# On Trade-offs Governing Real-Time Implementation of Model Predictive Control

Mazen Alamir

CNRS, University of Grenoble

# Overview of the talk

Cryogenic refrigerators

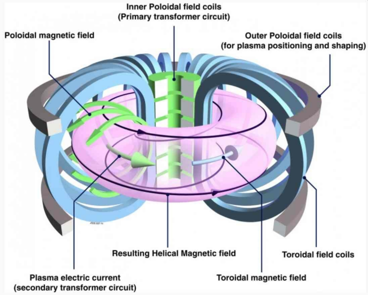
Ideal MPC

Real-Time MPC

Trade-offs

MPC certification

## The illustrative Process: Cryogenic Refrigerators

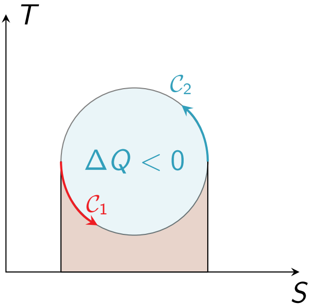



Source: https://www.euro-fusion.org

### Why?

Provide refrigeration capacity to cool down the supra-conducting coils used to accelerate the plasma in Nuclear Fusion Reactors (ITER, JT60)

# The illustrative Process: Cryogenic Refrigerators

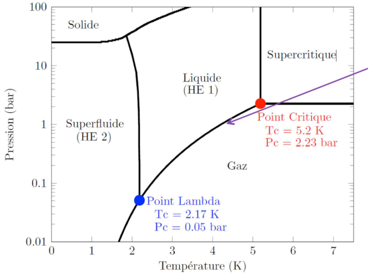



## How?

Force a thermodynamic fluid to make a counter-clock cycle in the $(S, T)$=(Entropy,Temperature) plan.

$$\int dQ = \underbrace{\int_{\mathcal{C}_1} TdS}_{>0} + \underbrace{\int_{\mathcal{C}_2} TdS}_{<<0}$$

# The illustrative Process: Cryogenic Refrigerators



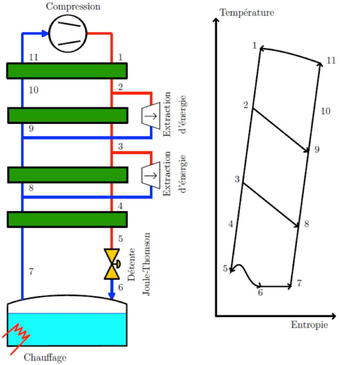

Source: F. Bonne PhD (2014).

## How?

Force a thermodynamic fluid to make a counter-clock cycle in the $(S, T)$=(Entropy,Temperature) plan.

$$\int dQ = \underbrace{\int_{\mathcal{C}_1} TdS}_{>0} + \underbrace{\int_{\mathcal{C}_2} TdS}_{<<0}$$

# The illustrative Process: Cryogenic Refrigerators



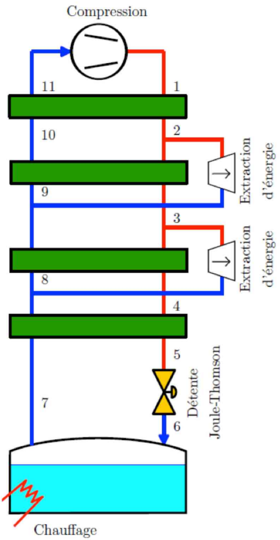

Source: F. Bonne PhD (2014).

## How?

Force a thermodynamic fluid to make a counter-clock cycle in the $(S,T)$=(Entropy,Temperature) plan.

$$\int dQ = \underbrace{\int_{\mathcal{C}_1} TdS}_{>0} + \underbrace{\int_{\mathcal{C}_2} TdS}_{<<0}$$

## The illustrative Process: Cryogenic Refrigerators




Source: F. Bonne PhD (2014).

### How?

Force a thermodynamic fluid to make a counter-clock cycle in the $(S, T)$=(Entropy,Temperature) plan.

$$\int dQ = \underbrace{\int_{\mathcal{C}_1} TdS}_{>0} + \underbrace{\int_{\mathcal{C}_2} TdS}_{<<0}$$

## The illustrative Process: Cryogenic Refrigerators

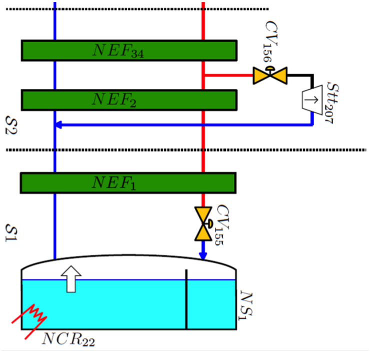

Source: F. Bonne PhD (2014).

### How?

Force a thermodynamic fluid to make a counter-clock cycle in the $(S, T)$=(Entropy,Temperature) plan.

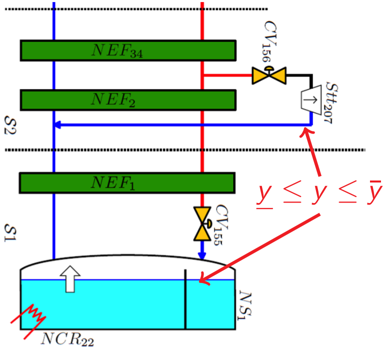
$$\int dQ = \underbrace{\int_{\mathcal{C}_1} TdS}_{>0} + \underbrace{\int_{\mathcal{C}_2} TdS}_{<<0}$$

# The illustrative Process: Cryogenic Refrigerators




**Why MPC?**

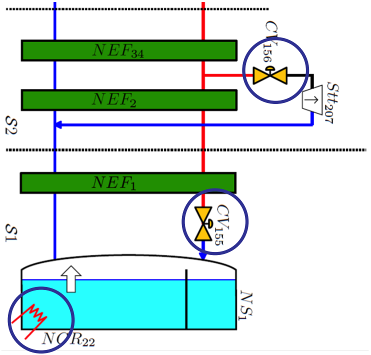
## The illustrative Process: Cryogenic Refrigerators




### Why MPC?

- State constraints

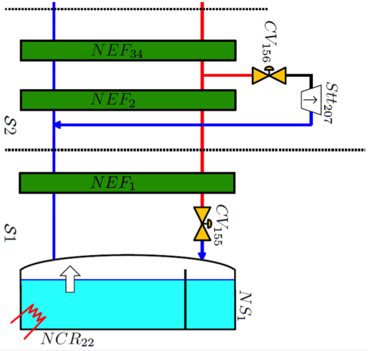
# The illustrative Process: Cryogenic Refrigerators




## Why MPC?

- State constraints
- Control Saturation

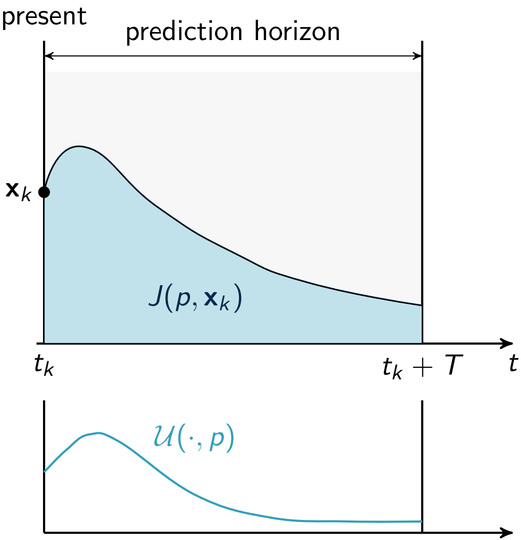
# The illustrative Process: Cryogenic Refrigerators



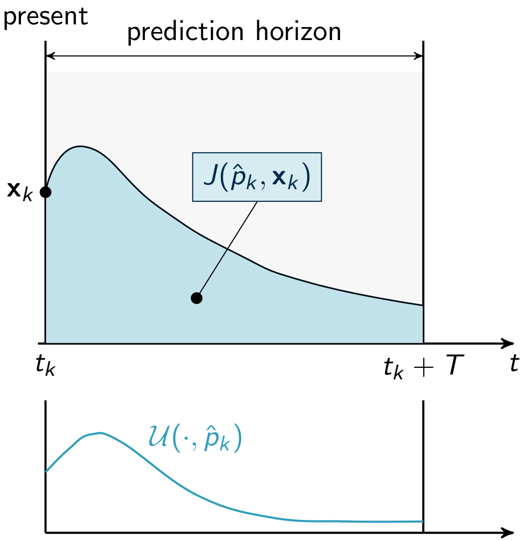

## Why MPC?

- State constraints
- Control Saturation
- Coupled dynamics
- Inverse response

## Ideal Framework: Recalls & Basic Notations



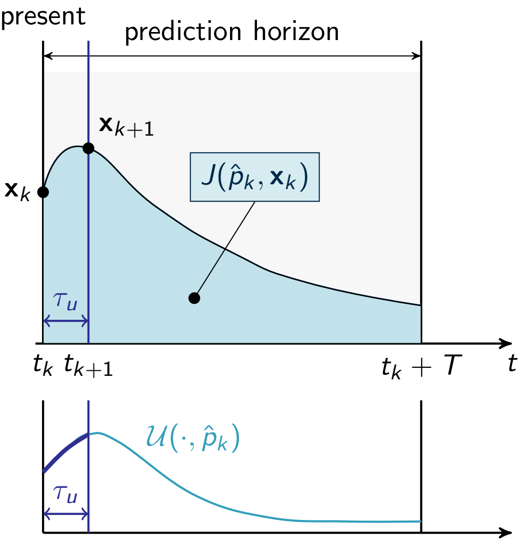
## Ideal Framework: Recalls & Basic Notations




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

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

## Ideal Framework: Recalls & Basic Notations




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

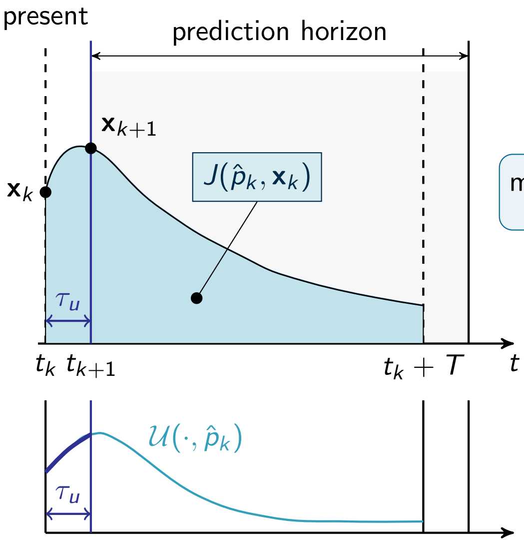
↓

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

↓

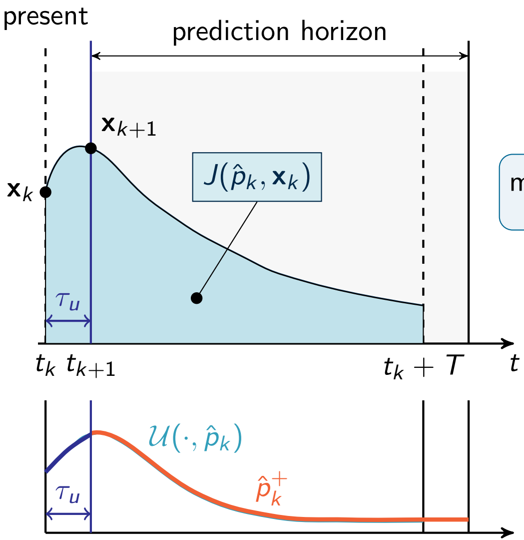
Apply $\mathcal{U}(\cdot, \hat{p}(\mathbf{x}_k))$ during $\tau_u$

