MPC Design under time shortage

Lessons from case-studies …

Mazen Alamir

CNRS - University of Grenoble Alpes, France.

Outline

MPC under Time Shortage

Recall on MPC

Real-time MPC?

Simplification

- Parametrization
- Observer
- Formulation

Distribution over time

- State-dependent sampling
- Co-design S/H

Distribution over space

- Fixed-point Hierarchical MPC
- GPU-based MPC

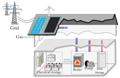
AMT

Cryogenics

Engine

Missile

Energy



Suspension

Disclaim

- A personal point of view
- A pragmatic (~~Non dogmatic~~) stand
- Set of reflexes/heuristics rather than ready to use recipes
- Collected mostly on real-life challenging problems
- Take home messages ☺
- Examples are only sketched to focus on the conveyed idea
- Only the final big picture really matters

Disclaim

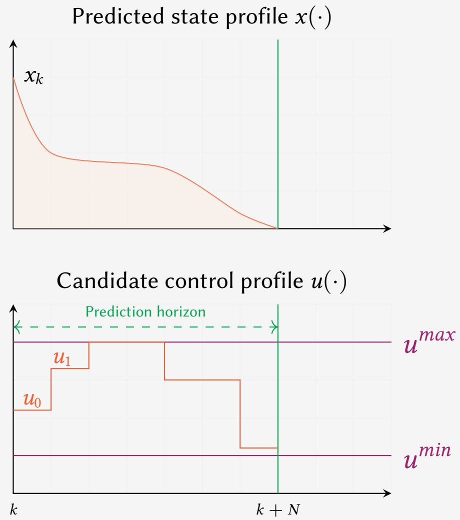
- A personal point of view
- A pragmatic (~~Non dogmatic~~) stand
- Set of reflexes/heuristics rather than ready to use recipes
- Collected mostly on real-life challenging problems
- Take home messages ☺
- Examples are only sketched to focus on the conveyed idea
- Only the final big picture really matters

- Missing although related items ☹:
  - Connection with probabilistic certification
  - Stochastic MPC
  - → [See plenary at the 2019-Colombian Control Conference]

(http://www.mazenalamir.fr)

Outline

Model Predictive Control (MPC): Definition



∀ candidate control sequence

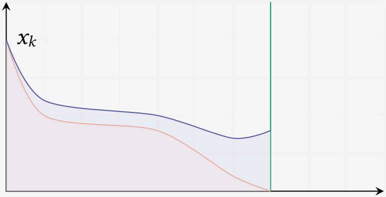
$$\boldsymbol{u}_k := (u_0, \dots, u_{N-1})$$

a cost can be predicted

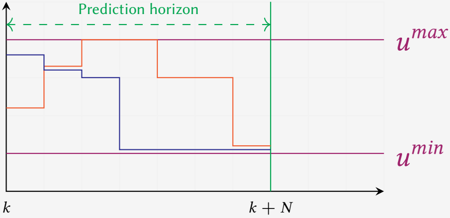
$$\mathcal{J}(x_k, \boldsymbol{u}_k)$$

Model Predictive Control (MPC): Definition

Predicted state profile $x(\cdot)$



Candidate control profile $u(\cdot)$



$\forall$ candidate control sequence

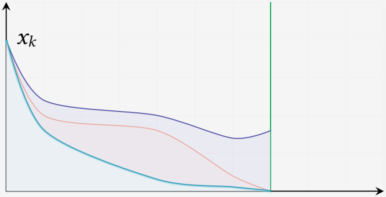
$$\boldsymbol{u}_k := (u_0, \dots, u_{N-1})$$

a cost can be predicted

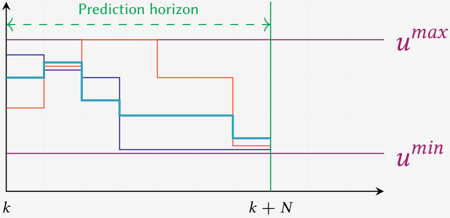
$$\mathcal{J}(x_k, \boldsymbol{u}_k)$$

Model Predictive Control (MPC): Definition

Predicted state profile $x(\cdot)$



Candidate control profile $u(\cdot)$



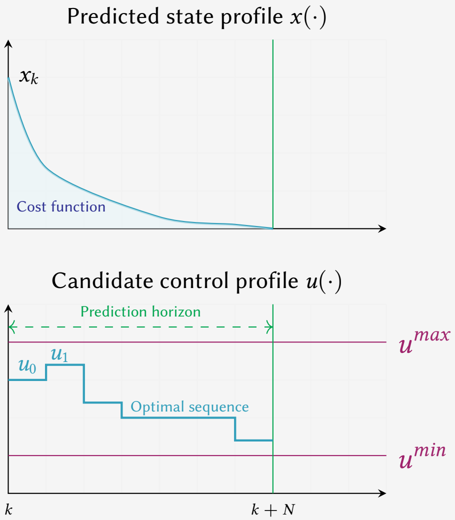
$\forall$ candidate control sequence

$$\boldsymbol{u}_k := (u_0, \dots, u_{N-1})$$

a cost can be predicted

$$\mathcal{J}(x_k, \boldsymbol{u}_k)$$

Model Predictive Control (MPC): Definition



$\forall$ candidate control sequence

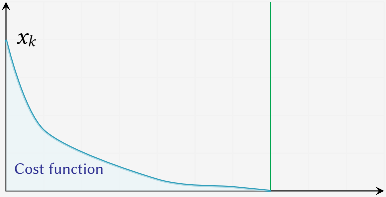
$$\boldsymbol{u}_k := (u_0, \dots, u_{N-1})$$

a cost can be predicted

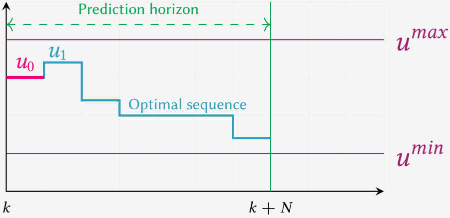
$$\mathcal{J}(x_k, \boldsymbol{u}_k)$$

Model Predictive Control (MPC): Definition

Predicted state profile $x(\cdot)$



Candidate control profile $u(\cdot)$



$\forall$ candidate control sequence

$$\boldsymbol{u}_k := (u_0, \dots, u_{N-1})$$

a cost can be predicted

$$\mathcal{J}(x_k, \boldsymbol{u}_k)$$

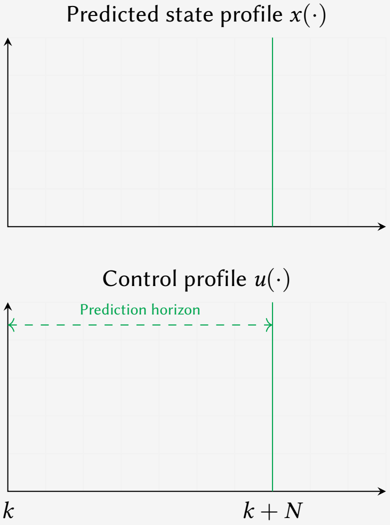
MPC feedback

$$u_k = u_0^{opt}(x_k)$$

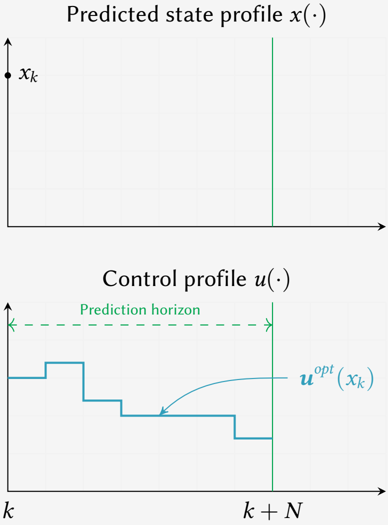
where

$$\mathcal{J}(x_k, \boldsymbol{u}^{opt}) = \min_{\boldsymbol{u} \in \mathbb{U}(x)} \mathcal{J}(x_k, \boldsymbol{u})$$

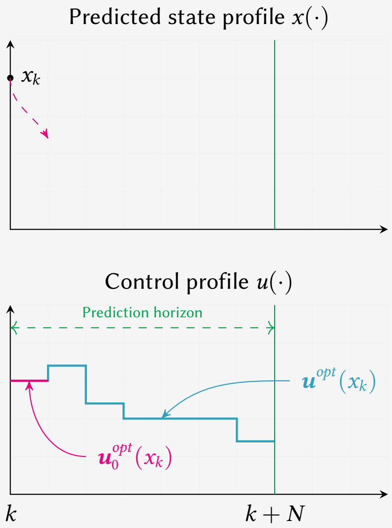
Model Predictive Control (MPC): Definition



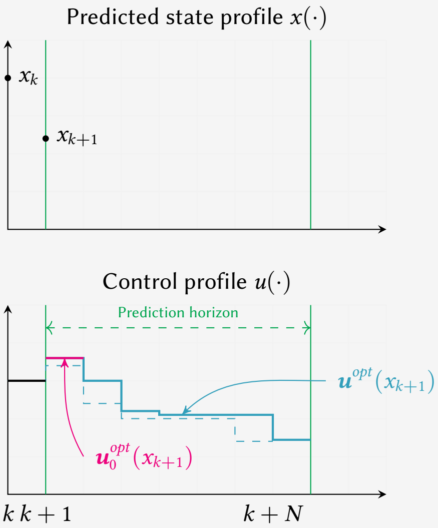
Model Predictive Control (MPC): Definition



