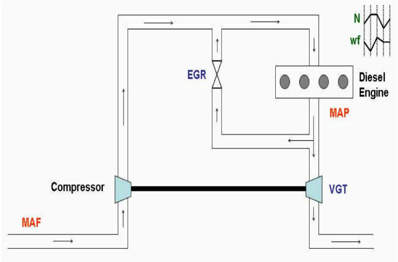
A. Murilo

P. Ortner (Linz)

R. Furhapter (Linz)

Murilo et al. Int. J. Control (2013)

Diesel Engine AirPath Control



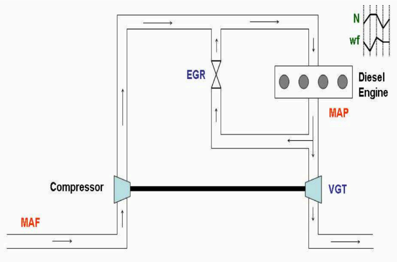
A. Murilo

P. Ortner (Linz)

R. Furhapter (Linz)

Murilo et al. Int. J. Control (2013)

Diesel Engine AirPath Control




A. Murilo


P. Ortner (Linz)


R. Furhapter (Linz)

Problem description

Nonlinear 8-dimensional model

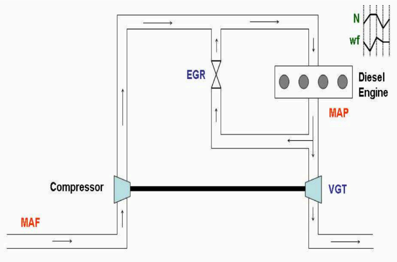
Non minimum-phase dynamic

Saturations on u and $\dot{u}$

Embedded Control

Murilo et al. Int. J. Control (2013)

Diesel Engine AirPath Control



A. Murilo

P. Ortner (Linz)

R. Furhapter (Linz)

Problem description

Nonlinear 8-dimensional model

Non minimum-phase dynamic

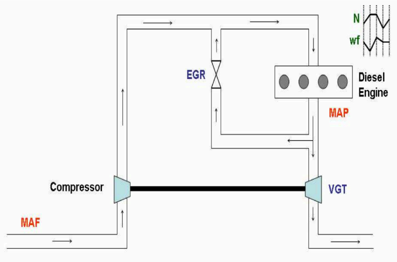
Saturations on u and $\dot{u}$

Embedded Control

$$\mathbf{u}(t) = Sat_{u_{min}}^{u_{max}}\Big(u^* + \alpha_1 e^{-\lambda_1 t} + \alpha_2 e^{-\lambda_2 t}\Big)$$

Murilo et al. Int. J. Control (2013)

Diesel Engine AirPath Control




A. Murilo


P. Ortner (Linz)


R. Furhapter (Linz)

Problem description

Nonlinear 8-dimensional model

Non minimum-phase dynamic

Saturations on u and $\dot{u}$

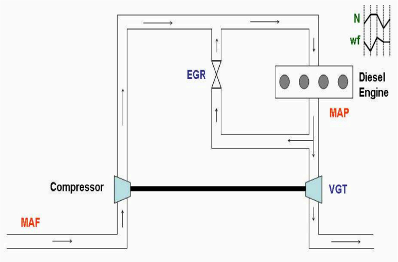
Embedded Control

$$\mathbf{u}(t) = Sat_{u_{min}}^{u_{max}}\Big(u^* + \alpha_1 e^{-\lambda_1 t} + \alpha_2 e^{-\lambda_2 t}\Big)$$

$$u^* + \alpha_1 + \alpha_2 = u_{k-1} \quad \text{(continuity)}$$

Murilo et al. Int. J. Control (2013)