# Ideal Framework: Recalls & Basic Notations



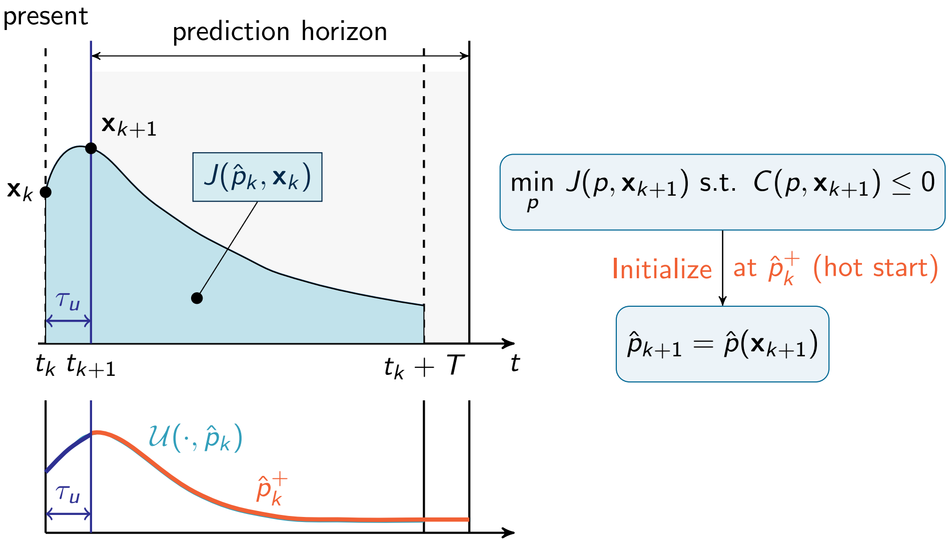

$$\min_p J(p, \mathbf{x}_{k+1}) \text{ s.t. } C(p, \mathbf{x}_{k+1}) \leq 0$$

## Ideal Framework: Recalls & Basic Notations

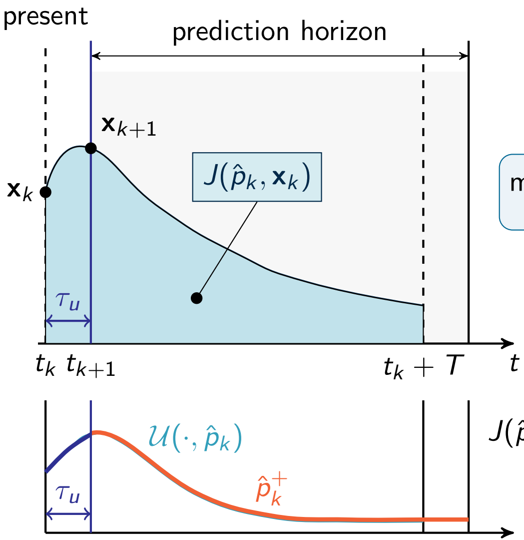


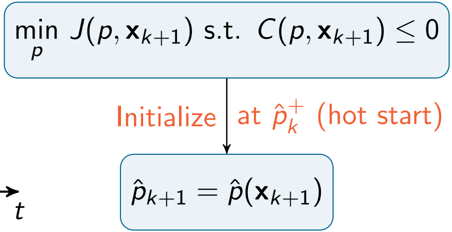

$$\min_{p} J(p, \mathbf{x}_{k+1}) \text{ s.t. } C(p, \mathbf{x}_{k+1}) \leq 0$$

# Ideal Framework: Recalls & Basic Notations



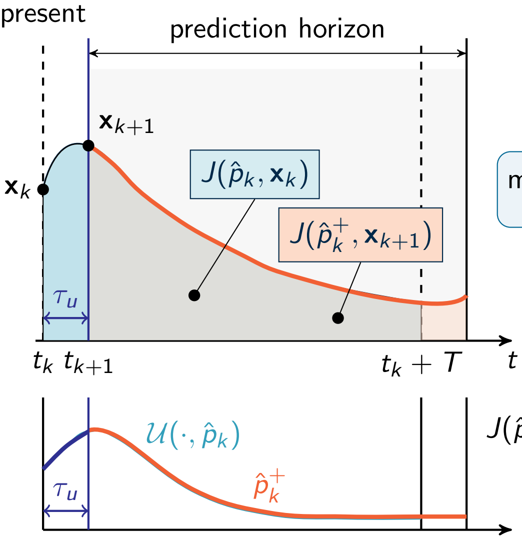
# Ideal Framework: Recalls & Basic Notations




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

# Ideal Framework: Recalls & Basic Notations



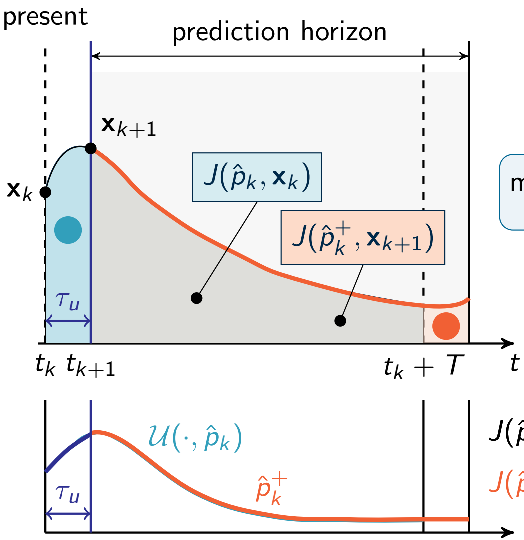

$$\min_p J(p, \mathbf{x}_{k+1}) \text{ s.t. } C(p, \mathbf{x}_{k+1}) \leq 0$$

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




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

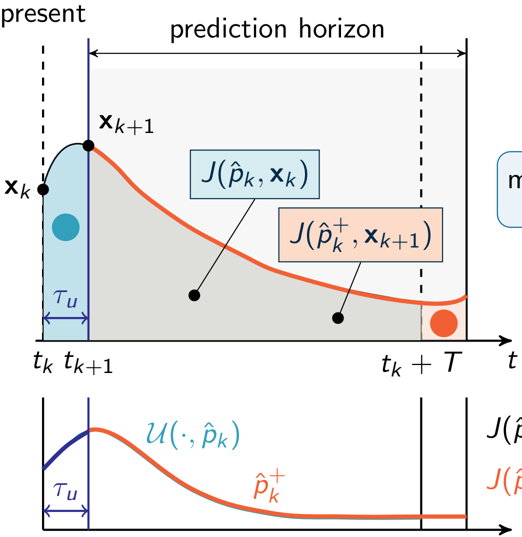
Initialize at $\hat{p}_k^+$ (hot start)

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

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

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

# 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)

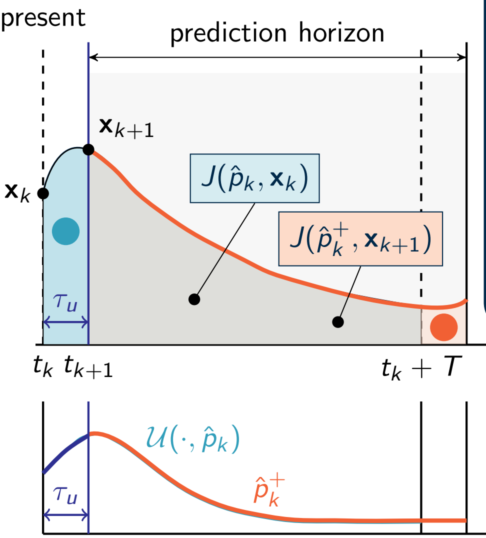
$$\hat{p}_{k+1} = \hat{p}(\mathbf{x}_{k+1})$$

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

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

$\leq 0$ ?

## 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)

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

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

$\leq 0$ ?

## 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
- $\hat{p}_k$ 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** $\leq 0$.

- Formulation involving Final constraints
- $\hat{p}_k$ sufficiently good

# Ideal MPC: The key assumptions

**Keep in mind**

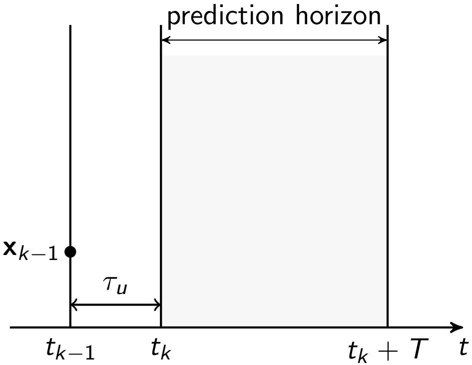
In a **realistic framework**, when the horizon moves, the hot start $p_k^+$ computed from the previous solution $p_k$ satisfies:

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

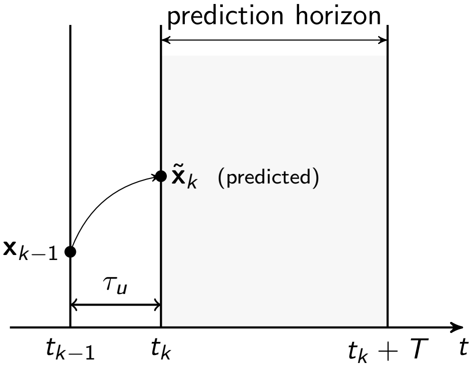
$D(0) = 0$
$D(\tau_u)$ is **not necessarily** $\leq 0$.

Even with **perfect undisturbed** model

## Preparation & Feedback Steps



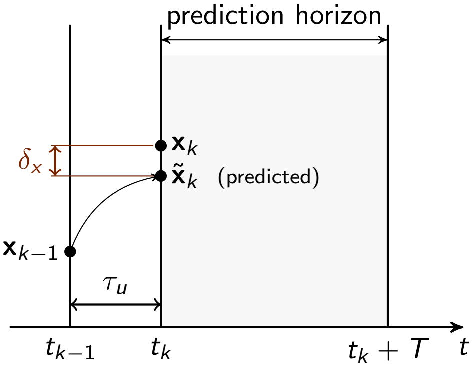
## Preparation & Feedback Steps




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

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

# Preparation & Feedback Steps



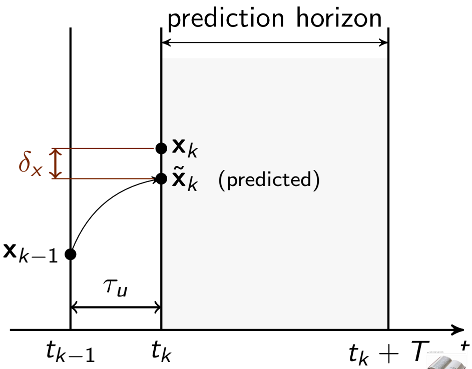

1. Predict $\tilde{\mathbf{x}}_k$
2. During $[t_{k-1}, t_k]$

   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 $\dfrac{\partial \hat{p}_k}{\partial \mathbf{x}}$]

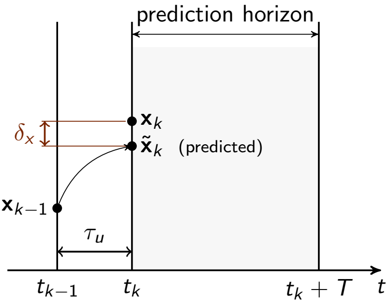
   (*preparation step*)
3. Once $\mathbf{x}_k$ is available:

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

(*feedback step*)

Diehl et al. SIAM J. Ctrl and Opt. (2005)
Zavala and Biegler. Auomatica (2009)

# Definition of Fast NMPC Problems

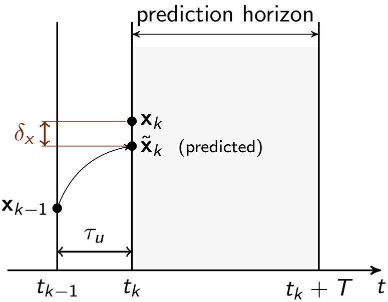



$\tau_u$ is the time between two control updating

$\tau_u$ is the time during which there is no feedback

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

## Definition of Fast NMPC Problems




$\tau_u$ is the time between two control updating

$\tau_u$ is the time during which there is no feedback

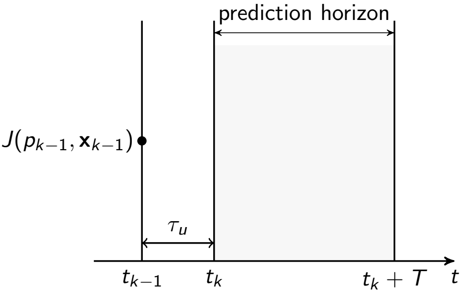
$$\Rightarrow \tau_u \leq \tau_u^{max}$$

**Fast NMPC Problems**

Fast NMPC problems are those for which

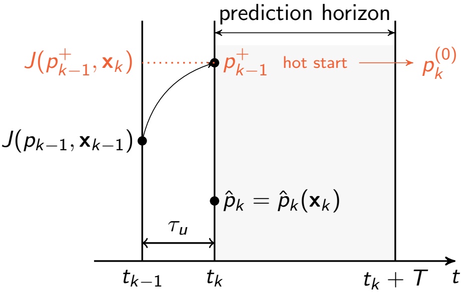
$$\tau_{solve}\Big(NLP(\tilde{\mathbf{x}}_k)\Big) \geq \tau_u^{max}$$