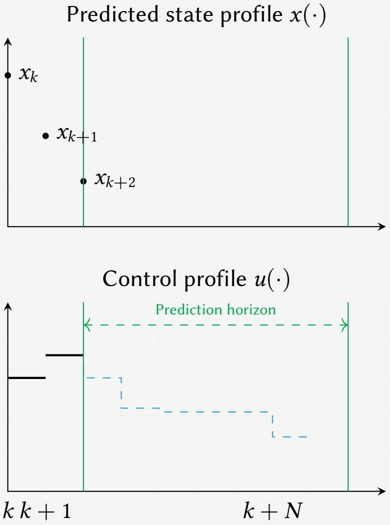
Model Predictive Control (MPC): Definition



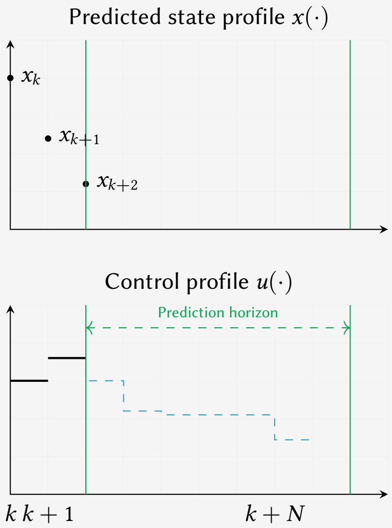
Model Predictive Control (MPC): Definition



Model Predictive Control (MPC): Definition



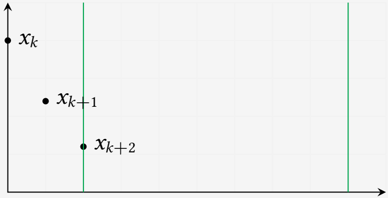
Model Predictive Control (MPC): Definition



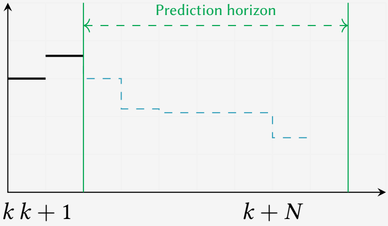
- At each **updating instant**
- The prediction horizon is shifted
- **A new optimization problem is defined**
- and **solved**
- The first action is applied
- until the next **updating instants**
- The process is repeated

Model Predictive Control (MPC): Definition

Predicted state profile $x(\cdot)$



Control profile $u(\cdot)$

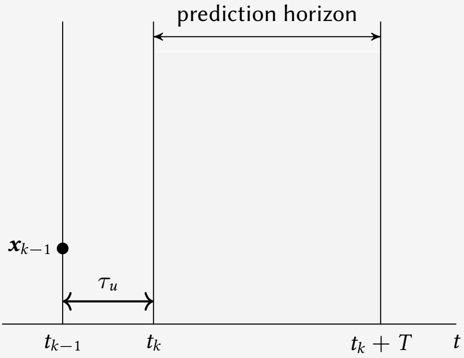


- At each **updating instant**
- The prediction horizon is shifted
- **A new optimization problem is defined**
- and **solved**
- The first action is applied
- until the next **updating instants**
- The process is repeated

→ No computation time in this statement.

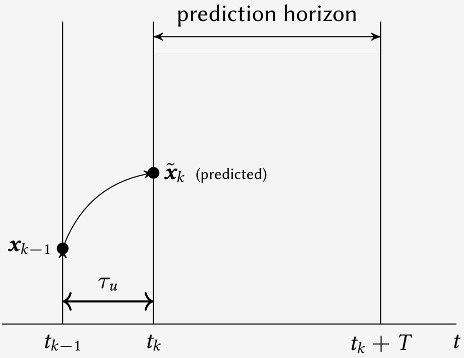
MPC: Real-time setting

Let p denote the decision variable: $\boldsymbol{u} = \mathcal{U}(p)$



MPC: Real-time setting

Let p denote the decision variable: $\boldsymbol{u} = \mathcal{U}(p)$

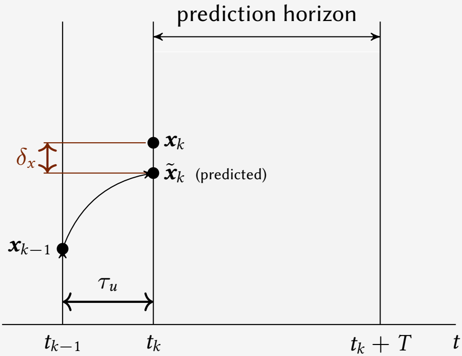


1. Predict $\tilde{\boldsymbol{x}}_k$
2. During $[t_{k-1}, t_k]$

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

MPC: Real-time setting

Let p denote the decision variable: $\boldsymbol{u} = \mathcal{U}(p)$



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

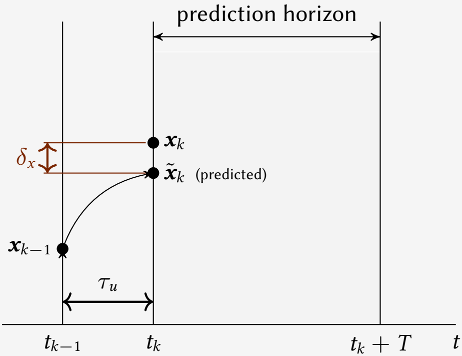
   Compute $\hat{p}(\tilde{\boldsymbol{x}}_k)$ [and $\dfrac{\partial \hat{p}_k}{\partial \boldsymbol{x}}$]

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

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

MPC: Real-time setting

Let p denote the decision variable: $\boldsymbol{u} = \mathcal{U}(p)$



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

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

   [preparation step]
3. Once $\boldsymbol{x}_k$ is available:

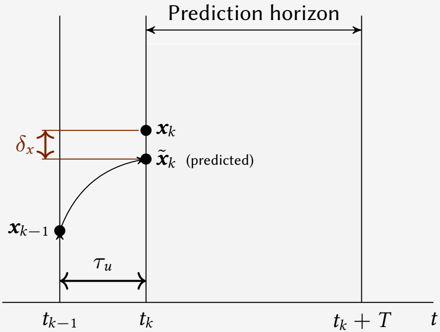
$$\hat{p}_k \leftarrow \hat{p}(\tilde{\boldsymbol{x}}_k) + \left[\frac{\partial \hat{p}_k}{\partial \boldsymbol{x}}\right] \cdot \delta_x$$

[feedback step]

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

Zavala and Biegler. Automatica (2009)