Diesel Engine AirPath Control



A. Murilo

P. Ortner (Linz)

R. Furhapter (Linz)

Problem description

Nonlinear 8-dimensional model

Non minimum-phase dynamic

Saturations on u and $\dot{u}$

Embedded Control

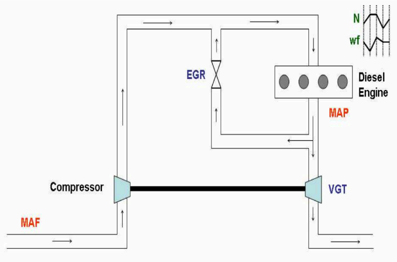
$$\mathbf{u}(t) = Sat_{u_{min}}^{u_{max}}\left(u^* + \alpha_1 e^{-\lambda_1 t} + \alpha_2 e^{-\lambda_2 t}\right)$$

$$u^* + \alpha_1 + \alpha_2 = u_{k-1} \text{ (continuity)}$$

$$-\lambda_1\alpha_1 - \lambda_2\alpha_2 = p \times \dot{u}^{max}, \quad p \in [-1, +1]^2$$

Murilo et al. Int. J. Control (2013)

Diesel Engine AirPath Control



A. Murilo

P. Ortner (Linz)

R. Furhapter (Linz)

Problem description

Nonlinear 8-dimensional model

Non minimum-phase dynamic

Saturations on u and $\dot{u}$

Embedded Control

$$\mathbf{u}(t) = Sat_{u_{min}}^{u_{max}}\left(u^* + \alpha_1 e^{-\lambda_1 t} + \alpha_2 e^{-\lambda_2 t}\right)$$

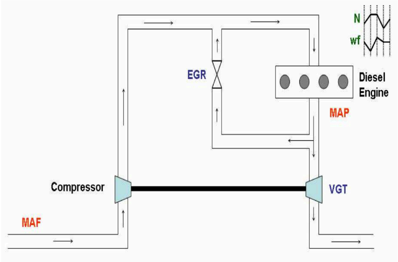
$$u^* + \alpha_1 + \alpha_2 = u_{k-1} \text{ (continuity)}$$

$$-\lambda_1\alpha_1 - \lambda_2\alpha_2 = p \times \dot{u}^{max}, \quad p \in [-1, +1]^2$$

$$\Rightarrow (\alpha_1, \alpha_2) := F(u_{k-1}, p)$$

Murilo et al. Int. J. Control (2013)

Diesel Engine AirPath Control



A. Murilo

P. Ortner (Linz)

R. Furhapter (Linz)

Problem description

Nonlinear 8-dimensional model

Non minimum-phase dynamic

Saturations on u and $\dot{u}$

Embedded Control

$$\mathbf{u}(t) = Sat_{u_{min}}^{u_{max}}\Big(u^* + \alpha_1 e^{-\lambda_1 t} + \alpha_2 e^{-\lambda_2 t}\Big)$$

$u^* + \alpha_1 + \alpha_2 = u_{k-1}$ (continuity)

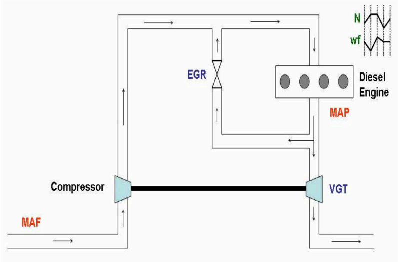
$-\lambda_1\alpha_1 - \lambda_2\alpha_2 = p \times \dot{u}^{max}$, $p \in [-1,+1]^2$

$\Rightarrow (\alpha_1, \alpha_2) := F(u_{k-1}, p)$

2D box optimization on $[-1,+1]^2$.

Murilo et al. Int. J. Control (2013)

Diesel Engine AirPath Control





A. Murilo P. Ortner (Linz) R. Furhapter (Linz)

Parameters of the solution

Model sampling period	50 ms
Prediction horizon	$30 \times \tau$
SQP / Trust region	
Number of function eval.	30

Problem description

Nonlinear 8-dimensional model

Non minimum-phase dynamic

Saturations on u and $\dot{u}$

Embedded Control

$$\mathbf{u}(t) = Sat_{u_{min}}^{u_{max}}\left(u^* + \alpha_1 e^{-\lambda_1 t} + \alpha_2 e^{-\lambda_2 t}\right)$$