## The Iterative Process



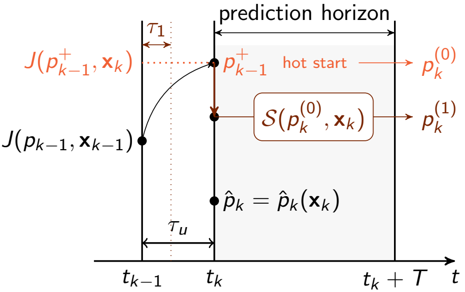

$$p^{(i+1)} \leftarrow \mathcal{S}(p^{(i)}, \tilde{\mathbf{x}}_k)$$

## The Iterative Process



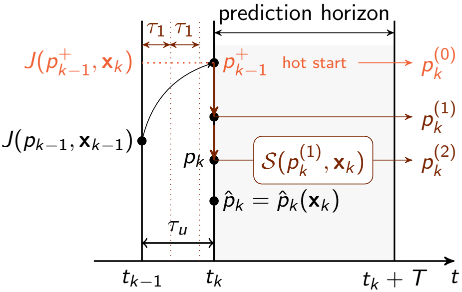

$$p^{(i+1)} \leftarrow \mathcal{S}(p^{(i)}, \tilde{\mathbf{x}}_k)$$

# The Iterative Process



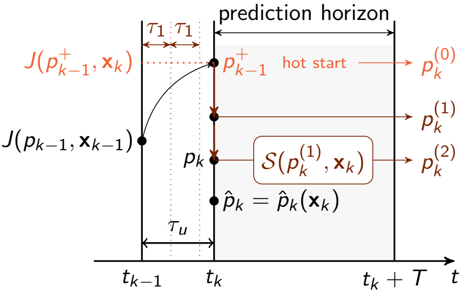

$$p^{(i+1)} \leftarrow \mathcal{S}(p^{(i)}, \tilde{\mathbf{x}}_k)$$

## The Iterative Process




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

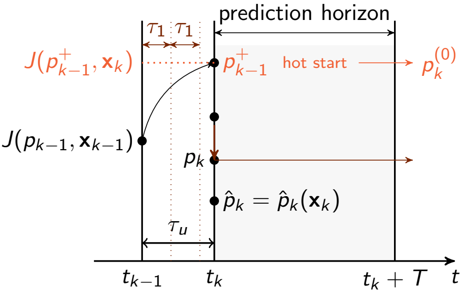
## The Iterative Process




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

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

# The Iterative Process




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

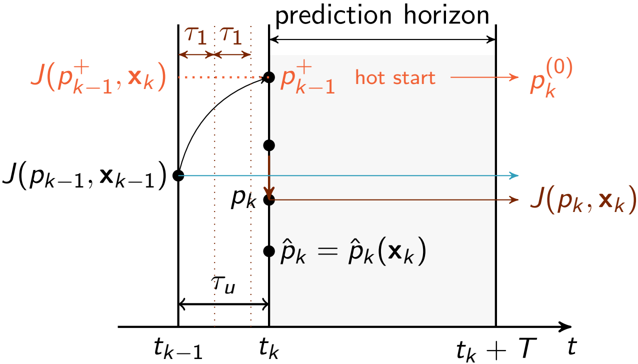
↓

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

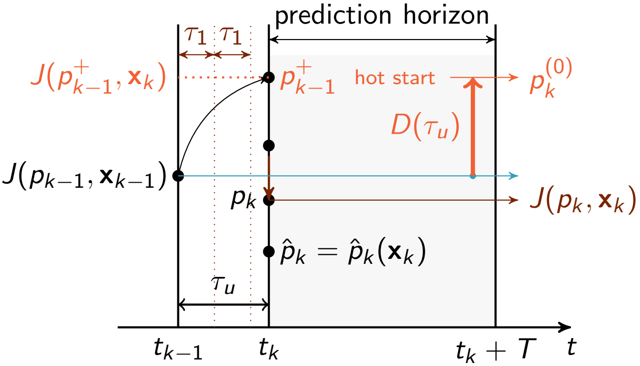
**dynamic equation for $p$**

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

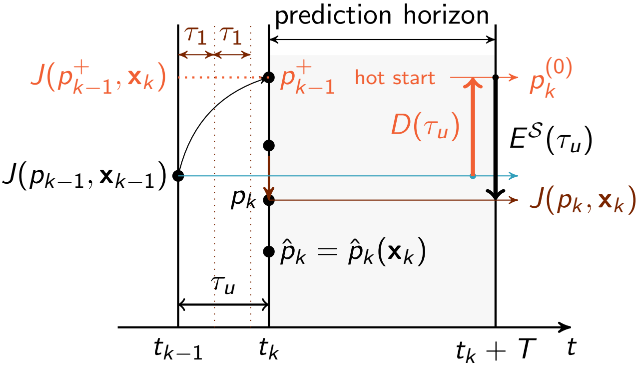
## Sufficient Conditions of Success



## Sufficient Conditions of Success



## Sufficient Conditions of Success

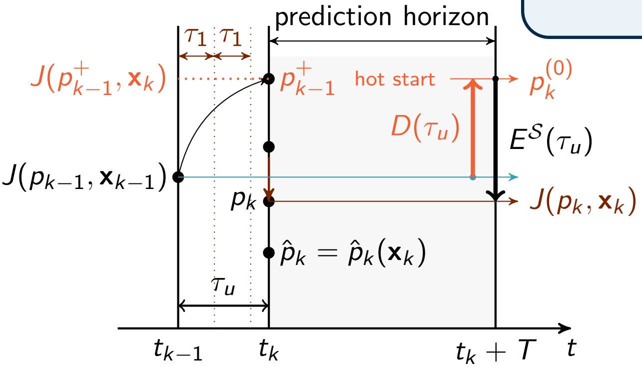


## Sufficient Conditions of Success

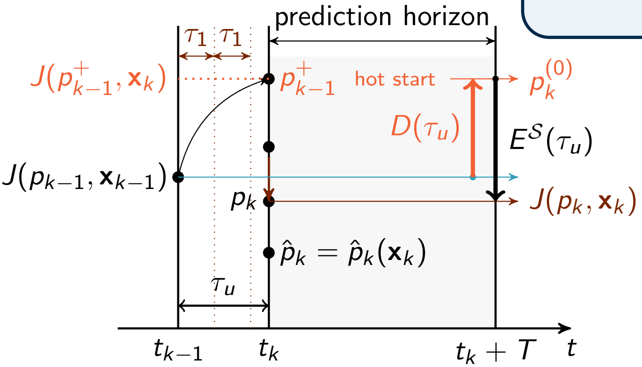
**Success Conditions**

Good choice of $(\mathcal{S}, \tau_u)$:

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



## Sufficient Conditions of Success

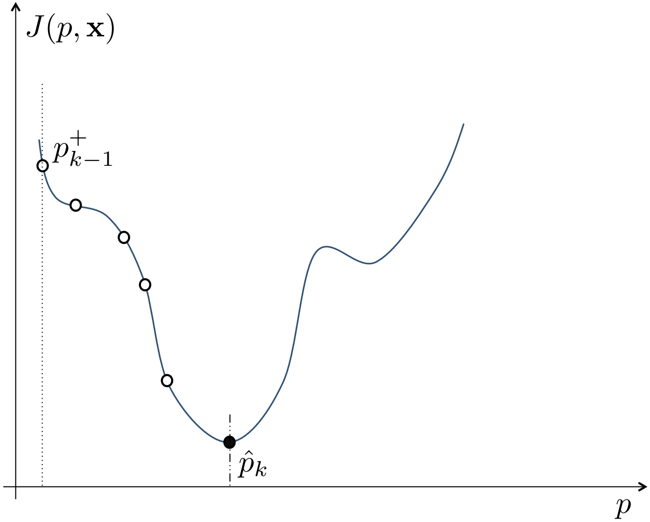



**Success Conditions**

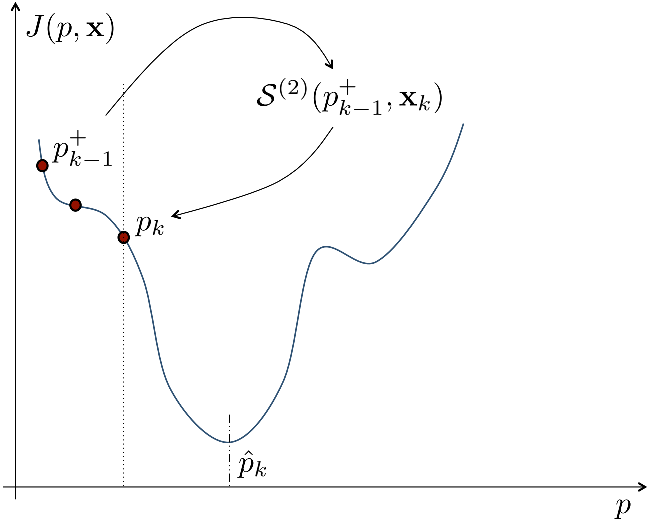
Good choice of $(\mathcal{S}, \tau_u)$:

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

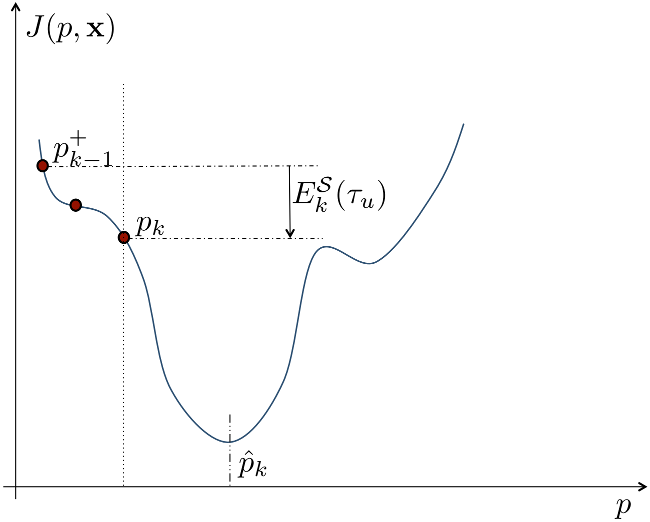
## Closed-Loop Evolution of the Cost Function



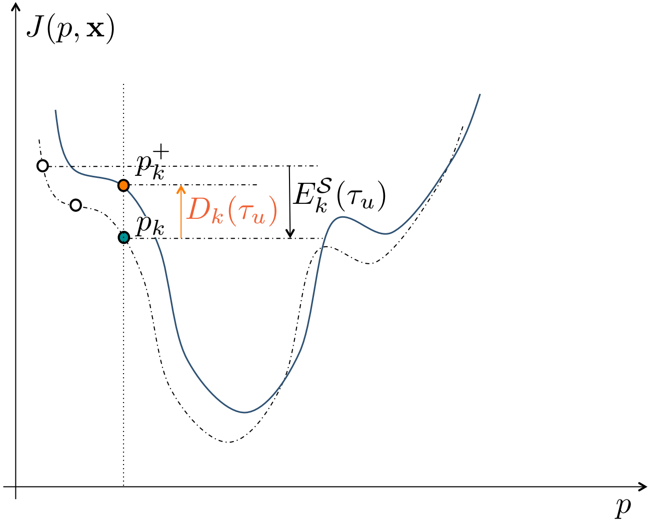
## Closed-Loop Evolution of the Cost Function



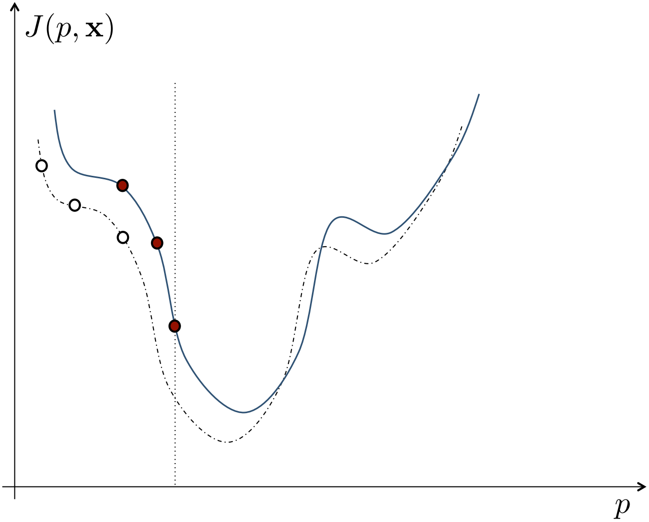
## Closed-Loop Evolution of the Cost Function



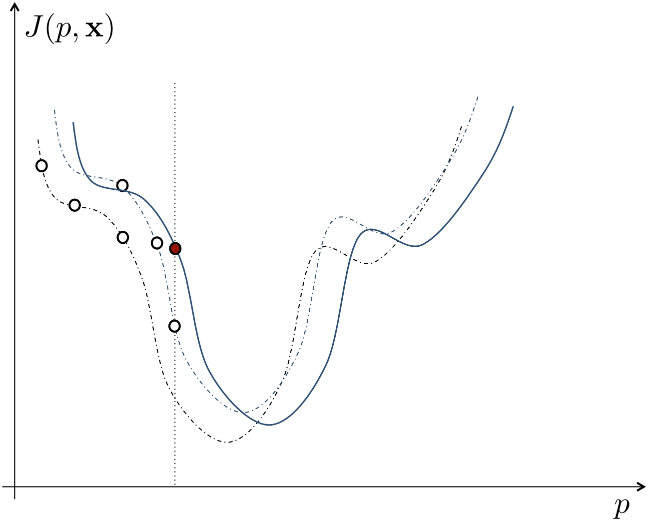
## Closed-Loop Evolution of the Cost Function



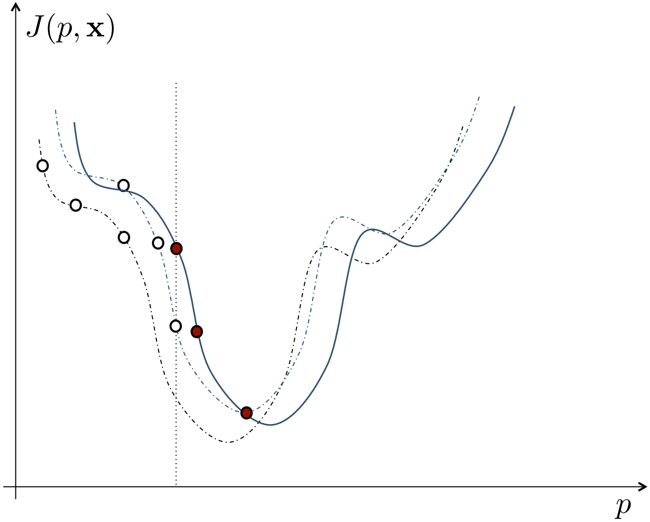
# Closed-Loop Evolution of the Cost Function