Time-shortage in MPC

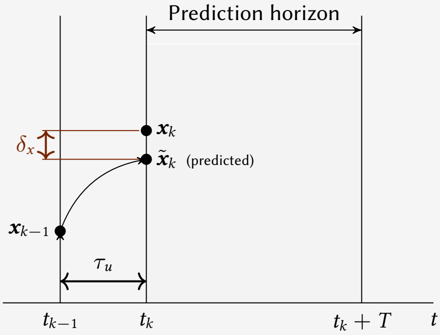


τ_u is the time between two control updating

No feedback during τ_u

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

Time-shortage in MPC



τ_u is the time between two control updating

No feedback during τ_u

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

Time shortage

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

Keywords involved in MPC Time shortage

Time shortage

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

Formulation

Prediction horizon
Constraints
Parametrization

Suboptimality

Uniform sampling
State-dependent

Architecture

Centralized
Hierarchical
Parallel

Hardware

PLC
Work Station
8-bit processor

Algorithm

IP
SQP
Gradient

Keywords involved in MPC Time shortage

Time shortage

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

Formulation

Prediction horizon
Constraints
Parametrization

Suboptimality

Uniform sampling
State-dependent

Architecture

Centralized
Hierarchical
Parallel

Hardware

PLC
Work Station
8-bit processor

Algorithm

IP
SQP
Gradient

Simplification

- Parametrization
- Observer
- Formulation

Distribution over time

- State-dependent sampling
- Co-design S/H

Distribution over space

- Fixed-point Hierarchical MPC
- GPU-based MPC

Keywords involved in MPC Time shortage

Time shortage

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

Formulation

Prediction horizon
Constraints
Parametrization

Suboptimality

Uniform sampling
State-dependent

Architecture

Centralized
Hierarchical
Parallel

Hardware

PLC
Work Station
8-bit processor

Algorithm

IP
SQP
Gradient

Simplification

- Parametrization
- Observer
- Formulation

Distribution over time

- State-dependent sampling
- Co-design S/H

Distribution over space

- Fixed-point Hierarchical MPC
- GPU-based MPC

Keywords involved in MPC Time shortage

Time shortage

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

Formulation

Prediction horizon
Constraints
Parametrization

Suboptimality

Uniform sampling
State-dependent

Architecture

Centralized
Hierarchical
Parallel

Hardware

PLC
Work Station
8-bit processor

Algorithm

IP
SQP
Gradient

Simplification

- Parametrization
- Observer
- Formulation

Distribution over time

- State-dependent sampling
- Co-design S/H

Distribution over space

- Fixed-point Hierarchical MPC
- GPU-based MPC

Keywords involved in MPC Time shortage

Time shortage

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

Formulation

Prediction horizon
Constraints
Parametrization

Suboptimality

Uniform sampling
State-dependent

Architecture

Centralized
Hierarchical
Parallel

Hardware

PLC
Work Station
8-bit processor

Algorithm

IP
SQP
Gradient

Simplification

- Parametrization
- Observer
- Formulation

Distribution over time

- State-dependent sampling
- Co-design S/H

Distribution over space

- Fixed-point Hierarchical MPC
- GPU-based MPC

Keywords involved in MPC Time shortage

Time shortage

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

Formulation

Prediction horizon
Constraints
Parametrization

Suboptimality

Uniform sampling
State-dependent

Architecture

Centralized
Hierarchical
Parallel

Hardware

PLC
Work Station
8-bit processor

Algorithm

IP
SQP
Gradient

Simplification

- Parametrization
- Observer
- Formulation

Distribution over time

- State-dependent sampling
- Co-design S/H

Distribution over space

- Fixed-point Hierarchical MPC
- GPU-based MPC

Keywords involved in MPC Time shortage

Time shortage

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

Formulation

Prediction horizon
Constraints
Parametrization

Suboptimality

Uniform sampling
State-dependent

Architecture

Centralized
Hierarchical
Parallel

Hardware

PLC
Work Station
8-bit processor

Algorithm

IP
SQP
Gradient

Simplification

- Parametrization
- Observer
- Formulation

Distribution over time

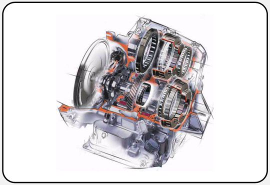
- State-dependent sampling
- Co-design S/H

Distribution over space

- Fixed-point Hierarchical MPC
- GPU-based MPC

Case study 1

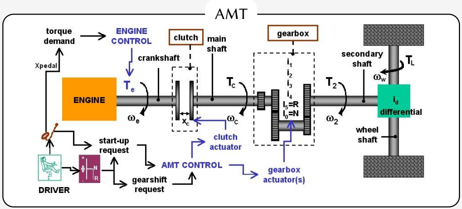
Automated manual transmission



Problem statement

Control Objective

- Smooth clutch $\omega_{sl} = \omega_e - \omega_c \to 0$
- Transparency (torque $\leftrightarrow$ Pedal Position)
  $\omega_e^{ref} = max\Big\{\omega_e^0, \mathcal{T}^{-1}\big(T_e^d(X_{pedal}, \omega_e)\big)\Big\}$



To summarize

Multivariable constrained tracking / sampling period = 1 ms.

Uncertainties

Inertias, load torque, clutch characteristics, local controllers behavior, road profile…

M. Alamir, A. Murilo, R. Amari, P. Tona, R. Furhapter and P. Ortner, On the use of parameterized NMPC in real-time automotive control. In Automotive Model Predictive Control. L. Del Re (Ed). Lecture Notes in Control and Information Sciences, Springer-Verlag, (2009)

Problem statement

Control Objective

- Smooth clutch $\omega_{sl} = \omega_e - \omega_c \to 0$
- Transparency (torque $\leftrightarrow$ Pedal Position)
  $\omega_e^{ref} = max\left\{\omega_e^0, \mathcal{T}^{-1}\left(T_e^d(X_{pedal}, \omega_e)\right)\right\}$

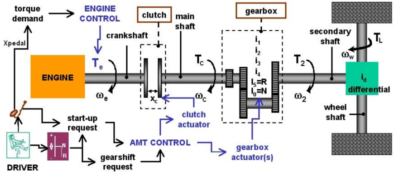
To summarize

Multivariable constrained tracking / sampling period = 1 ms.

Uncertainties

Inertias, load torque, clutch characteristics, local controllers behavior, road profile…

AMT



$$\mathcal{J}_e\dot{\omega}_e = T_e - T_c(x_c)$$

$$[\mathcal{J}_c + \mathcal{J}_{eq}(i_g, i_d)]\dot{\omega}_c = T_c(x_c) - \\ - \frac{1}{i_g\, i_d}\left[k_{tw}\theta_{cw} + \beta_{tw}\left(\frac{\omega_c}{i_g\, i_d} - \omega_w\right)\right]$$

$$\mathcal{J}_w\dot{\omega}_w = k_{tw}\theta_{cw} + \beta_{tw}\left(\frac{\omega_c}{i_g\, i_d} - \omega_w\right) - T_L(\omega_w)$$

$$\dot{\theta}_{cw} = \frac{\omega_c}{i_g\, i_d} - \omega_w$$

Problem statement

Control Objective

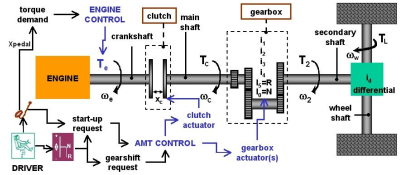
- Smooth clutch $\omega_{sl} = \omega_e - \omega_c \to 0$
- Transparency
  (torque $\leftrightarrow$ Pedal Position)
  $\omega_e^{ref} = max\Big\{\omega_e^0, \mathcal{T}^{-1}\big(T_e^d(X_{pedal}, \omega_e)\big)\Big\}$

> **To summarize**
>
> Multivariable constrained tracking / sampling period = 1 ms.

Uncertainties

Inertias, load torque, clutch characteristics, local controllers behavior, road profile...

AMT



$$\mathcal{J}_e\dot{\omega}_e = \cancel{T_e} - \cancel{T_c(x_c)}$$

$$[\mathcal{J}_c + \mathcal{J}_{eq}(i_g, i_d)]\dot{\omega}_c = \cancel{T_c(x_c)} -$$

$$- \frac{1}{i_g\, i_d}\Big[k_{tw}\theta_{cw} + \beta_{tw}\big(\frac{\omega_c}{i_g\, i_d} - \omega_w\big)\Big]$$

$$\mathcal{J}_w\dot{\omega}_w = k_{tw}\theta_{cw} + \beta_{tw}\big(\frac{\omega_c}{i_g\, i_d} - \omega_w\big) - T_L(\omega_w)$$

$$\dot{\theta}_{cw} = \frac{\omega_c}{i_g\, i_d} - \omega_w$$

Problem statement

Control Objective

- Smooth clutch $\omega_{sl} = \omega_e - \omega_c \to 0$
- Transparency
  (torque $\leftrightarrow$ Pedal Position)
  $\omega_e^{ref} = max\left\{\omega_e^0, \mathcal{T}^{-1}\left(T_e^d(X_{pedal}, \omega_e)\right)\right\}$

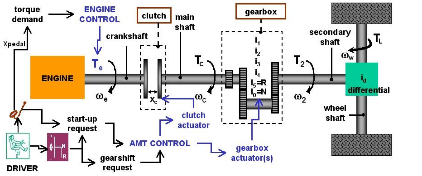
To summarize

Multivariable constrained tracking / sampling period = 1 ms.

Uncertainties

Inertias, load torque, clutch characteristics, local controllers behavior, road profile…

AMT



$$\mathcal{J}_e\dot{\omega}_e = T_e^{sp} - T_c^{sp} - \delta_e$$

$$[\mathcal{J}_c + \mathcal{J}_{eq}(i_g, i_d)]\dot{\omega}_c = T_c^{sp} - \delta_c - \frac{1}{i_g\, i_d}\left[k_{tw}\theta_{cw} + \beta_{tw}\left(\frac{\omega_c}{i_g\, i_d} - \omega_w\right)\right]$$

$$\mathcal{J}_w\dot{\omega}_w = k_{tw}\theta_{cw} + \beta_{tw}\left(\frac{\omega_c}{i_g\, i_d} - \omega_w\right) - T_L(\omega_w)$$

$$\dot{\theta}_{cw} = \frac{\omega_c}{i_g\, i_d} - \omega_w$$

Problem statement

Control Objective

- Smooth clutch $\omega_{sl} = \omega_e - \omega_c \to 0$
- Transparency (torque $\leftrightarrow$ Pedal Position)

$$\omega_e^{ref} = max\left\{\omega_e^0, \mathcal{T}^{-1}\left(T_e^d\left(X_{pedal}, \omega_e\right)\right)\right\}$$

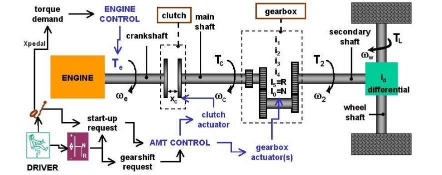
To summarize

Multivariable constrained tracking / sampling period = 1 ms.