$u^* + \alpha_1 + \alpha_2 = u_{k-1}$ (continuity)

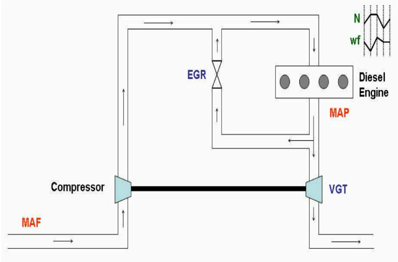
$-\lambda_1\alpha_1 - \lambda_2\alpha_2 = p \times \dot{u}^{max}$, $p \in [-1, +1]^2$

$\Rightarrow (\alpha_1, \alpha_2) := F(u_{k-1}, p)$

2D box optimization on $[-1, +1]^2$.

Murilo et al. Int. J. Control (2013)

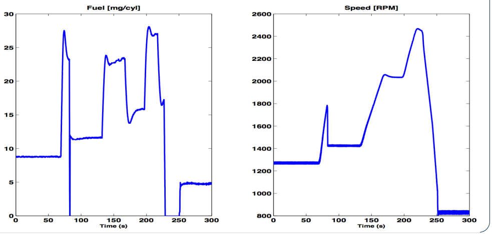
Diesel Engine AirPath Control



A. Murilo

P. Ortner (Linz)

R. Furhapter (Linz)



Problem description

Nonlinear 8-dimensional model

Non minimum-phase dynamic

Saturations on u and $\dot{u}$

Embedded Control

$$\mathbf{u}(t) = Sat_{u_{min}}^{u_{max}}\left(u^* + \alpha_1 e^{-\lambda_1 t} + \alpha_2 e^{-\lambda_2 t}\right)$$

$u^* + \alpha_1 + \alpha_2 = u_{k-1}$ (continuity)

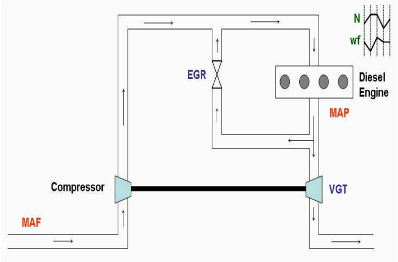
$-\lambda_1\alpha_1 - \lambda_2\alpha_2 = p \times \dot{u}^{max}$, $p \in [-1, +1]^2$

$\Rightarrow (\alpha_1, \alpha_2) := F(u_{k-1}, p)$

2D box optimization on $[-1, +1]^2$.

Murilo et al. Int. J. Control (2013)

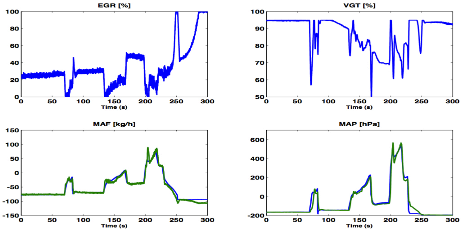
Diesel Engine AirPath Control



A. Murilo

P. Ortner (Linz)

R. Furhapter (Linz)



Problem description

Nonlinear 8-dimensional model

Non minimum-phase dynamic

Saturations on u and $\dot{u}$

Embedded Control

$$\mathbf{u}(t) = Sat_{u_{min}}^{u_{max}}\Big(u^* + \alpha_1 e^{-\lambda_1 t} + \alpha_2 e^{-\lambda_2 t}\Big)$$

$u^* + \alpha_1 + \alpha_2 = u_{k-1}$ (continuity)

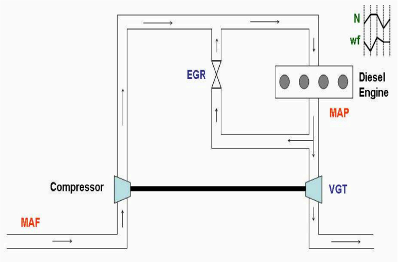
$-\lambda_1\alpha_1 - \lambda_2\alpha_2 = p \times \dot{u}^{max}$, $p \in [-1, +1]^2$

$\Rightarrow (\alpha_1, \alpha_2) := F(u_{k-1}, p)$

2D box optimization on $[-1, +1]^2$.