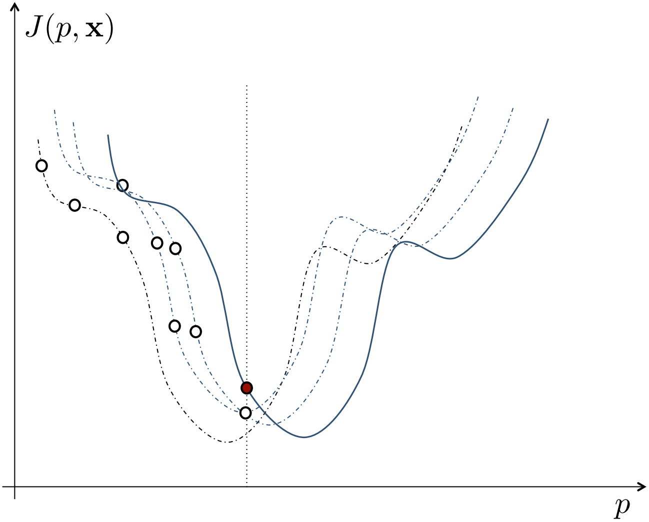
## Closed-Loop Evolution of the Cost Function



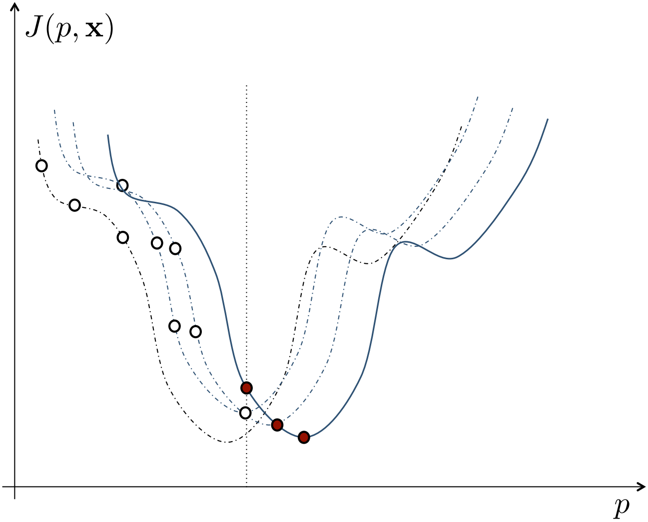
## Closed-Loop Evolution of the Cost Function



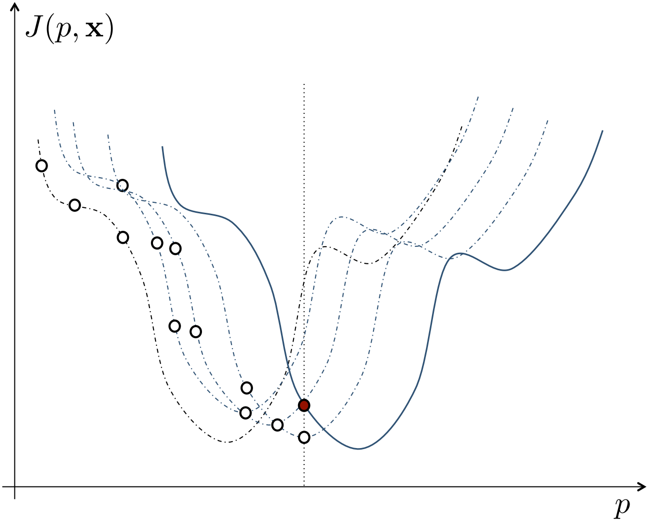
# Closed-Loop Evolution of the Cost Function



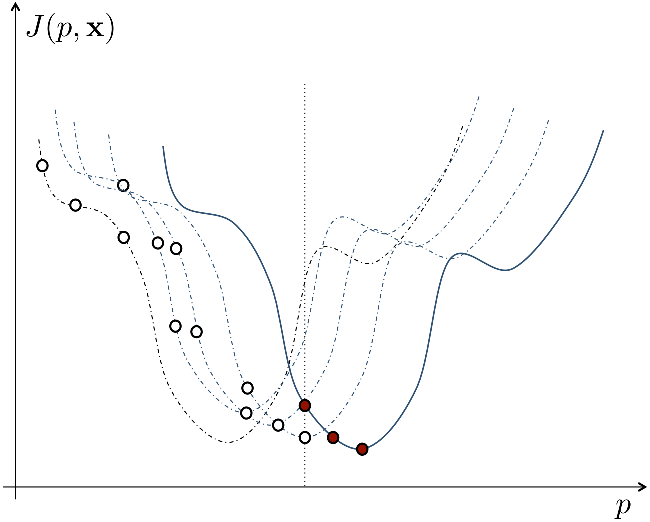
## Closed-Loop Evolution of the Cost Function



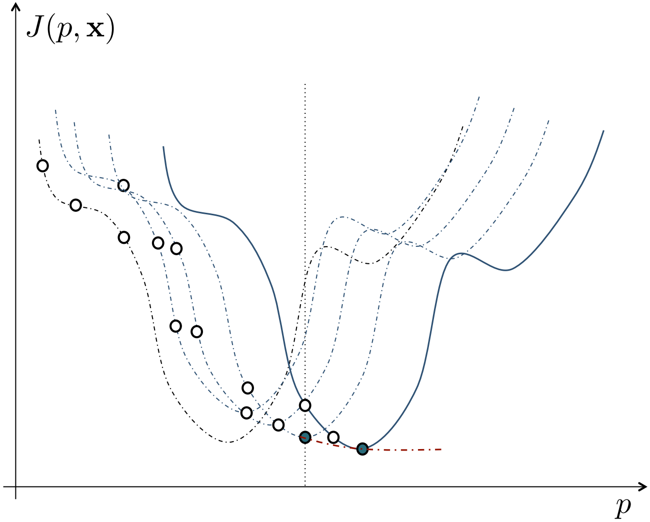
## Closed-Loop Evolution of the Cost Function



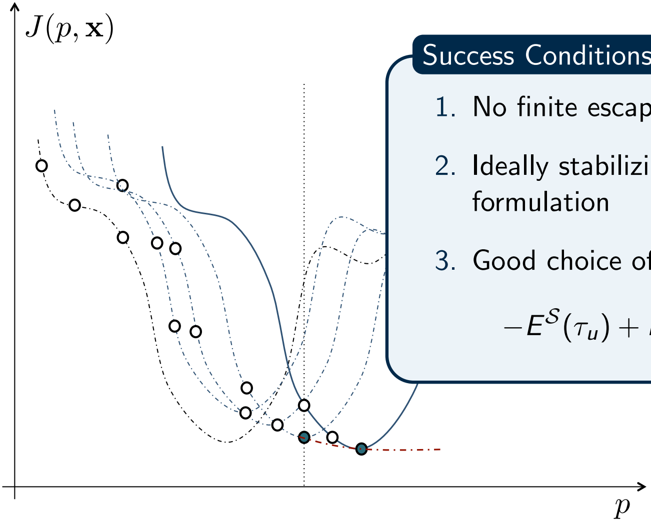
## Closed-Loop Evolution of the Cost Function



## 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 $(\mathcal{S}, \tau_u)$:

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

## Reminder

$$E^{\mathcal{S}}(\tau_u) \; > D(\tau_u)$$

## Reminder

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

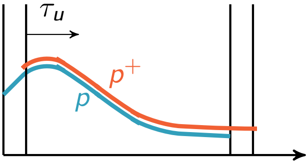
$D(\tau_u)$

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

## Reminder

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

### $D(\tau_u)$



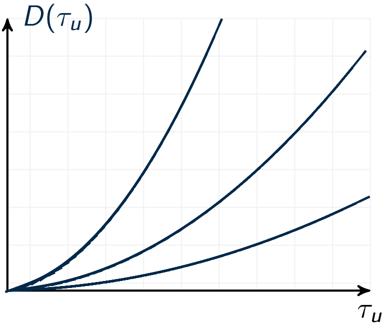

- $D(\tau_u) := J(p^+, \mathbf{x}^+) - J(p, \mathbf{x})$
- $D(0) = 0$, $D(\tau_u)$ can be $\geq 0$
- $\tau_u \in [0, \tau_u^{max}]$
- Independent of the solver $\mathcal{S}$

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

## Reminder

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

$D(\tau_u)$




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

## Reminder

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

**$D(\tau_u)$**

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

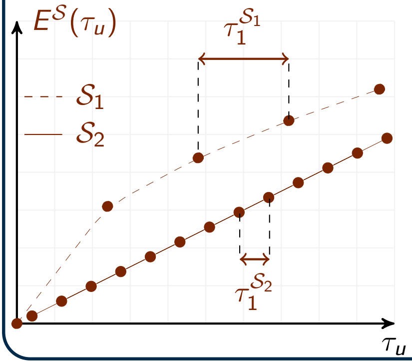
- $E^{\mathcal{S}}(\tau_u) := J(p^{(0)}, \mathbf{x}) - J(p^{(q^{\mathcal{S}})}, \mathbf{x})$
- $q^{\mathcal{S}} = int(\tau_u/\tau_1^{\mathcal{S}})$
- $\tau_1^{\mathcal{S}}$ time for a single iteration
- $\tau_u \in \{1, 2, \dots\} \times \tau_1^{\mathcal{S}}$
- $E^{\mathcal{S}}(0) = 0$