Uncertainties

Inertias, load torque, clutch characteristics, local controllers behavior, road profile…

AMT



$$\mathcal{J}_e\dot{\omega}_e = T_e^{sp} - T_c^{sp} - \delta_e$$

$$[\mathcal{J}_c + \mathcal{J}_{eq}(i_g, i_d)]\dot{\omega}_c = T_c^{sp} - \delta_c$$

$$- \frac{1}{i_g\, i_d}\left[k_{tw}\theta_{cw} + \beta_{tw}\left(\frac{\omega_c}{i_g\, i_d} - \omega_w\right)\right]$$

$$\mathcal{J}_w\dot{\omega}_w = k_{tw}\theta_{cw} + \beta_{tw}\left(\frac{\omega_c}{i_g\, i_d} - \omega_w\right) - T_L(\omega_w)$$

$$\dot{\theta}_{cw} = \frac{\omega_c}{i_g\, i_d} - \omega_w$$

Problem statement

Control Objective

- Smooth clutch $\omega_{sl} = \omega_e - \omega_c \to 0$
- Transparency (torque $\leftrightarrow$ Pedal Position)

$$\omega_e^{ref} = max\left\{\omega_e^0, \mathcal{T}^{-1}\left(T_e^d(X_{pedal}, \omega_e)\right)\right\}$$

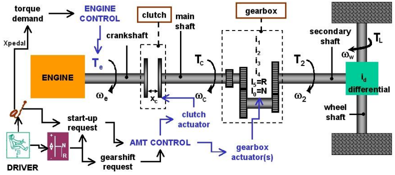
To summarize

Multivariable constrained tracking / sampling period = 1 ms.

Uncertainties

Inertias, load torque, clutch characteristics, local controllers behavior, road profile…

AMT



$$\mathcal{J}_e\dot{\omega}_e = T_e^{sp} - T_c^{sp} - \delta_e$$

$$[\mathcal{J}_c + \mathcal{J}_{eq}(i_g, i_d)]\dot{\omega}_c = T_c^{sp} - \delta_c$$

$$\mathcal{J}_w\dot{\omega}_w = k_{tw}\theta_{cw} + \beta_{tw}\left(\frac{\omega_c}{i_g\, i_d} - \omega_w\right) - T_L(\omega_w)$$

$$\dot{\theta}_{cw} = \frac{\omega_c}{i_g\, i_d} - \omega_w$$

Problem statement

Control Objective

- Smooth clutch $\omega_{sl} = \omega_e - \omega_c \to 0$
- Transparency
  (torque $\leftrightarrow$ Pedal Position)
  $\omega_e^{ref} = max\left\{\omega_e^0, \mathcal{T}^{-1}\left(T_e^d(X_{pedal}, \omega_e)\right)\right\}$

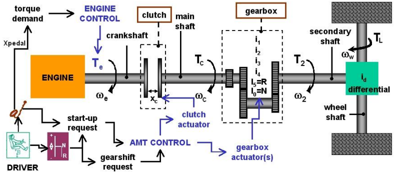
To summarize

Multivariable constrained tracking / sampling period = 1 ms.

Uncertainties

Inertias, load torque, clutch characteristics, local controllers behavior, road profile...

AMT



$$\mathcal{J}_e\dot{\omega}_e = T_e^{sp} - T_c^{sp} - \delta_e$$

$$[\mathcal{J}_c + \mathcal{J}_{eq}(i_g, i_d)]\dot{\omega}_c = T_c^{sp} - \delta_c$$

Problem statement

Control Objective

- Smooth clutch $\omega_{sl} = \omega_e - \omega_c \to 0$
- Transparency (torque $\leftrightarrow$ Pedal Position)

$$\omega_e^{ref} = max\left\{\omega_e^0, \mathcal{T}^{-1}\left(T_e^d\left(X_{pedal}, \omega_e\right)\right)\right\}$$

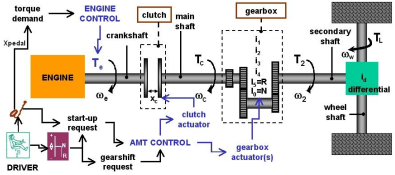
To summarize

Multivariable constrained tracking / sampling period = 1 ms.

Uncertainties

Inertias, load torque, clutch characteristics, local controllers behavior, road profile...

AMT



$$\mathcal{J}_e\dot{\omega}_e = u_1 - u_2 - \delta_e$$
$$[\mathcal{J}_c + \mathcal{J}_{eq}(i_g, i_d)]\dot{\omega}_c = u_2 - \delta_c$$
$$\dot{\delta}_e = 0$$
$$\dot{\delta}_c = 0$$

Problem statement

Control Objective

- Smooth clutch $\omega_{sl} = \omega_e - \omega_c \to 0$
- Transparency
  (torque $\leftrightarrow$ Pedal Position)
  $\omega_e^{ref} = max\left\{\omega_e^0, \mathcal{T}^{-1}\left(T_e^d\left(X_{pedal}, \omega_e\right)\right)\right\}$

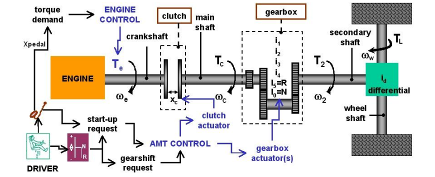
To summarize

Multivariable constrained tracking / sampling period = 1 ms.

Uncertainties

Inertias, load torque, clutch characteristics, local controllers behavior, road profile...

AMT



$$\mathcal{J}_e\dot{\omega}_e = u_1 - u_2 - \delta_e$$
$$[\mathcal{J}_c + \mathcal{J}_{eq}(i_g, i_d)]\dot{\omega}_c = u_2 - \delta_c$$
$$\dot{\delta}_e = 0$$
$$\dot{\delta}_c = 0$$

$\rightarrow$ Observer-based parameterized MPC

One single degree of freedom

Problem statement

Control Objective

- Smooth clutch $\omega_{sl} = \omega_e - \omega_c \to 0$
- Transparency
  (torque $\leftrightarrow$ Pedal Position)
  $\omega_e^{ref} = max\left\{\omega_e^0, \mathcal{T}^{-1}\left(T_e^d\left(X_{pedal}, \omega_e\right)\right)\right\}$

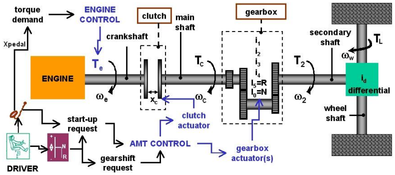
To summarize

Multivariable constrained tracking / sampling period = 1 ms.

Uncertainties

Inertias, load torque, clutch characteristics, local controllers behavior, road profile…

AMT



$$\mathcal{J}_e\dot{\omega}_e = u_1 - u_2 - \delta_e$$

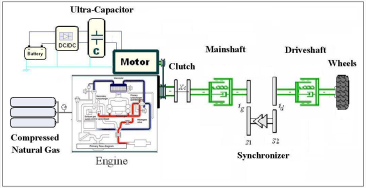
$$[\mathcal{J}_c + \mathcal{J}_{eq}(i_g, i_d)]\dot{\omega}_c = u_2 - \delta_c$$

$$\dot{\delta}_e = 0$$

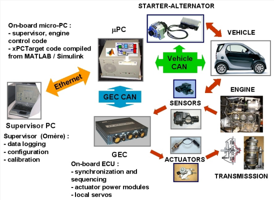
$$\dot{\delta}_c = 0$$

M. Alamir, A. Murilo, R. Amari, P. Tona, R. Furhapter and P. Ortner, On the use of parameterized NMPC in real-time automotive control. In Automotive Model Predictive Control. L. Del Re (Ed). Lecture Notes in Control and Information Sciences, Springer-Verlag, (2009)

The IFPEN SMART demo architecture



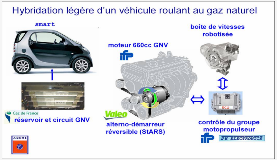
The mild-Hybrid powertrain of the VEHGAN demo-car



VEHGAN on-board Control System

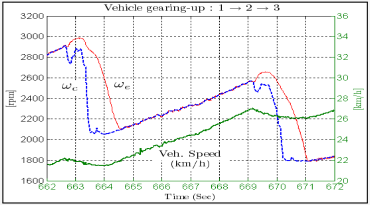
R. Amari (IFPEN)

P. Tona (IFPEN)

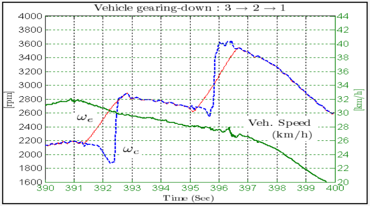


Demo SMART car

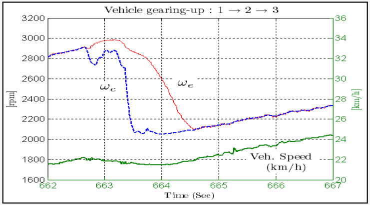
Experimental Results



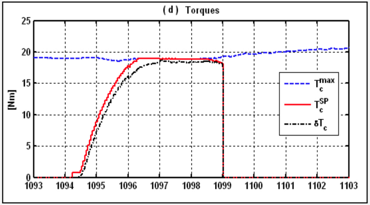
Parameterized NMPC - Gearing up



Parameterized NMPC - Gearing down

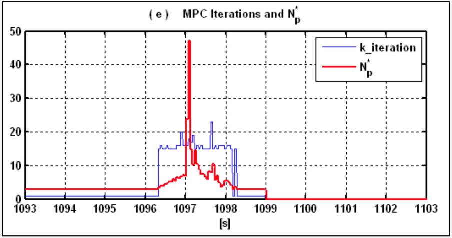


Parameterized NMPC - Gearing up, 1 → 2

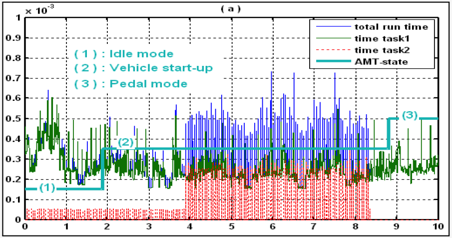


Parametrized NMPC: Constraints activation

Experimental Results



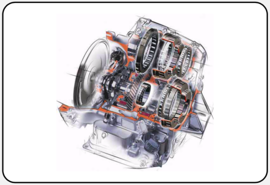
Parametrized NMPC: Closed-loop evolution of the decision variable



Parameterized NMPC (Task 2) - CPU time (300μsec in 2006 !!)