Murilo et al. Int. J. Control (2013)

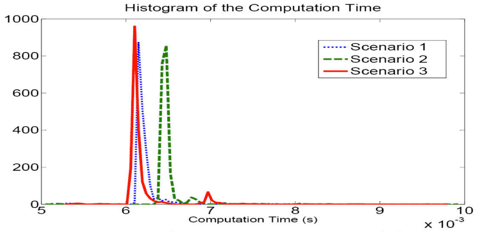
Diesel Engine AirPath Control



A. Murilo

P. Ortner (Linz)

R. Furhapter (Linz)



Problem description

Nonlinear 8-dimensional model

Non minimum-phase dynamic

Saturations on u and $\dot{u}$

Embedded Control

$$\mathbf{u}(t) = Sat_{u_{min}}^{u_{max}}\left(u^* + \alpha_1 e^{-\lambda_1 t} + \alpha_2 e^{-\lambda_2 t}\right)$$

$u^* + \alpha_1 + \alpha_2 = u_{k-1}$ (continuity)

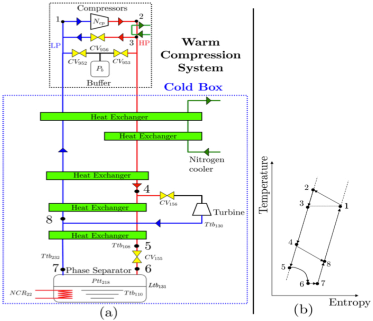
$-\lambda_1\alpha_1 - \lambda_2\alpha_2 = p \times \dot{u}^{max}$, $p \in [-1, +1]^2$

$\Rightarrow (\alpha_1, \alpha_2) := F(u_{k-1}, p)$

2D box optimization on $[-1, +1]^2$.

Murilo et al. Int. J. Control (2013)

Control of a Cryogenic System

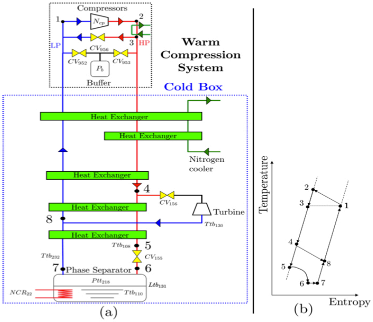


F. Bonne (CEA)

P. Bonnay (CEA)

Bonne et al. J. Process Control (2014)

Control of a Cryogenic System



F. Bonne (CEA)

P. Bonnay (CEA)

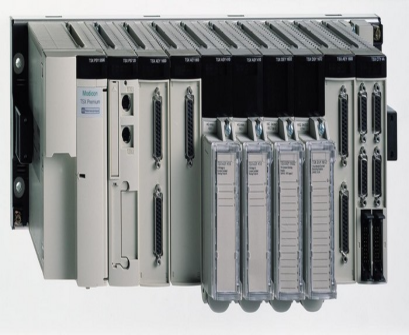
Despite the context, hard limitations:

- Computation power
- Memory use

(Schneider TSX Premium automata)

Bonne et al. J. Process Control (2014)

Control of a Cryogenic System



F. Bonne (CEA)

P. Bonnay (CEA)

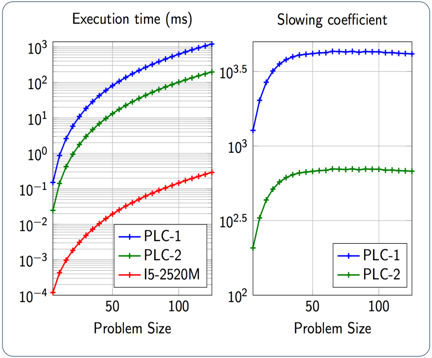
Despite the context, hard limitations:

- Computation power
- Memory use

(Schneider TSX Premium automata)