## Reminder

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

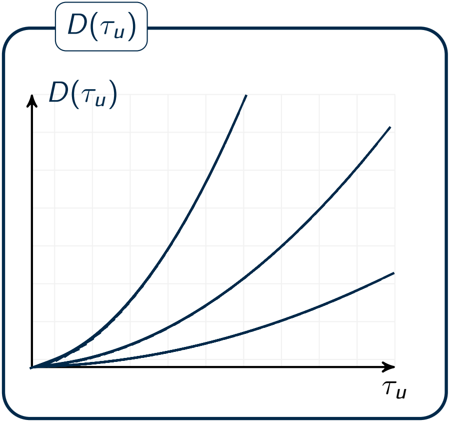
$D(\tau_u)$

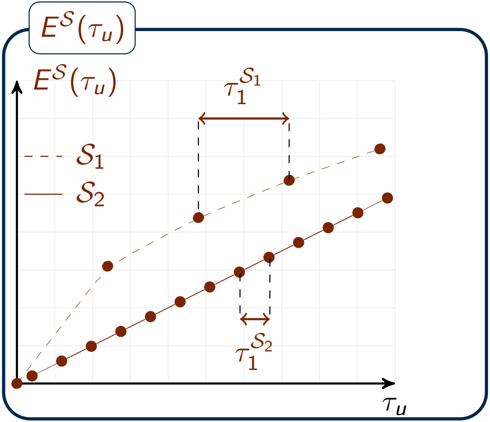
$E^{\mathcal{S}}(\tau_u)$

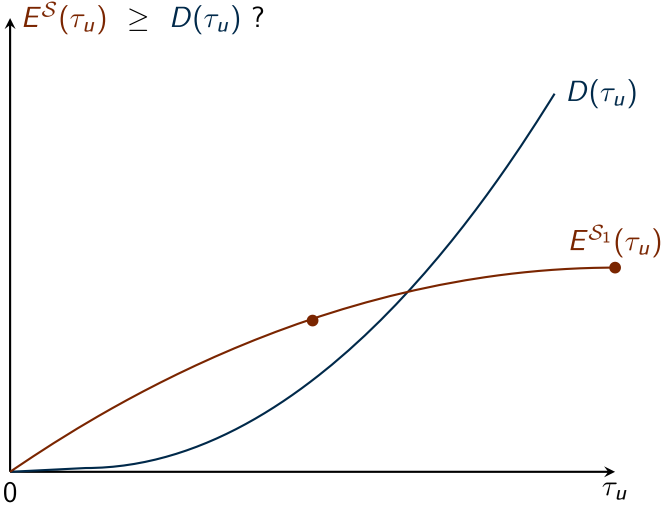


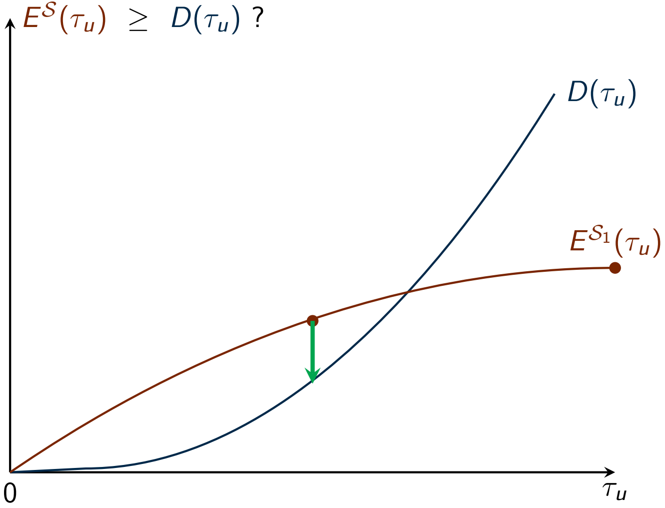
# Reminder

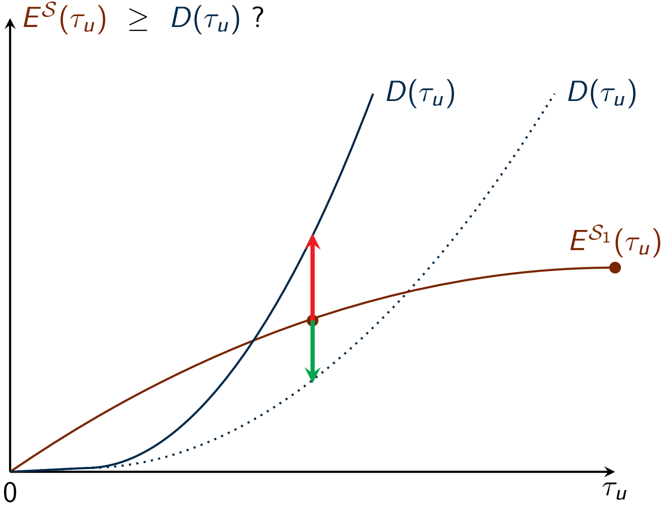
$$E^{\mathcal{S}}(\tau_u) > D(\tau_u)$$

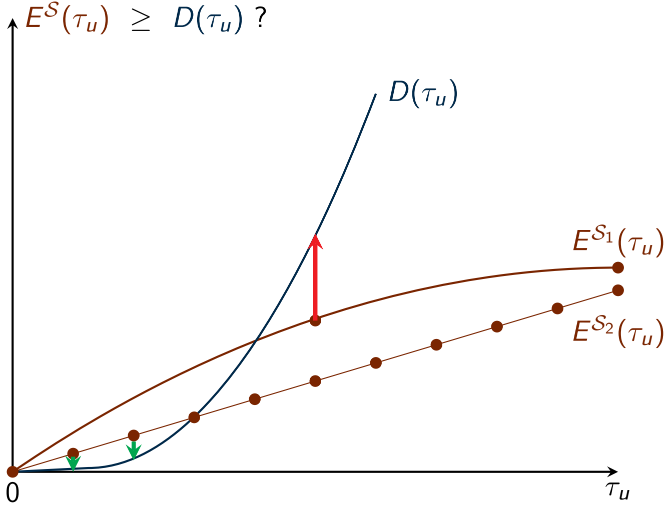


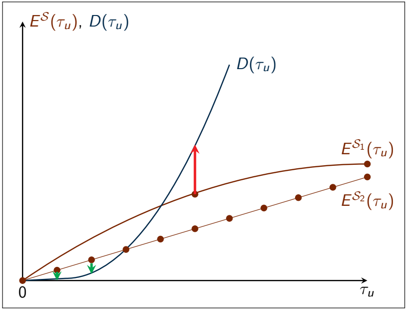

$E^{\mathcal{S}}(\tau_u) \geq D(\tau_u)$ ?
$D(\tau_u)$
$E^{\mathcal{S}_1}(\tau_u)$
0
$\tau_u$


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


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


$E^{\mathcal{S}}(\tau_u) \geq D(\tau_u)$ ?
$D(\tau_u)$
$E^{\mathcal{S}_1}(\tau_u)$
$E^{\mathcal{S}_2}(\tau_u)$
0
$\tau_u$

# Key properties of a solver

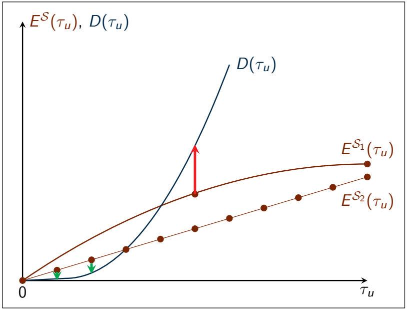



## Keep in Mind

It is sometimes better to choose a **less efficient**[a] solver with **shorter preparation step** duration $\tau_1$.

[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 $\tau_1$.

[a] per iteration

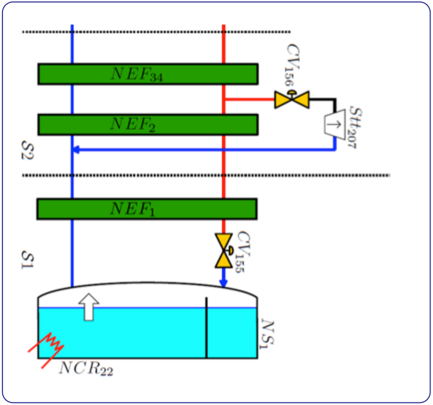
Gradient-based studies

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

Heuristics for second order methods

Bock et al. SIAM (2007)

# Back to Cryogenics



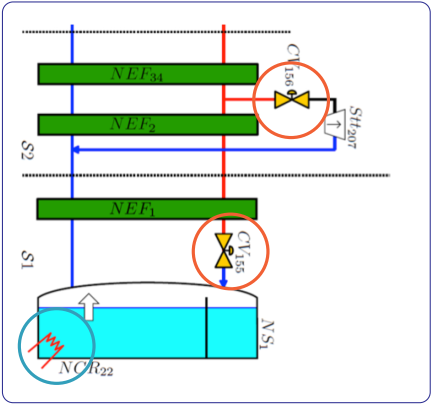

Source: Fr. Bonne PhD defense

After linearization:

$$x_{k+1} = Ax_k + B\begin{pmatrix} u_k \\ w_k \end{pmatrix}$$

$$y_k = Cx_k + D\begin{pmatrix} u_k \\ w_k \end{pmatrix}$$

# Back to Cryogenics



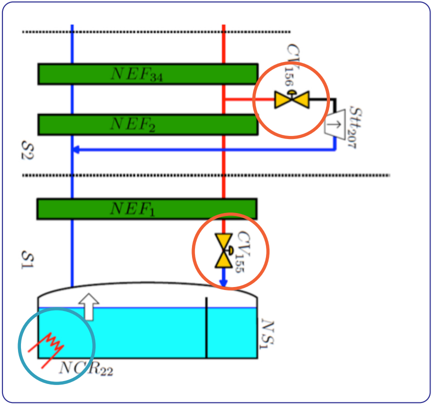

Source: Fr. Bonne PhD defense

After linearization:

$$x_{k+1} = Ax_k + B\begin{pmatrix} u_k \\ w_k \end{pmatrix}$$

$$y_k = Cx_k + D\begin{pmatrix} u_k \\ w_k \end{pmatrix}$$

# Back to Cryogenics




Source: Fr. Bonne PhD defense

After linearization:

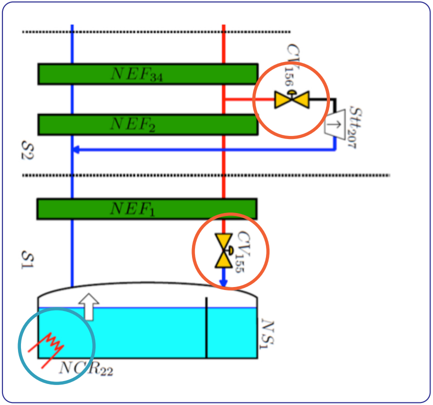
$$x_{k+1} = Ax_k + B\begin{pmatrix} u_k \\ w_k \end{pmatrix}$$

$$y_k = Cx_k + D\begin{pmatrix} u_k \\ w_k \end{pmatrix}$$

Constraints are bounds on the state and control components

(affine in **u**)

## Back to Cryogenics




Source: Fr. Bonne PhD defense

After linearization:

$$x_{k+1} = Ax_k + B\begin{pmatrix} u_k \\ w_k \end{pmatrix}$$

$$y_k = Cx_k + D\begin{pmatrix} u_k \\ w_k \end{pmatrix}$$

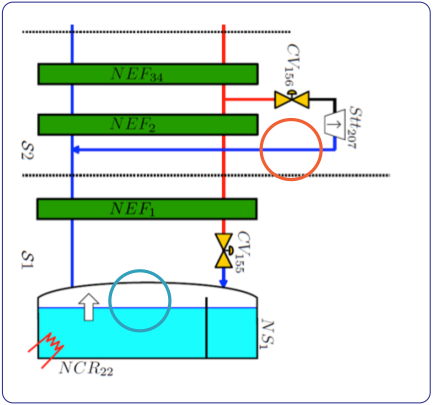
Constraints are bounds on the state and control components

(affine in $\mathbf{u}$)

$\Downarrow$

**QP problems** to be solved at each updating period

## Back to Cryogenics

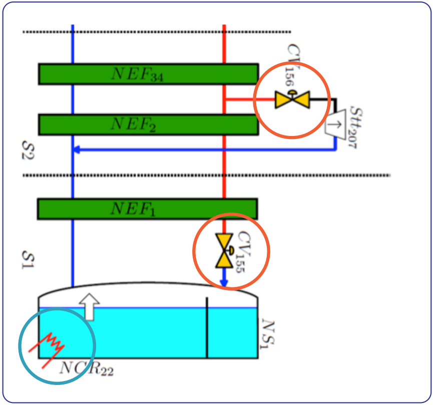



Source: Fr. Bonne PhD defense

- Output Turbine temperature must be higher than some threshold to avoid solid droplets
- The helium bath level must remains between a lower and an upper bound to avoid extreme situation

$$\rightarrow y_{min} \leq y_k \leq y_{max}$$

# Back to Cryogenics

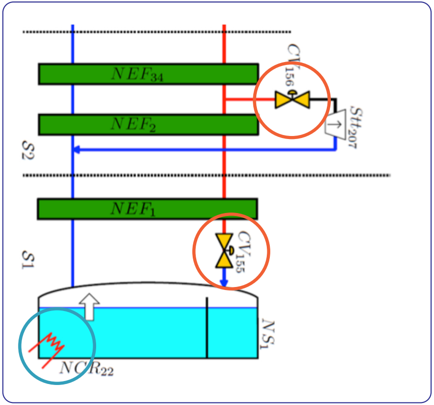



Source: Fr. Bonne PhD defense

- Valves opening is constrained between 0 and 100%
- Speed of valve opening is also limited

$$\begin{pmatrix} u_{min} \\ \delta_{min} \end{pmatrix} \leq \begin{pmatrix} u_k \\ \delta u_k \end{pmatrix} \leq \begin{pmatrix} u_{max} \\ \delta_{max} \end{pmatrix}$$

# Back to Cryogenics




Source: Fr. Bonne PhD defense

Degrees of freedom:

$$p = \mathbf{u}_k := \begin{pmatrix} u_k & u_{k+1} & \dots & u_{k+N_p-1} \end{pmatrix}$$

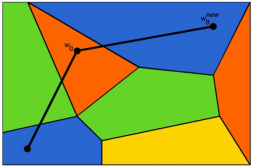
Cost function:

$$J(p, \mathbf{x}_k) := \sum_{i=1}^{N_p} \|\mathbf{x}_{k+i}(p) - \mathbf{x}^{ref}_{k+i}\|^2_Q + \sum_{i=0}^{N_p-1} \|u_{k+i}(p) - u^{ref}_{k+i}\|^2_Q$$

# Back to Cryogenics: The QP-OASES Solver

## qpOASES User's Manual

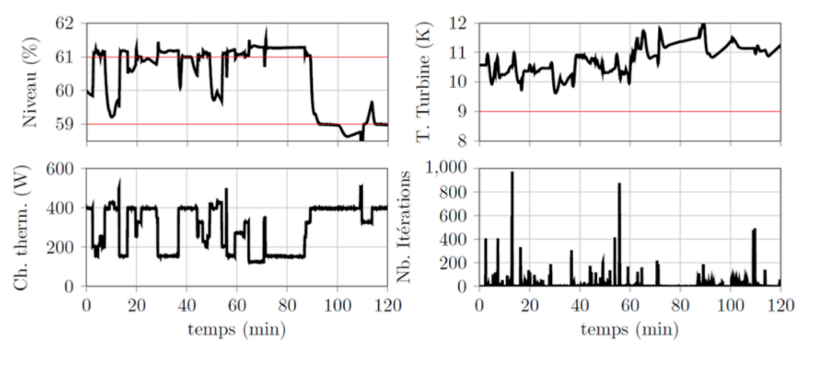
**Version 3.0beta (April 2014)**



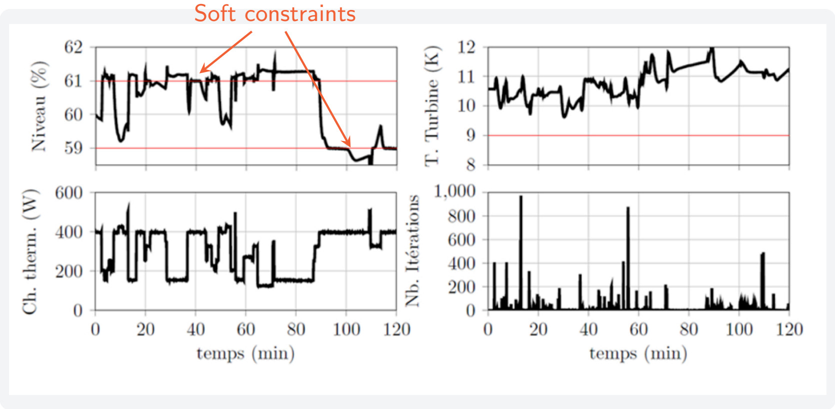

Hans Joachim Ferreau et al.[1]
ABB Corporate Research, Switzerland
support@qpOASES.org

[1] past and present qpOASES developers and contributors in alphabetical order: Eckhard Arnold, Alexander Buchner, Holger Diedam, Hans Joachim Ferreau, Boris Houska, Christian Kirches, Manuel Kudruss, Aude Perrin, Andreas Potschka, Milan Vukov, Thomas Wiese, Sebastian F. Walter, Leonard Wirsching

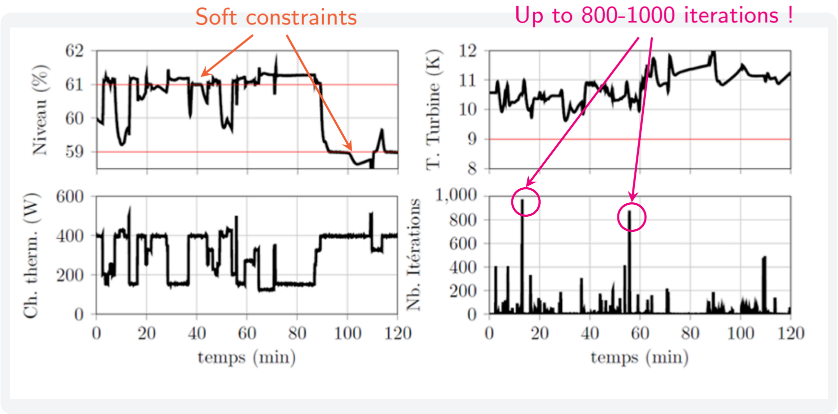
# Back to Cryogenics: Results



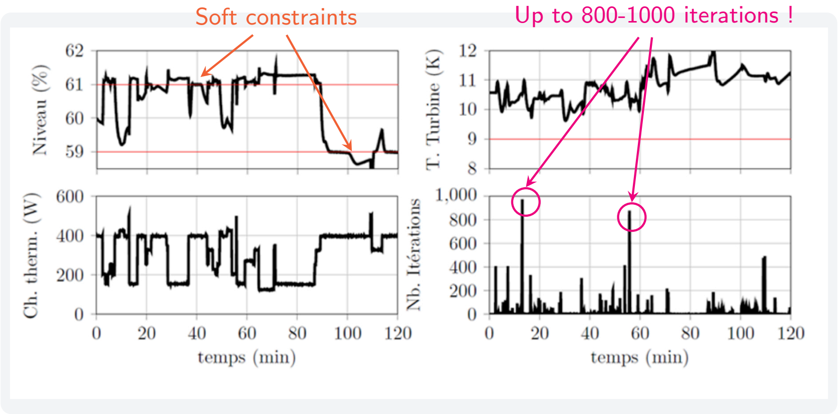
# Back to Cryogenics: Results



# Back to Cryogenics: Results



# Back to Cryogenics: Results




Is it real-time compatible ?