Case study 1

Automated manual transmission



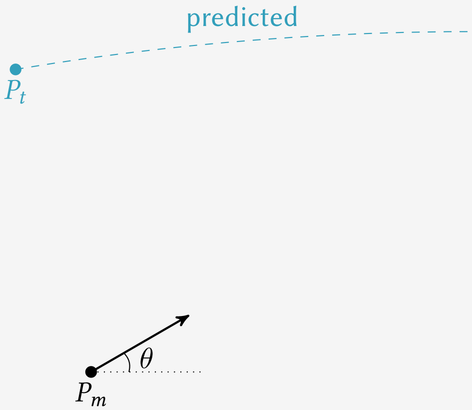
Take away

- Simplify
- Gather **complexity** in observable blocks
- Design **observer**
- **Only then**, design NMPC.

Case study 2

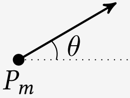
Interception missile

Interception of moving target

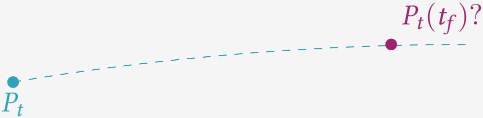


Interception of moving target

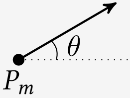
- Optimal Rendez-vous problem
- Free final time
- Highly uncertain



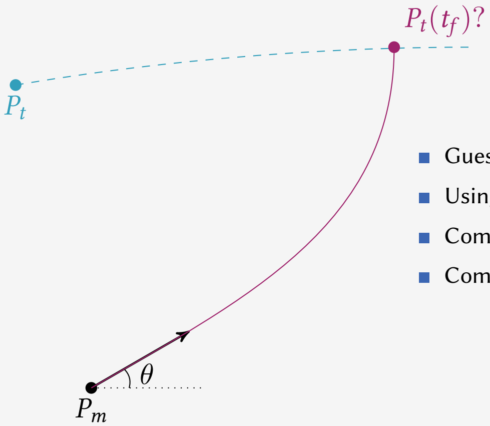
Interception of moving target



- Guess some t_f.
- Using presumed target trajectory
- Compute the corresponding $P_t(t_f)$

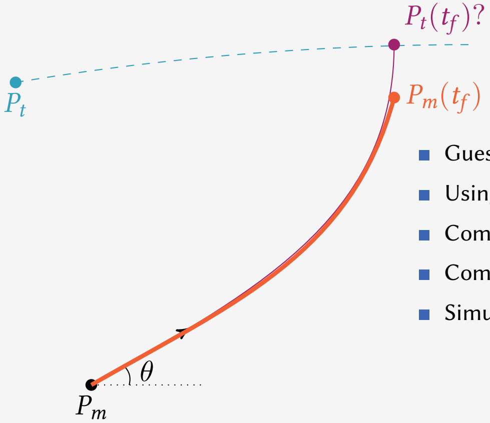


Interception of moving target



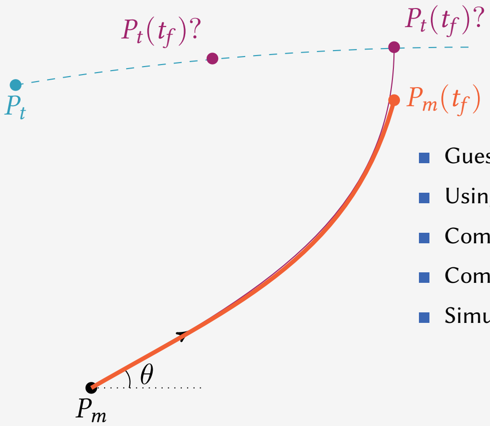
- Guess some t_f.
- Using presumed target trajectory
- Compute the corresponding $P_t(t_f)$
- Compute the parabola (respecting initial conditions)

Interception of moving target



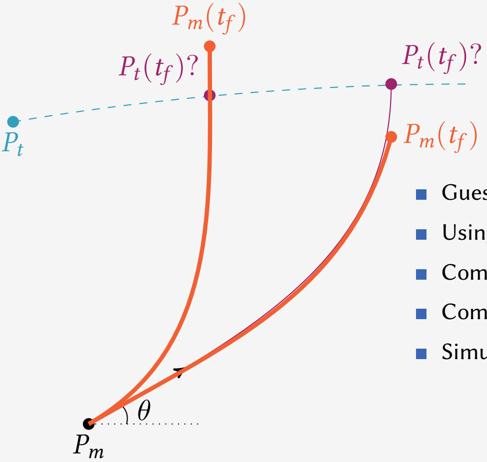
- Guess some t_f.
- Using presumed target trajectory
- Compute the corresponding $P_t(t_f)$
- Compute the parabola (respecting initial conditions)
- Simulate parabola tracking

Interception of moving target



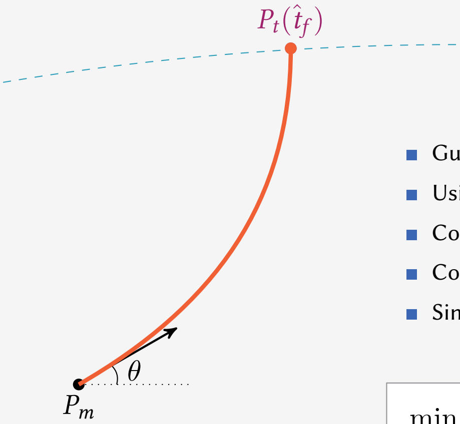
- Guess some t_f.
- Using presumed target trajectory
- Compute the corresponding $P_t(t_f)$
- Compute the parabola (respecting initial conditions)
- Simulate parabola tracking

Interception of moving target



- Guess some t_f.
- Using presumed target trajectory
- Compute the corresponding $P_t(t_f)$
- Compute the parabola (respecting initial conditions)
- Simulate parabola tracking

Interception of moving target

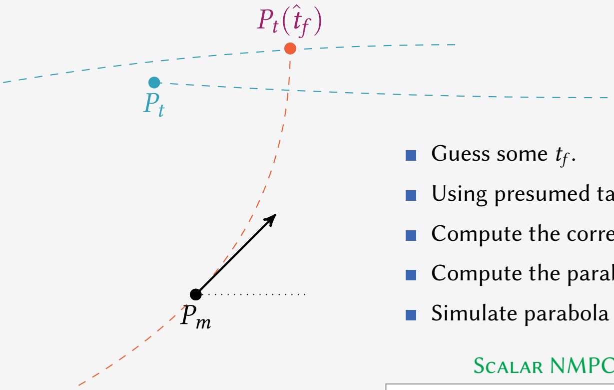


- Guess some t_f.
- Using presumed target trajectory
- Compute the corresponding $P_t(t_f)$
- Compute the parabola (respecting initial conditions)
- Simulate parabola tracking

Scalar NMPC

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

Interception of moving target

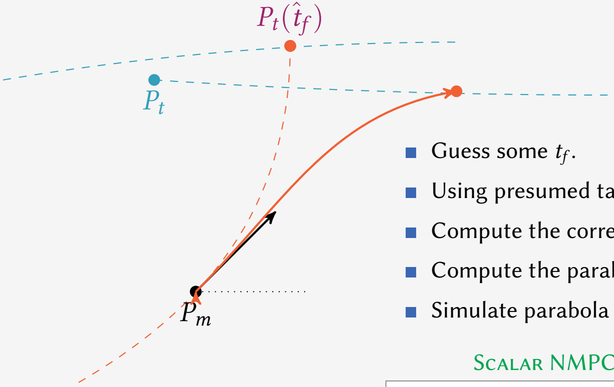


- Guess some t_f.
- Using presumed target trajectory
- Compute the corresponding $P_t(t_f)$
- Compute the parabola (respecting initial conditions)
- Simulate parabola tracking