Bonne et al. J. Process Control (2014)

Control of a Cryogenic System



F. Bonne (CEA)

P. Bonnay (CEA)

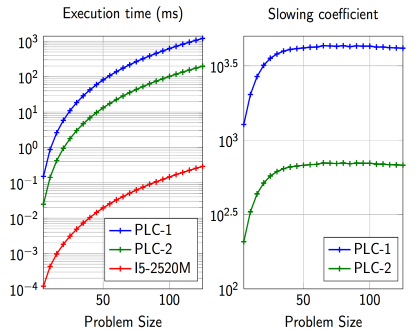
Despite the context, hard limitations:

- Computation power
- Memory use

(Schneider TSX Premium automata)

Bonne et al. J. Process Control (2014)

Control of a Cryogenic System



F. Bonne (CEA)

P. Bonnay (CEA)

Despite the context, hard limitations:

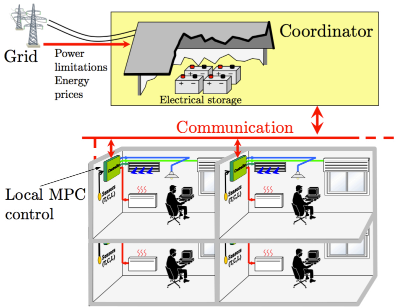
- Computation power
- Memory use

(Schneider TSX Premium automata)

⇒ This was a fast NMPC problem !!

Bonne et al. J. Process Control (2014)

Building Management Systems

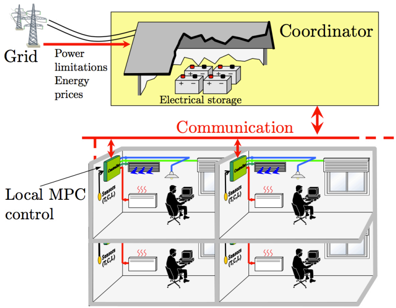


PhD Lamoudi (2012).

M. Y. Lamoudi (Schneider Electric Industry)

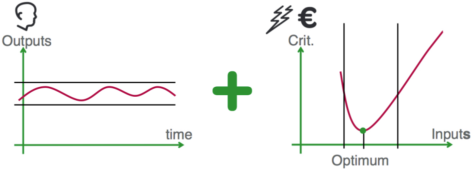
Lamoudi et al. in *distributed MPC made easy* Springer (2014)

Building Management Systems



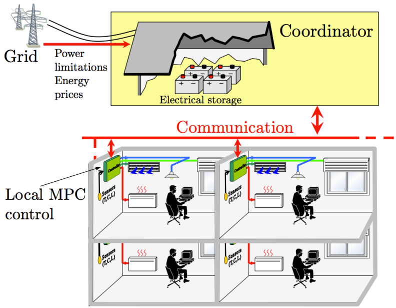
PhD Lamoudi (2012).

M. Y. Lamoudi (Schneider Electric Industry)



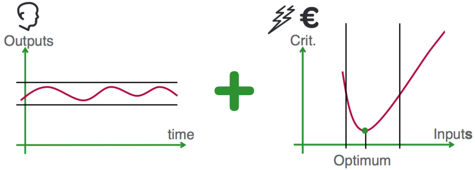
Lamoudi et al. in *distributed MPC made easy* Springer (2014)

Building Management Systems



PhD Lamoudi (2012).

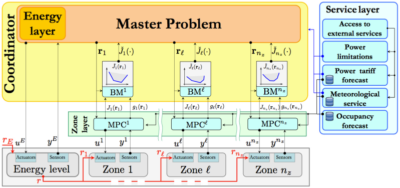
M. Y. Lamoudi (Schneider Electric Industry)



- Constrained nonlinear NLP.
- Solve using fixed-point iterations.