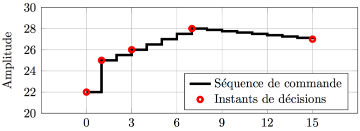
$$n_u = 3 + 2 \times 2 = 7$$

$$N_p = 100 \rightarrow n_p = 700 \text{ !!}$$

# Back to Cryogenics: Reducing the problem's complexity

**The cost** of a single iteration **depends** on:

- The number of decision variables $n_p$ (dimension of $p$)
- The number of constraints $n_c$ (number of line in $A$)



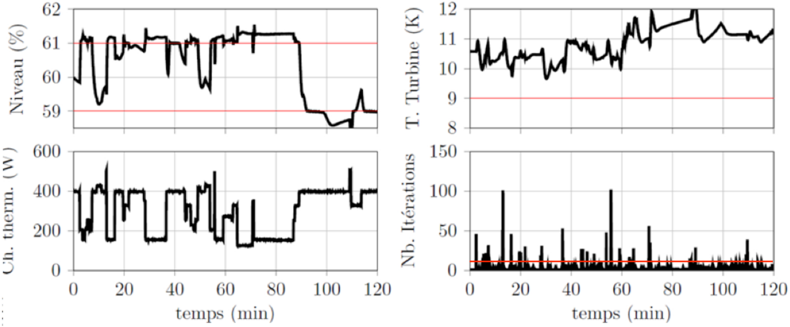

Using linear interpolation:

$n_p : 700 \rightarrow 49$

Checking constraints only at some chosen instants

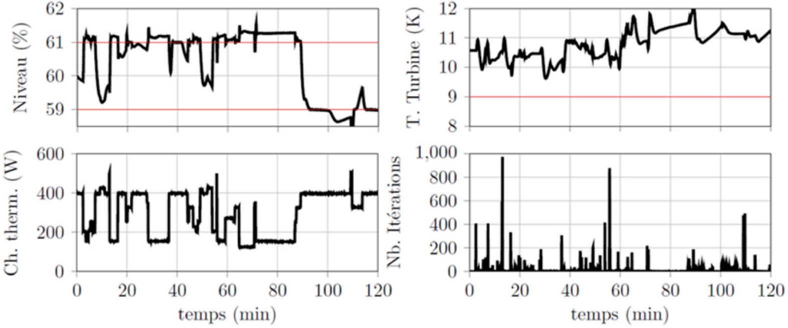
$n_c : 1000 \rightarrow 98$

# Back to Cryogenics: Simulation of the reduced dimensional formulation




Results with the reduced dimensional parametrization

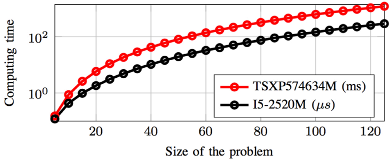
## Back to Cryogenics: Simulation of the reduced dimensional formulation




Results with the original parametrization

# Back to Cryogenics: Real-Time Considerations

Schneider TSX premium PLC

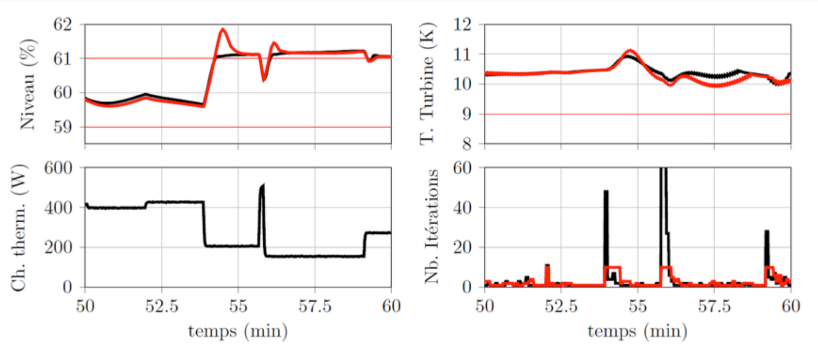



Time needed for a Cholesky factorization

| | |
|---|---|
| Slowing factor | $\approx 4000$ |
| 1 qpOASES iter (Pentium) | $\approx 120\mu s$ |
| 1 qpOASES iter (PLC) | $\approx 0.48\ s$ |
| Sampling period | $= 5\ s$ |

Only 10 iterations of QP-OASES can be performed during the sampling period

# Back to Cryogenics: Results with interrupted QPOASES solver



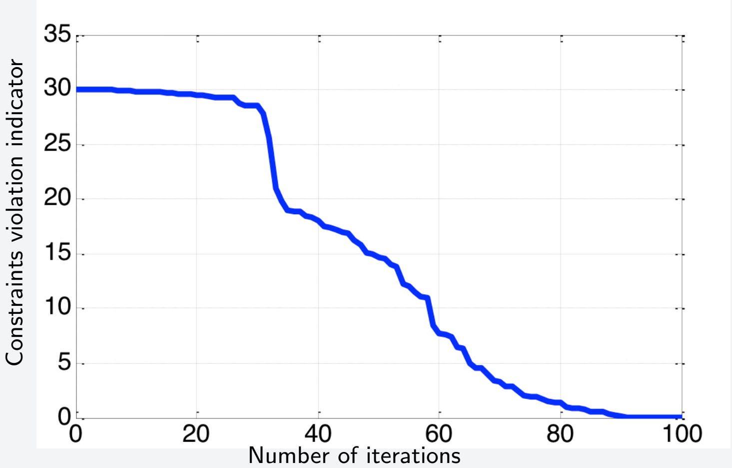

QPOASES limited to 10 iterations

QPOASES without interruption

Source: Fr. Bonne PhD defense

# What happened?

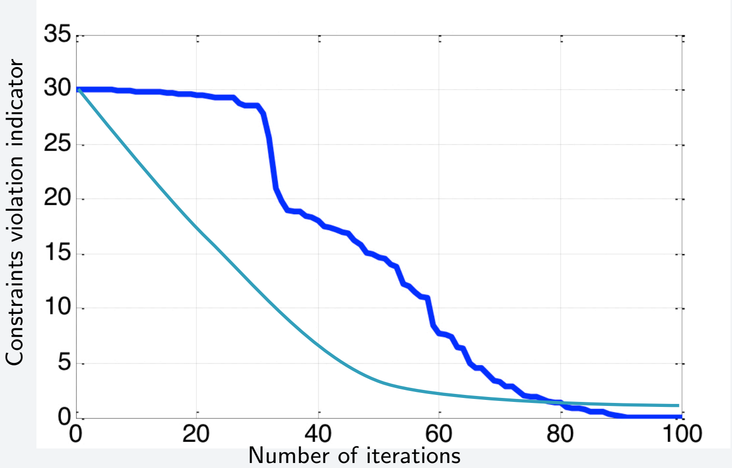
Typical behavior of active set solvers




Source: Fr. Bonne PhD defense

# What happened?

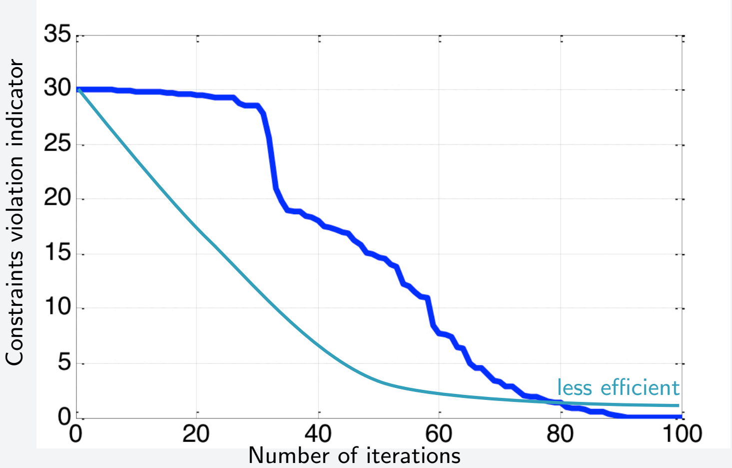
Typical behavior of active set solvers




Source: Fr. Bonne PhD defense

## What happened?

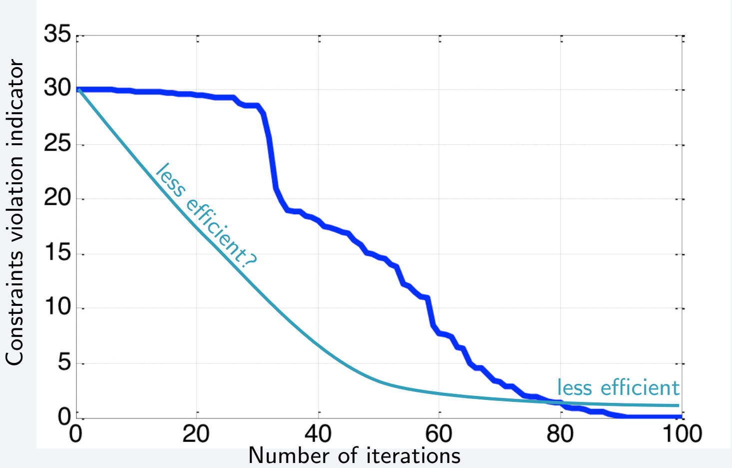
Typical behavior of active set solvers




Source: Fr. Bonne PhD defense

## What happened?

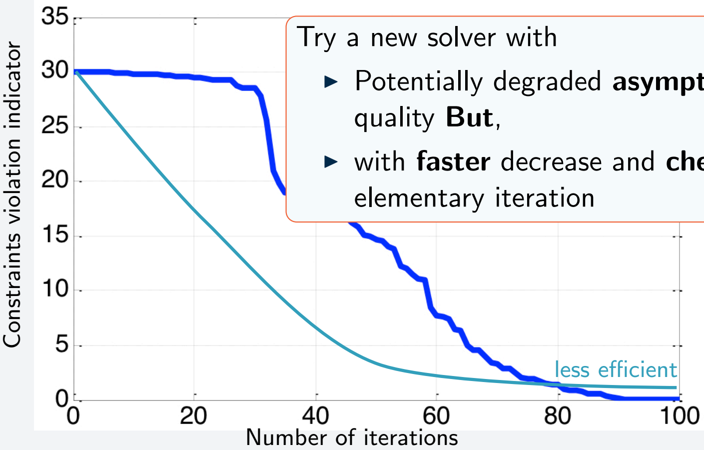
Typical behavior of active set solvers




Source: Fr. Bonne PhD defense

## What happened?

Typical behavior of active set solvers




Try a new solver with

- Potentially degraded **asymptotic** quality **But**,
- with **faster** decrease and **cheaper** elementary iteration