SCALAR NMPC

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

Interception of moving target

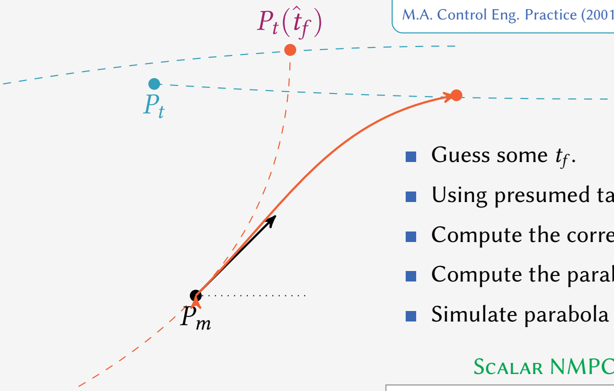


- Guess some t_f.
- Using presumed target trajectory
- Compute the corresponding $P_t(t_f)$
- Compute the parabola (respecting initial conditions)
- Simulate parabola tracking

Scalar NMPC

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

Interception of moving target



Full solution	40	ms	
Fast solution	0.4	ms	One Gradient step per updating period

M.A. Control Eng. Practice (2001)

- Guess some t_f.
- Using presumed target trajectory
- Compute the corresponding $P_t(t_f)$
- Compute the parabola (respecting initial conditions)
- Simulate parabola tracking

Scalar NMPC

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

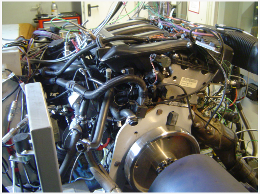
Case study 2

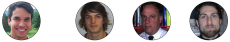
Interception missile

Take away

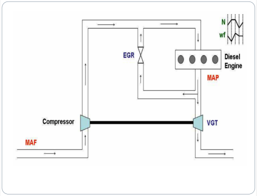
- Accept a priori **sub-optimality**
- rely on high frequency updating
- Carefully **choose your decision variable**

Case study 3
Diesel Engines



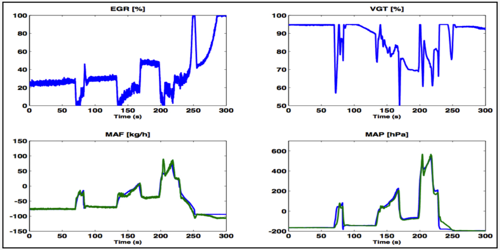


A. Murilo P. Ortner, L. Del Re, R. Furhapter (Linz)



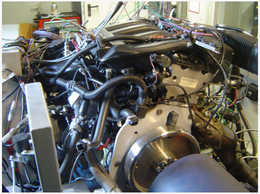
- Nonlinear 8-dimensional model
- non minimum phase dynamics
- 10 msec updating period

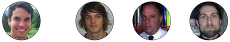
✓ **2 d.o.f parametrization**



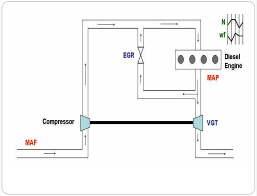
A. Murilo et al. A General NMPC Framework for a Diesel Engine Air Path. International Journal of Control, Volume 87, No 10, pp 2194-2207, 2014..

Case study 3
Diesel Engines



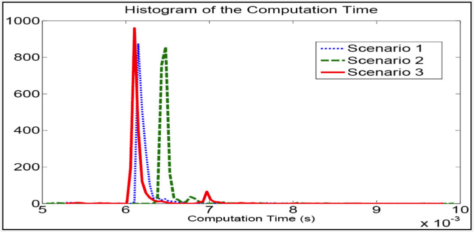


A. Murilo P. Ortner, L. Del Re, R. Furhapter (Linz)



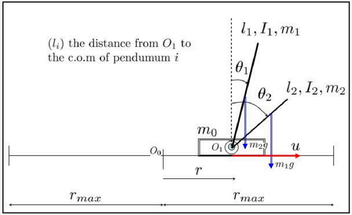
- Nonlinear 8-dimensional model
- non minimum phase dynamics
- 10 msec updating period

✓ **2 d.o.f parametrization**



A. Murilo et al. A General NMPC Framework for a Diesel Engine Air Path. International Journal of Control, Volume 87, No 10, pp 2194-2207, 2014..

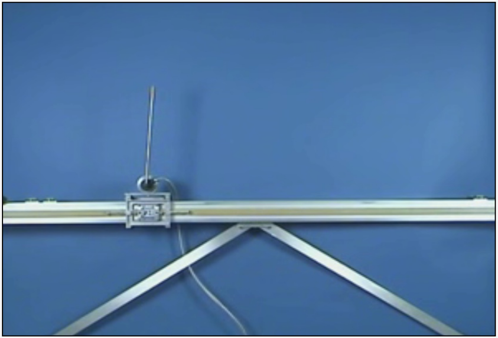
Case study 4
Twin pendulums

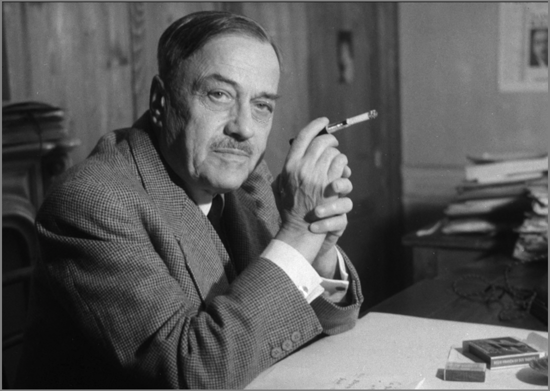


MA. and Murilo, A. Swing-up and stabilization of a Twin-Pendulum under state and control constraints by fast NMPC scheme. Automatica, Vol. 44, 2008.

- Nonlinear 6-dimensional model
- highly unstable
- 20 msec updating period

✓ **Scalar parametrization**



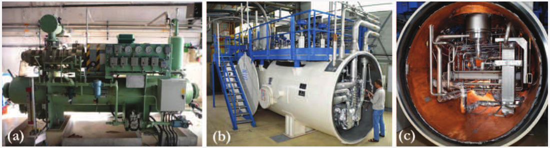


The common-sense solution is the last specialists generally think of!

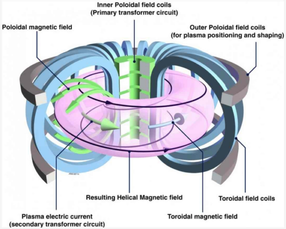
Bernard Grasset (French editor)

Case study 5

Cryogenic refrigerators



Cryogenic Refrigerators - Problem statement



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

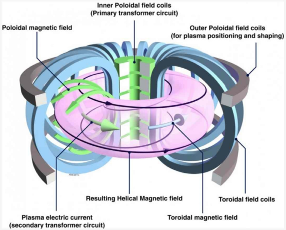
Why?

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

F. Bonne (CEA)

P. Bonnay (CEA)

Cryogenic Refrigerators - Problem statement



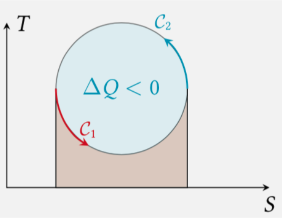
Source: https://www.euro-fusion.org

Why?

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

F. Bonne (CEA)

P. Bonnay (CEA)

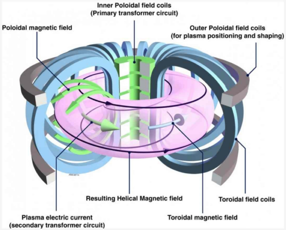


How?

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

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

Cryogenic Refrigerators - Problem statement



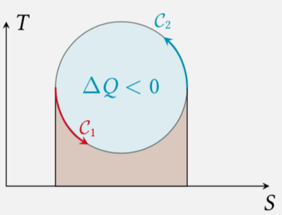
Source: https://www.euro-fusion.org

Why?

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

F. Bonne (CEA)

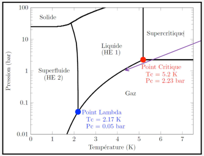
P. Bonnay (CEA)

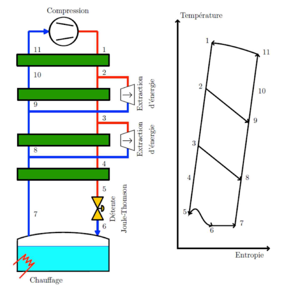


How?

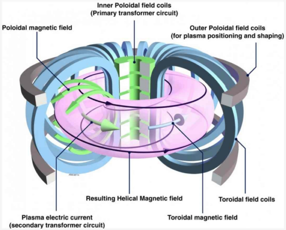
Force a thermodynamic fluid to make a counter-clock cycle in the (S, T)=(Entropy,Temperature) plan.

$$\int dQ = \underbrace{\int_{C_1} TdS}_{>0} + \underbrace{\int_{C_2} TdS}_{<<0}$$

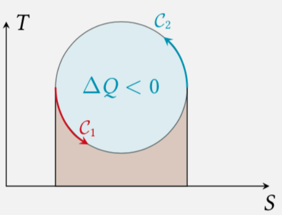


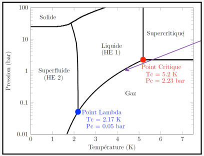


Cryogenic Refrigerators - Problem statement



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)

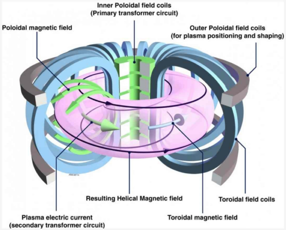
How?