Lamoudi et al. in *distributed MPC made easy* Springer (2014)

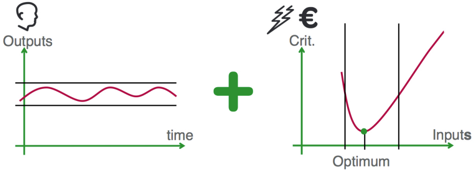
Building Management Systems



M. Y. Lamoudi (Schneider Electric Industry)

Need fast local solution in order to

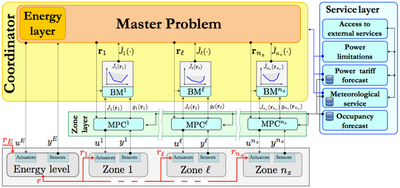
1. Enable Master iterations
2. Perform yearly simulations



- Constrained nonlinear NLP.
- Solve using fixed-point iterations.

Lamoudi et al. in *distributed MPC made easy* Springer (2014)

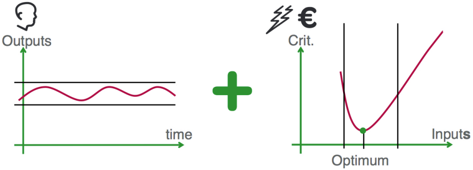
Building Management Systems



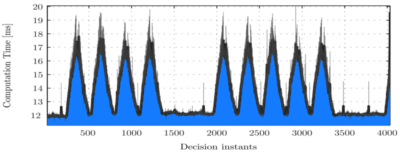
M. Y. Lamoudi (Schneider Electric Industry)

Need fast local solution in order to

1. Enable Master iterations
2. Perform yearly simulations



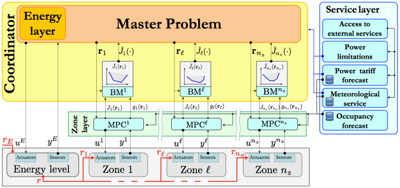
- Constrained nonlinear NLP.
- Solve using fixed-point iterations.



Lamoudi et al. in *distributed MPC made easy* Springer (2014)

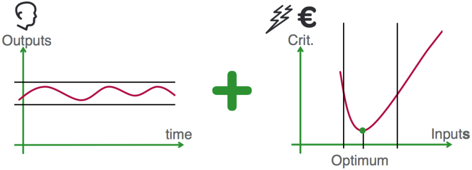
Building Management Systems

M. Y. Lamoudi (Schneider Electric Industry)

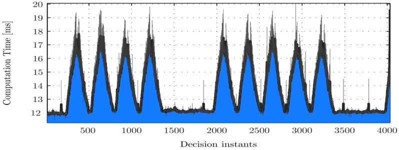


Need fast local solution in order to

1. Enable Master iterations
2. Perform yearly simulations



- Constrained nonlinear NLP.
- Solve using fixed-point iterations.



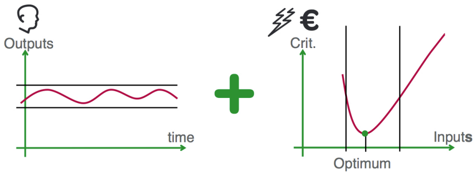
Data for 20 zones, $\tau_u = 5$ min

- 2×10^6 optimization pb
- 900 dof, 1000 constraints
- Simulation time = 18 h.

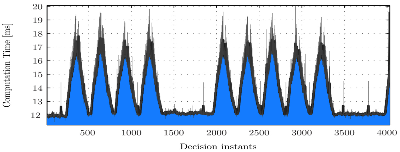
Lamoudi et al. in *distributed MPC made easy* Springer (2014)

Building Management Systems

M. Y. Lamoudi (Schneider Electric Industry)



- Constrained nonlinear NLP.
- Solve using fixed-point iterations.

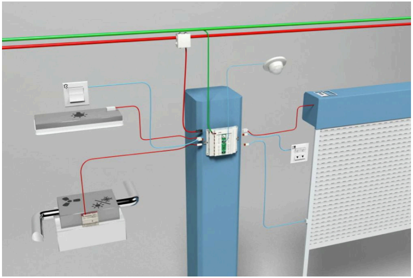


Data for 20 zones, $\tau_u = 5$ min

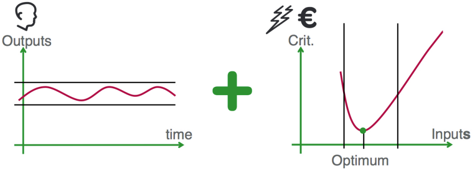
- 2×10^6 optimization pb
- 900 dof, 1000 constraints
- Simulation time = 18 h.