Source: Fr. Bonne PhD defense

## Solver based on integrating stiff ODEs

$$\min_p \quad [J_0(p)] \quad \text{under } g(p) \leq 0$$

↓ Embed constraints in cost

$$\min_p \quad [J(p)] := \left[J_0(p) + \rho \max\{0, g(p)\}^2\right]$$

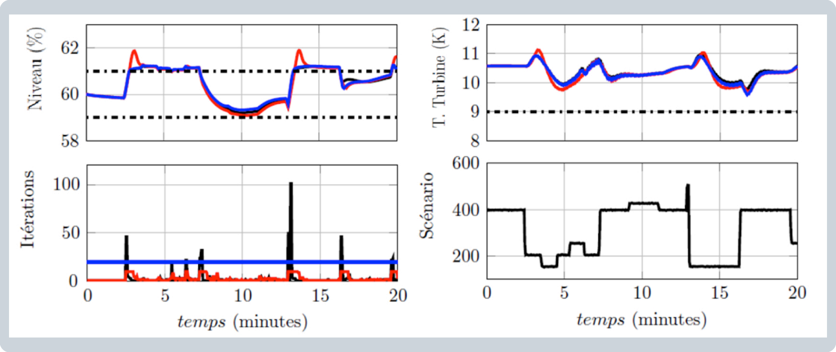
↓ Solve the stiff ODE

$$\dot{p} = -\left[\frac{\partial J}{\partial p}(p)\right]$$

↓ Consequence

**20 iterations are possible** (instead of 10 for QPOASES)

# Comparison between RT-performances of QPOASES and ODE-solver



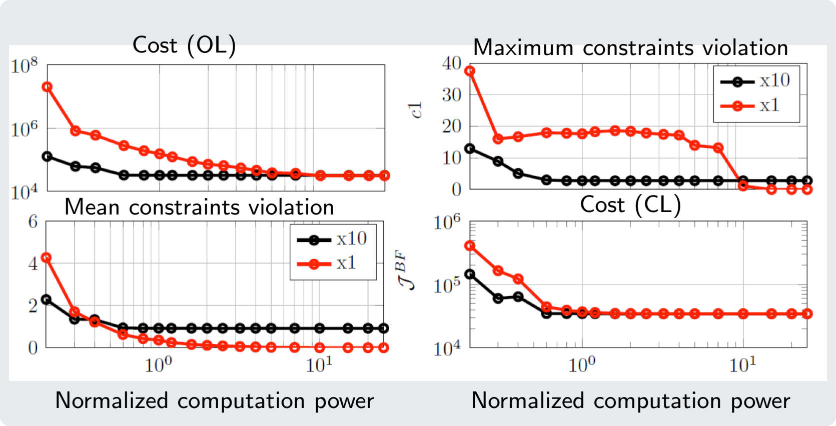

QPOASES unlimited

QPOASES limited to 10 iterations

ODE-solver limited to 20 iterations

Source: Fr. Bonne PhD defense

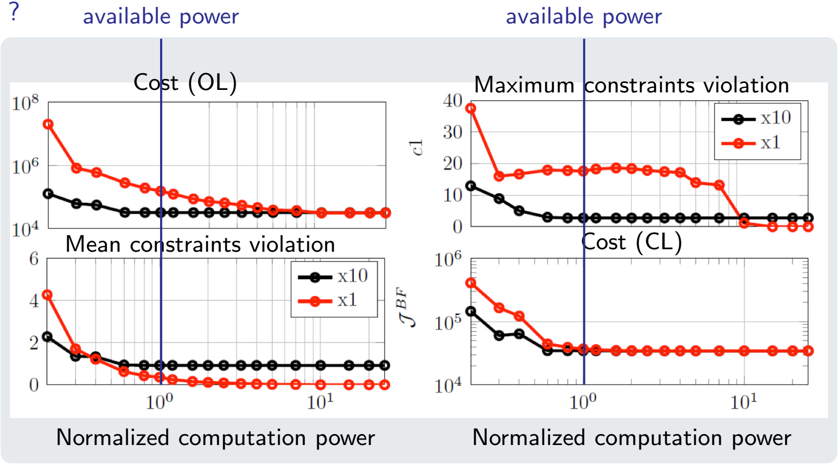
## Why ?




QPOASES / ODE-solver

Source: Fr. Bonne PhD defense

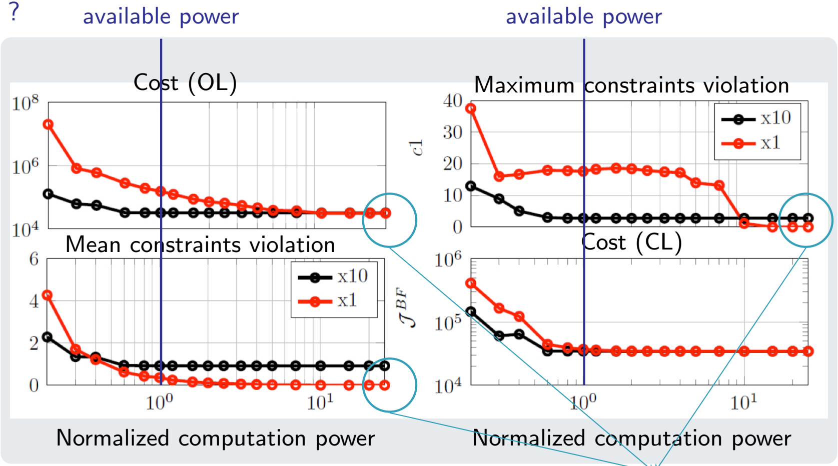
## Why ?




QPOASES / ODE-solver

Source: Fr. Bonne PhD defense

## Why ?

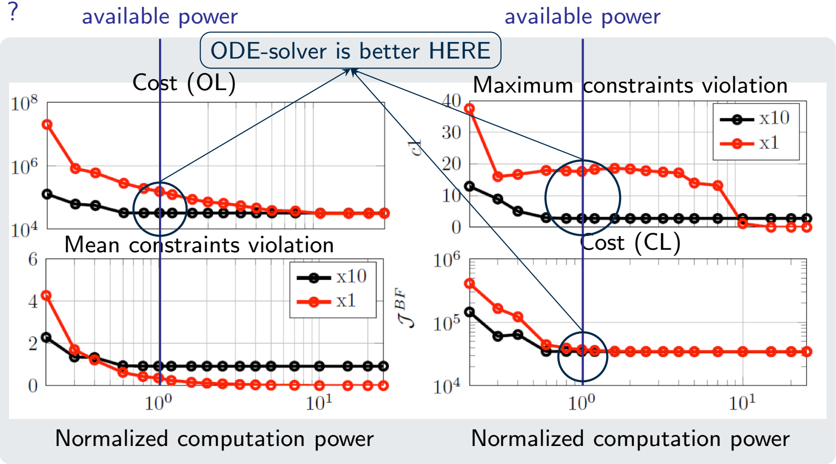



QPOASES is better in an ideal world . . . !

QPOASES / ODE-solver

Source: Fr. Bonne PhD defense

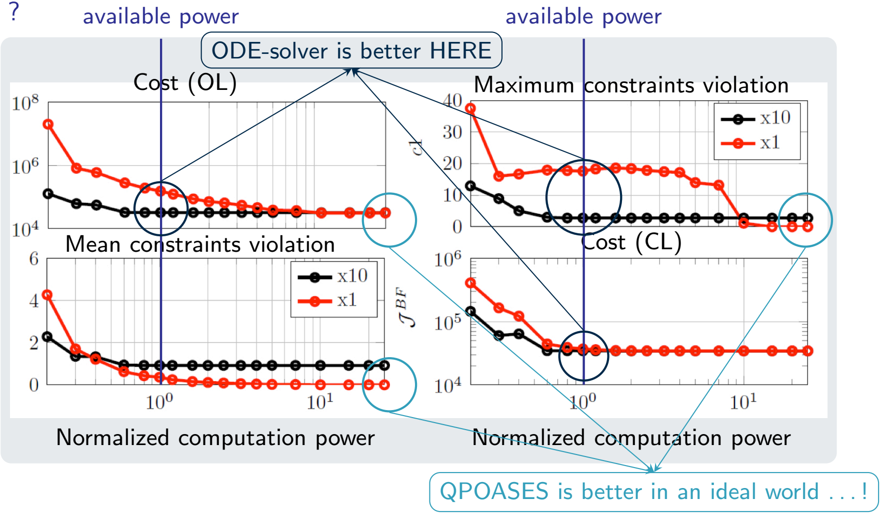
## Why ?




QPOASES / ODE-solver

Source: Fr. Bonne PhD defense

## Why ?




QPOASES / ODE-solver

Source: Fr. Bonne PhD defense

## Regarding the Solver Choice

### Keep in Mind

1) In RT-MPC, what does matter is the **Arithmetical** Complexity and not the **Analytical** Complexity[1].

2) In RT-MPC, what does matter is the **Transient** Behavior and not the **Asymptotic** Behavior.

| | |
|---|---|
| Arithmetical | Number of elementary operations |
| Analytical | Number of iterations |

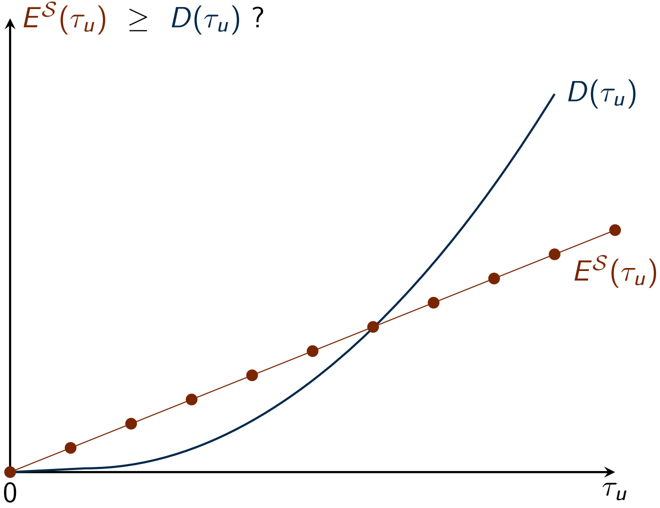
[1] Y.Nesterov. Introductory lectures in convex optimization 2004

## Updating Scheme For a Given Solver

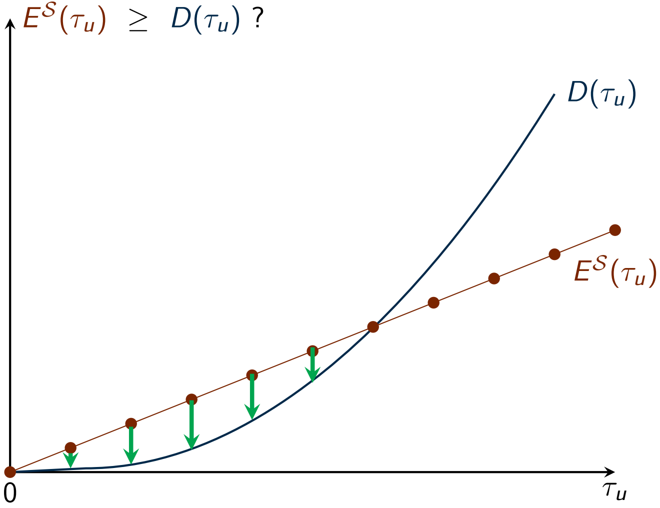
Assume that a solver $\mathcal{S}$ has been chosen . . .

Is there any remaining choice ?

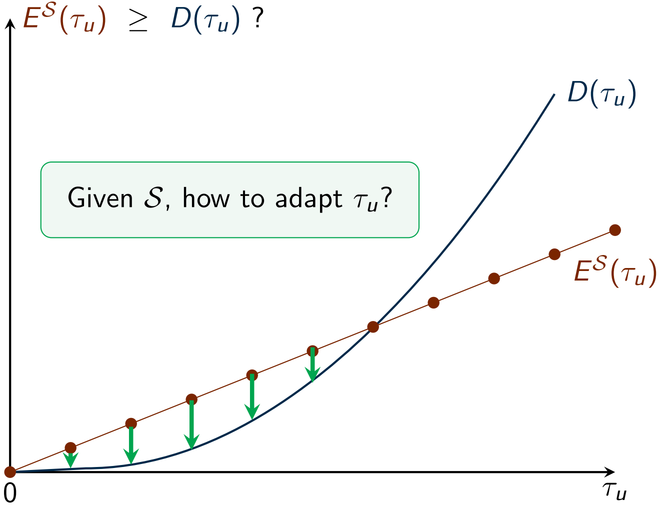
## What is the optimal $\tau_u$ for a given solver?



# What is the optimal $\tau_u$ for a given solver?



## What is the optimal $\tau_u$ for a given solver?



## Updating $\tau_u$ is a control problem . . . !

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

## Updating $\tau_u$ 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 $\tau_u$ 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 $\tau_u$ is a control problem . . . !

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

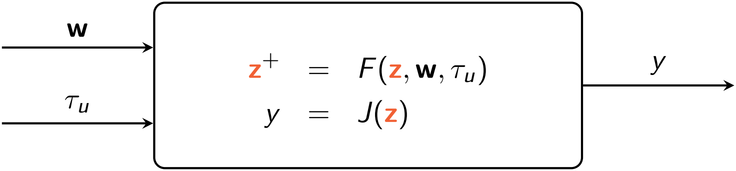
## Updating $\tau_u$ is a control problem . . . !

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

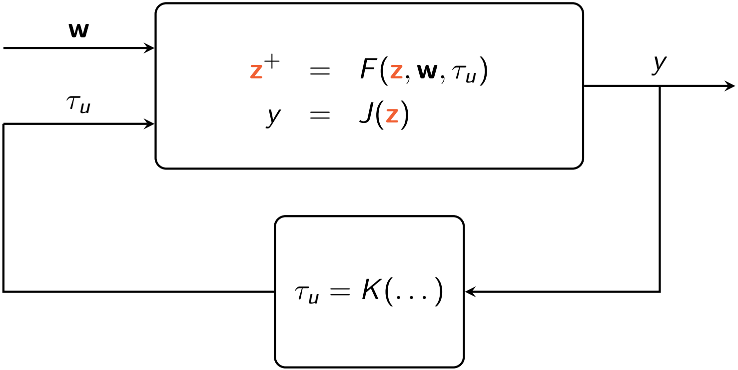
## Updating $\tau_u$ is a control problem . . . !