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

$$\int dQ = \underbrace{\int_{C_1} TdS}_{>0} + \underbrace{\int_{C_2} TdS}_{<<0}$$

F. Bonne (CEA) P. Bonnay (CEA)

Cryogenic Refrigerators - Problem statement



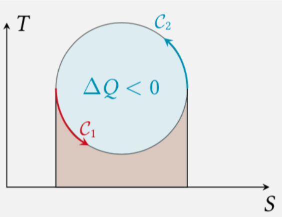
Source: https://www.euro-fusion.org

Why?

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

F. Bonne (CEA)

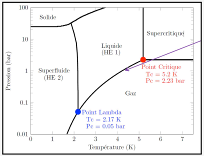
P. Bonnay (CEA)



How?

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

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



Model

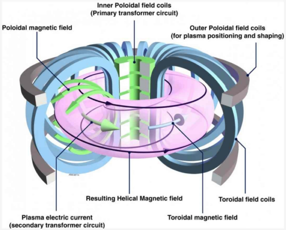
$$x^+ = Ax + Bu + Gw$$
$$y = Cx + Du$$

- $(x, u, w) \in \mathbb{R}^{24} \times \mathbb{R}^3 \times \mathbb{R}$
- w unpredictable heat pulse
- $u \in \mathbb{U}$ two valves+heating (bath)

Objective

Keep the station sustainable despite of unmeasured and unpredictable w.

Cryogenic Refrigerators - Problem statement



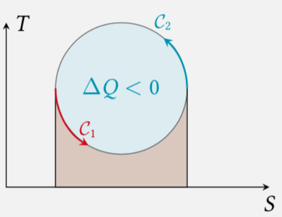
Source: https://www.euro-fusion.org

Why?

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

F. Bonne (CEA)

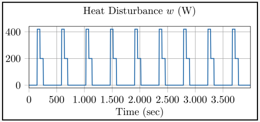
P. Bonnay (CEA)



How?

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

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



Model

$$x^+ = Ax + Bu + Gw$$
$$y = Cx + Du$$

- $(x, u, w) \in \mathbb{R}^{24} \times \mathbb{R}^3 \times \mathbb{R}$
- w unpredictable heat pulse
- $u \in \mathbb{U}$ two valves+heating (bath)

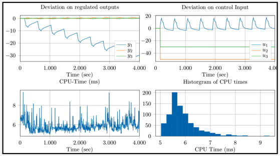
Objective

Keep the station sustainable despite of unmeasured and unpredictable w.

Cryogenic refrigerator: The impact of formulation

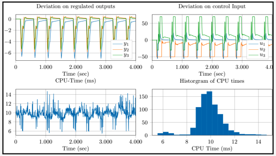
Cost function

$$\sum_{i=1}^{N}\Big[\|y_{k+i}\|^2_{Q_y} + \epsilon\|x_{k+i}\|^2\Big]$$



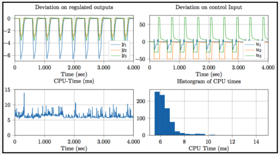
Cost function

$$\sum_{i=1}^{N}\Big[\|y_{k+i}\|^2_{Q_y} + \epsilon\|x_{k+i}\|^2\Big] + \epsilon_f\|x_{i+N}\|^2$$



Cost function

$$\sum_{i=1}^{N}\Big[\frac{i}{N}\Big]^m\Big[\|y_{k+i}\|^2_{Q_y} + \epsilon\|x_{k+i}\|^2\Big]$$



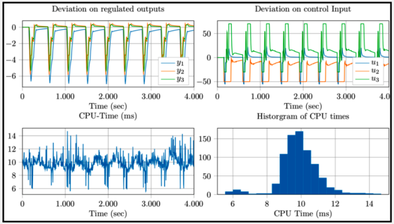
Optimization problems are solved using the GUROBI-Python-3 framework implemented on a Mac-PowerBook 2.9GHz (High Sierra OS version 10.13.3).

MA. Stability proof for nonlinear MPC design using monotonically increasing weighting profiles without terminal constraints. Automatica, Vol 87, pp. 455-459, 2018.

Cryogenic refrigerator: The impact of formulation

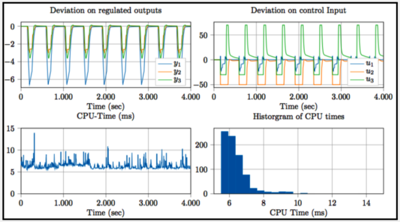
Cost function

$$\sum_{i=1}^{N}\left[\|y_{k+i}\|^2_{Q_y}+\epsilon\|x_{k+i}\|^2\right]+\epsilon_f\|x_{i+N}\|^2$$



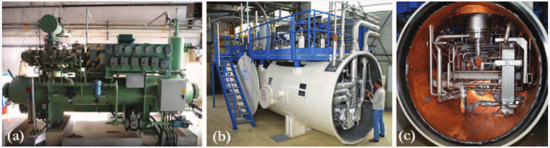
Cost function

$$\sum_{i=1}^{N}\left[\frac{i}{N}\right]^m\left[\|y_{k+i}\|^2_{Q_y}+\epsilon\|x_{k+i}\|^2\right]$$



Case study 5

Cryogenic refrigerators



Take away

- So many nice solvers freely available
- GUROBI, CVXGEN, CASADI, ACADO, QPOASIS, PDFMPC,. . .
- This totally frees creative formulations
- Formulations highly impact the results (stability/cpu-time)

Keywords involved in MPC Time shortage

Time shortage

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

Formulation

Prediction horizon
Constraints
Parametrization

Suboptimality

Uniform sampling
State-dependent

Architecture

Centralized
Hierarchical
Parallel

Hardware

PLC
Work Station
8-bit processor

Algorithm

IP
SQP
Gradient

Simplification

- Parametrization
- Observer
- Formulation

Distribution over time

- State-dependent sampling
- Co-design S/H

Distribution over space

- Fixed-point Hierarchical MPC
- GPU-based MPC

What if despite of the smartest parametrization, ...

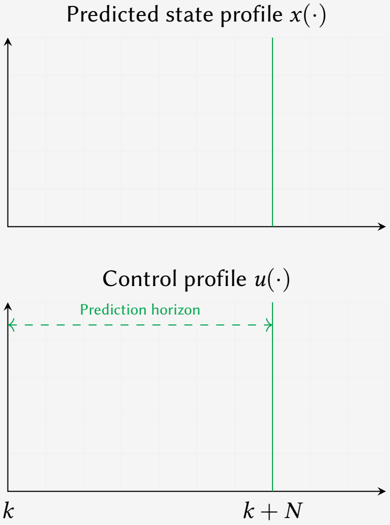
The problem is **not solvable** in the available time?

What if despite of the smartest parametrization, ...

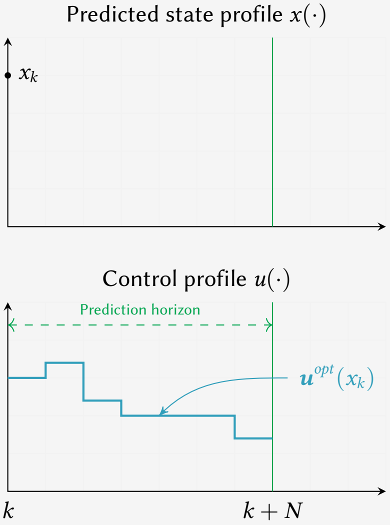
The problem is **not solvable** in the available time?

Distributing the computation over time ...!

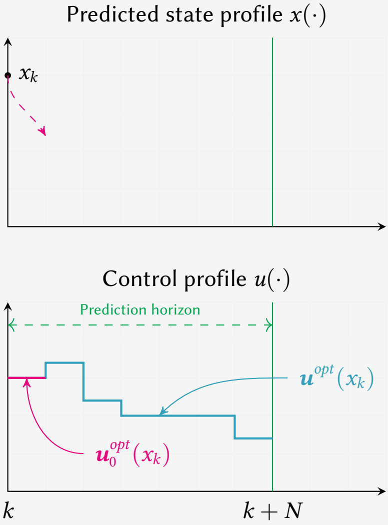
Distributing the computation: Intuitive illustration



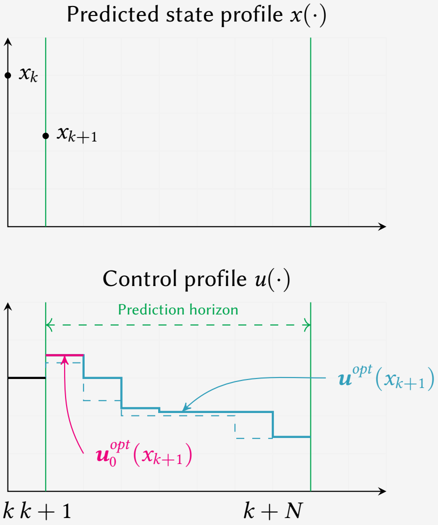
Distributing the computation: Intuitive illustration