Lamoudi et al. in *distributed MPC made easy* Springer (2014)

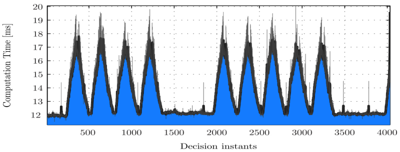
Building Management Systems



M. Y. Lamoudi (Schneider Electric Industry)



- Constrained nonlinear NLP.
- Solve using fixed-point iterations.

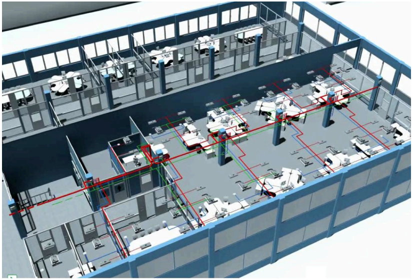


Data for 20 zones, $\tau_u = 5$ min

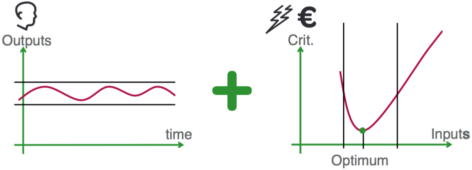
- 2×10^6 optimization pb
- 900 dof, 1000 constraints
- Simulation time = 18 h.

Lamoudi et al. in *distributed MPC made easy* Springer (2014)

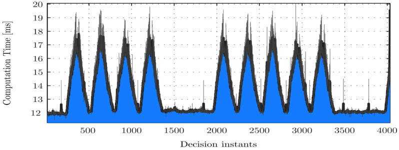
Building Management Systems



M. Y. Lamoudi (Schneider Electric Industry)



- Constrained nonlinear NLP.
- Solve using fixed-point iterations.



Data for 20 zones, $\tau_u = 5$ min

- 2×10^6 optimization pb
- 900 dof, 1000 constraints
- Simulation time = 18 h.

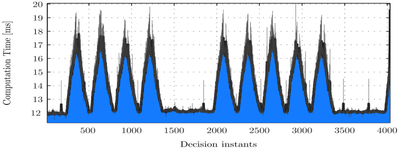
Lamoudi et al. in *distributed MPC made easy* Springer (2014)

Building Management Systems

M. Y. Lamoudi (Schneider Electric Industry)

Room-Box data

- Prediction horizon = 12h
- 8.2% memory usage
- 6 sec to solve the NLP.



Data for 20 zones, $\tau_u = 5$ min

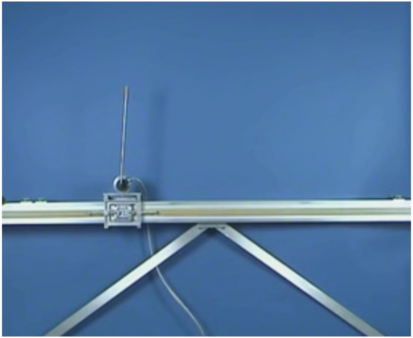
- 2×10^6 optimization pb
- 900 dof, 1000 constraints
- Simulation time = 18 h.

Lamoudi et al. in *distributed MPC made easy* Springer (2014)

Swing-up of a Twin Pendulum System


A. Murilo


G. Buche



- scalar NMPC
- 5 system integration / iteration
- control updating period 20 ms.

M.A. and A. Murilo, Automatica (2008)

Conclusion

Good News

- Huge progress
- Free dedicated software
- MPC-reflex in expansion
- Promising future

Keep in Mind

- MPC needed for transient
- Our proofs are tautologies
- Heavy use of heuristics
- Keep feet on earth