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

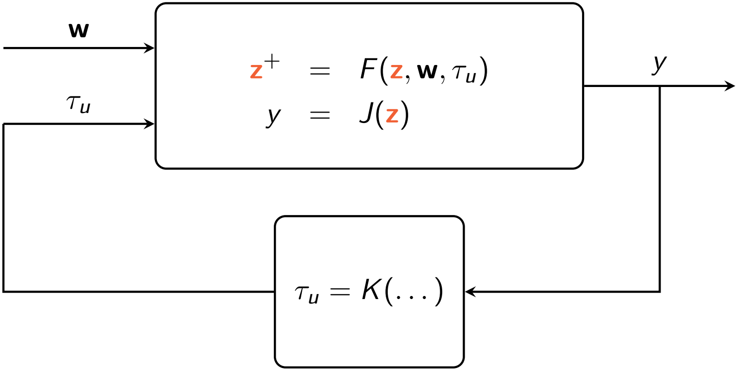
## Updating $\tau_u$ is a control problem . . . !



## Updating $\tau_u$ is a control problem ...!



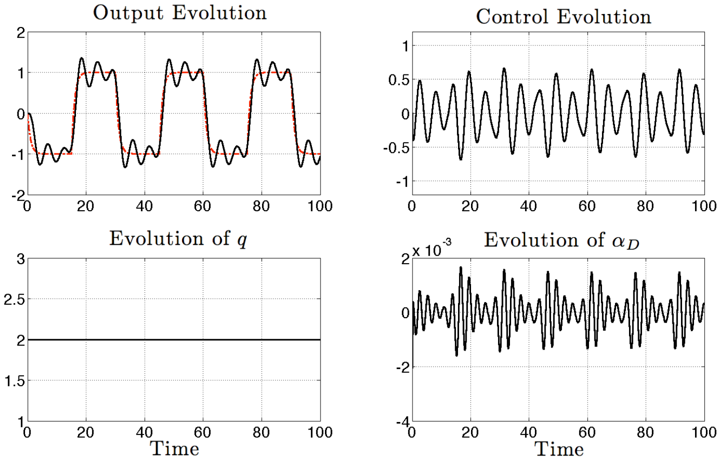
## Updating $\tau_u$ is a control problem ...!




Complexity: (5±, 5×, 5÷ and 1 log)

M.A. ECC (2013)

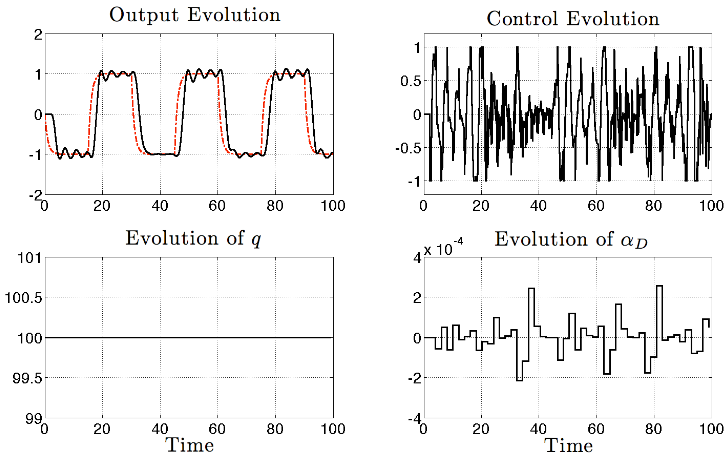
## $q = 2$ without adaptation





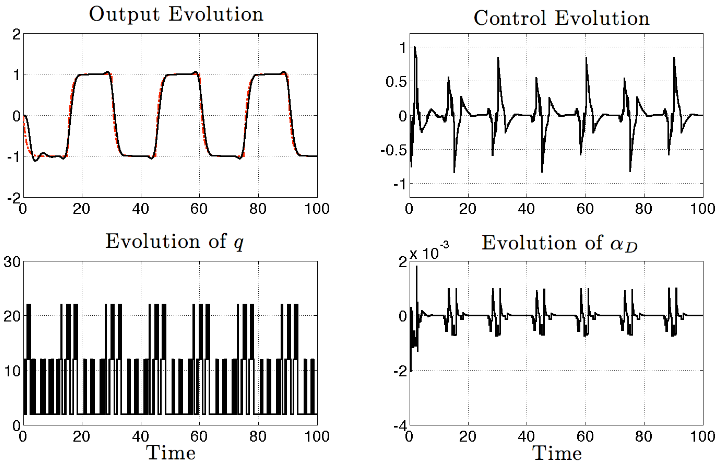
MA, ECC (2013)

## $q = 100$ without adaptation



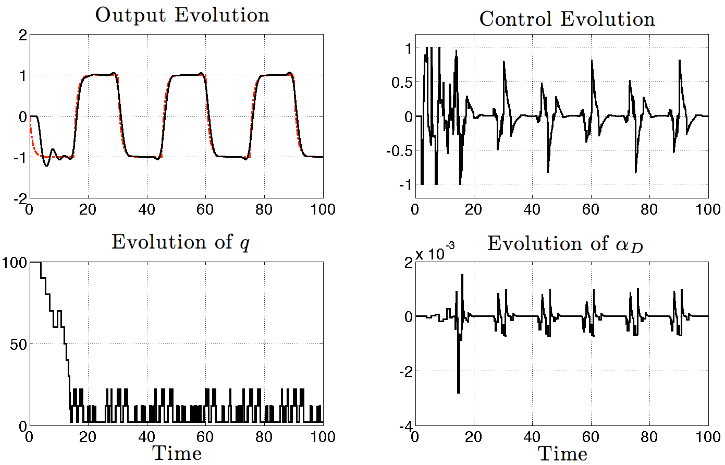

MA, ECC (2013)

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




MA, ECC (2013)

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




MA, ECC (2013)

**Certification bound**

The integer $N(p^{(0)}, \epsilon)$ s.t

$$|J(p^{(i)}, \mathbf{x}_k) - J(p^{opt}, \mathbf{x}_k)| \le \epsilon$$

for all $i \ge N(p^{(0)}, \epsilon)$.

| | |
|---|---|
| Bemporad and Patrinos | (2012) |
| Richter et al. Automatica | (2012) |
| Jones et al. | (2012) |
| MA | (2015) |

**Certification bound**

The integer $N(p^{(0)}, \epsilon)$ s.t

$$|J(p^{(i)}, \mathbf{x}_k) - J(p^{opt}, \mathbf{x}_k)| \leq \epsilon$$

for all $i \geq N(p^{(0)}, \epsilon)$.

When available

N$^o$ of iterations ($q$) $\Leftrightarrow$ guaranteed precision ($\epsilon$)

| | |
|---|---|
| Bemporad and Patrinos | (2012) |
| Richter et al. Automatica | (2012) |
| Jones et al. | (2012) |
| MA | (2015) |

**Certification bound**

The integer $N(p^{(0)},\epsilon)$ s.t

$$|J(p^{(i)},\mathbf{x}_k)-J(p^{opt},\mathbf{x}_k)| \le \epsilon$$

for all $i \ge N(p^{(0)},\epsilon)$.

When available ↓

N$^o$ of iterations $(q)$ $\Leftrightarrow$ guaranteed precision $(\epsilon)$

Easier to include in stability analysis

| | |
|---|---|
| Bemporad and Patrinos | (2012) |
| Richter et al. Automatica | (2012) |
| Jones et al. | (2012) |
| MA | (2015) |

**Certification bound**

The integer $N(p^{(0)}, \epsilon)$ s.t

$$|J(p^{(i)}, \mathbf{x}_k) - J(p^{opt}, \mathbf{x}_k)| \leq \epsilon$$

for all $i \geq N(p^{(0)}, \epsilon)$.

When available

N$^o$ of iterations ($q$) $\Leftrightarrow$ guaranteed precision ($\epsilon$)

Easier to include in stability analysis

| | |
|---|---|
| Bemporad and Patrinos | (2012) |
| Richter et al. Automatica | (2012) |
| Jones et al. | (2012) |
| MA | (2015) |

Reminder $\longrightarrow$ Arithmetical/Analytical Complexity

**Certification bound**

The integer $N(p^{(0)}, \epsilon)$ s.t

$$|J(p^{(i)}, \mathbf{x}_k) - J(p^{opt}, \mathbf{x}_k)| \leq \epsilon$$

for all $i \geq N(p^{(0)}, \epsilon)$.

| | |
|---|---|
| Bemporad and Patrinos | (2012) |
| Richter et al. Automatica | (2012) |
| Jones et al. | (2012) |
| MA | (2015) |

Reminder → Arithmetical/Analytical Complexity

When available

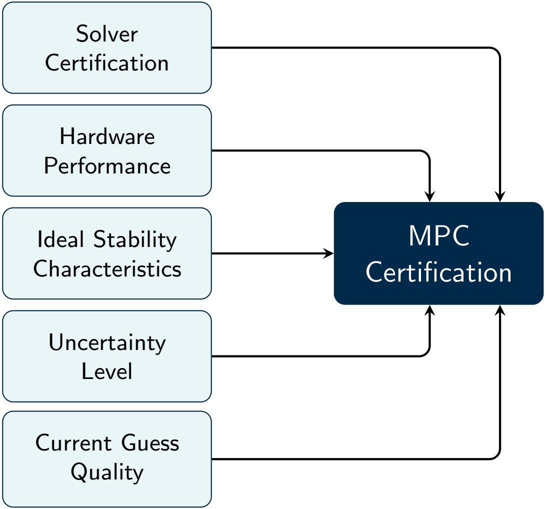
N$^o$ of iterations ($q$) ⇔ guaranteed precision ($\epsilon$)

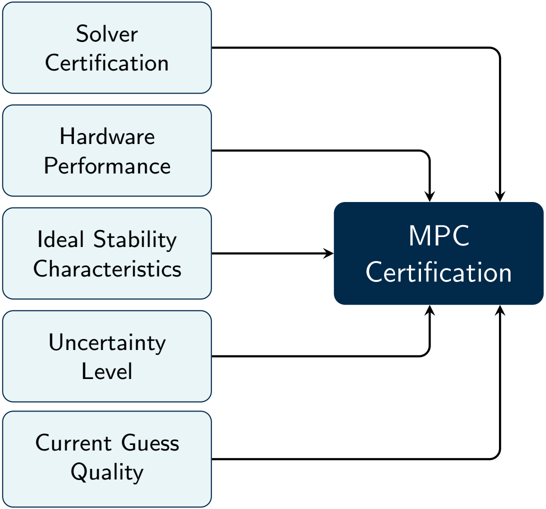
Easier to include in stability analysis

Solver certification

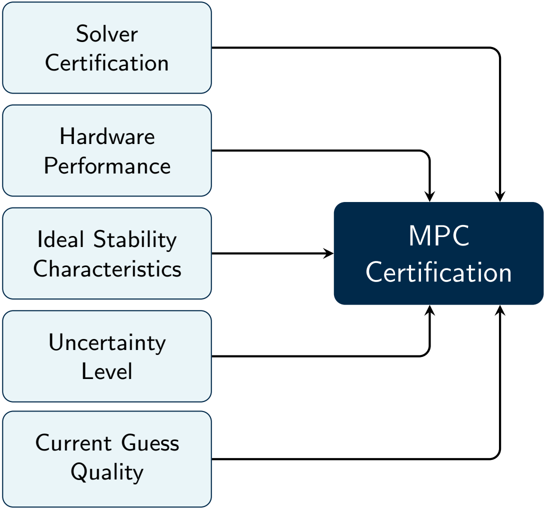
≠

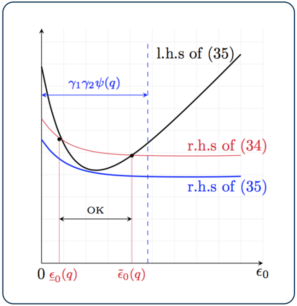
MPC Certification


Solver Certification
Hardware Performance
Ideal Stability Characteristics
Uncertainty Level
Current Guess Quality
MPC Certification




MA. From Certification of Algorithms To Certified MPC. NMPC2015, Seville.




MA. From Certification of Algorithms To Certified MPC. NMPC2015, Seville.

## Conclusion

**Certified** Real-time MPC needs **Co-Design** approach involving:

- **Carefully** chosen Certified **Solver**
- **Carefully** designed **MPC Formulation**
- **Carefully** chosen **embedded computation** facility
- **Carefully** characterized **uncertainties** and set-point **dynamics**
- **Carefully** chosen **initialization rule**

Remember! MPC was first successful, theory only followed . . .

# Acknowledgment


### PhD students

R. Amari, F. Bonne,
F. Clavel, C. Dedouits, A. Hably,
M. Y. Lamoudi, H. Mesnage,
N. Marchand, A. Murilo, P. Pflaum.

### Industrial Partners

P. Tona (**IFPEN**)
N. Perrissin-Fabert (**ALSTOM**)
P. Bonnay (**CEA-INAC-SBT**)
P. Béguery, C. Le Pape (**SCHNEIDER ELECTRIC**)
P. D. Gualino (**SCHNEIDER ELECTRIC**)

### Academic Partners

G. Bornard,
L. Del Re, P. Ortner, R. Furhapter
N. Sheibat-Othman, S. Othman
J. P. Corriou, F. Boyer.

### Institutions

CNRS
ANR
Grenoble-Inp
Université Jospeh Fourier

### GIPSA-lab staff

M. R. Alfara, P. Bellemain, G. Buche, Ch. Bulfone, O. Chabert, M. Di-Maria, J. Dumon, A. Fradin, E. Genin, C. Mendes, V. Messina, A. Mokhtari, S. Noguera, J. M. Thiriet,