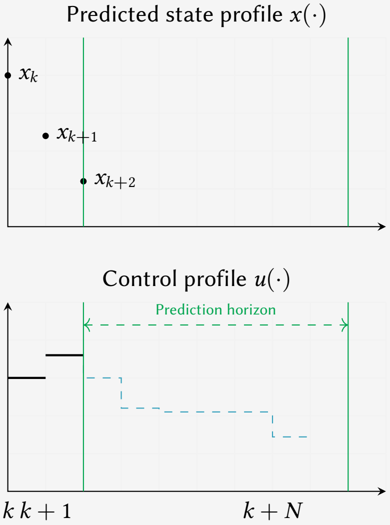
Distributing the computation: Intuitive illustration



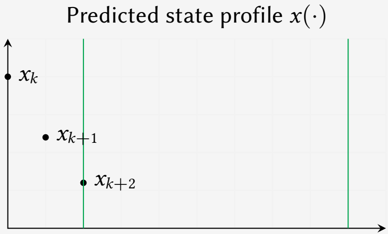
Distributing the computation: Intuitive illustration

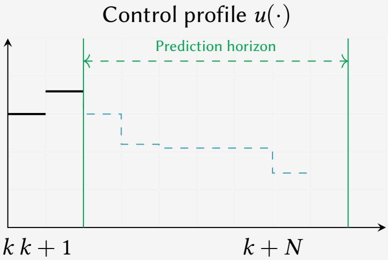


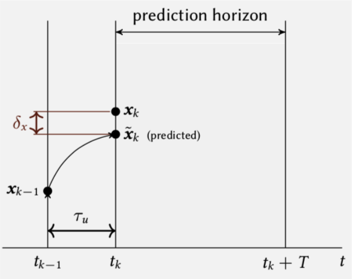
Distributing the computation: Intuitive illustration



Distributing the computation: Intuitive illustration

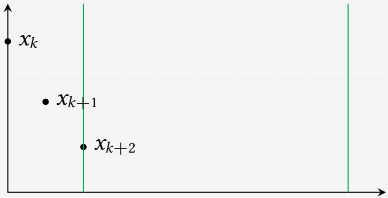




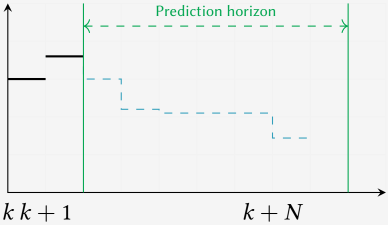


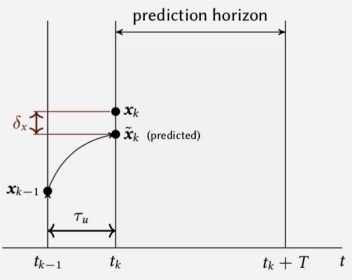
Distributing the computation: Intuitive illustration

Predicted state profile $x(\cdot)$



Control profile $u(\cdot)$

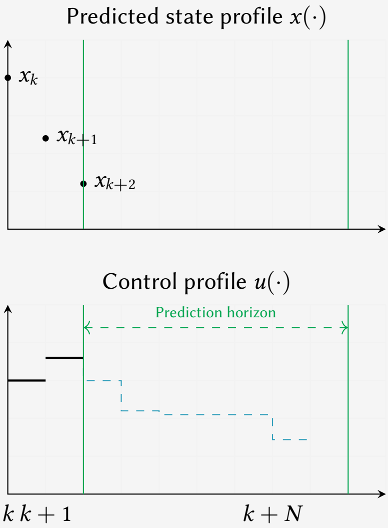


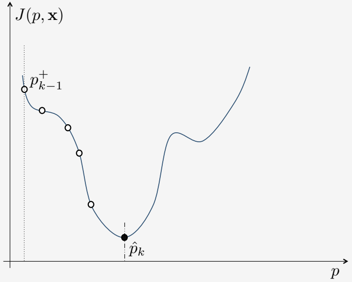


- if a single iteration costs τ_c time unites
- $\rightarrow$ Number of iterations during τ_u

$$q(\tau_u) := \left\lfloor \frac{\tau_u}{\tau_c} \right\rfloor$$

Distributing the computation: Intuitive illustration

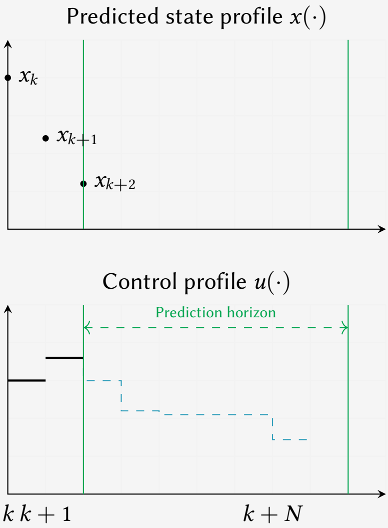


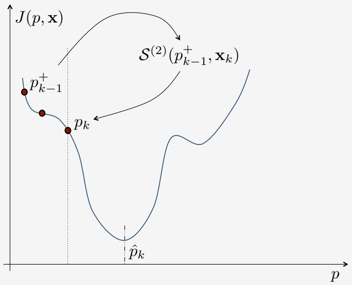


- if a single iteration costs τ_c time unites
- $\to$ Number of iterations during τ_u

$$q(\tau_u) := \left\lfloor \frac{\tau_u}{\tau_c} \right\rfloor$$

Distributing the computation: Intuitive illustration

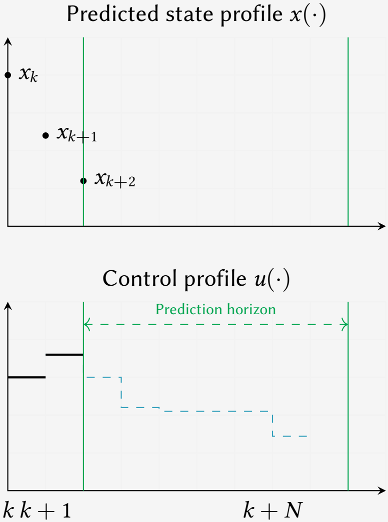


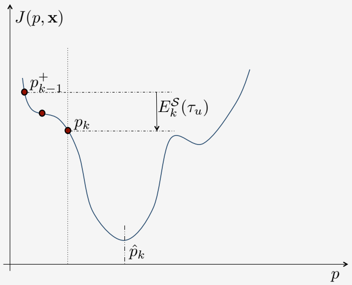


- if a single iteration costs τ_c time unites
- $\rightarrow$ Number of iterations during τ_u

$$q(\tau_u) := \left\lfloor \frac{\tau_u}{\tau_c} \right\rfloor$$

Distributing the computation: Intuitive illustration



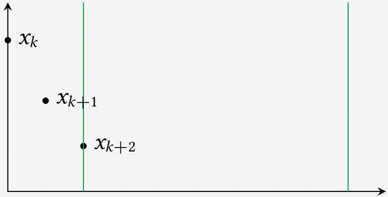


- if a single iteration costs τ_c time unites
- $\rightarrow$ Number of iterations during τ_u

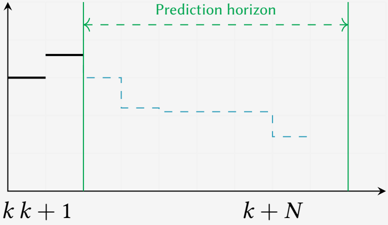
$$q(\tau_u) := \left\lfloor \frac{\tau_u}{\tau_c} \right\rfloor$$

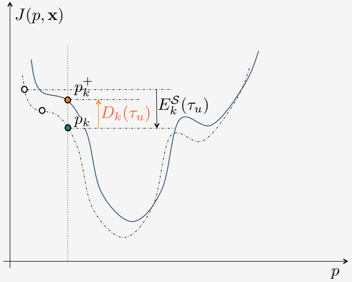
Distributing the computation: Intuitive illustration

Predicted state profile $x(\cdot)$



Control profile $u(\cdot)$

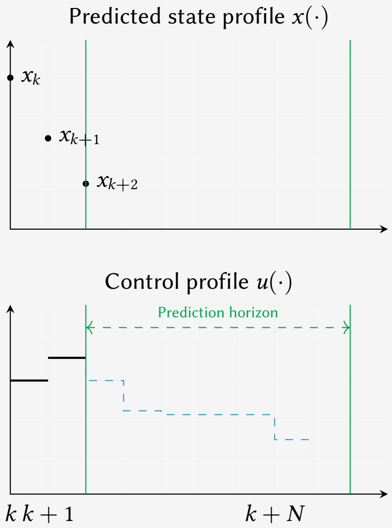


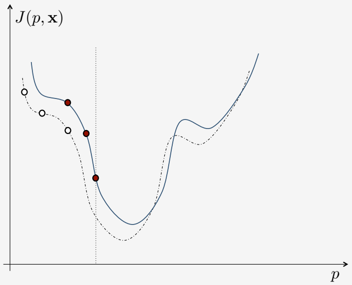


- if a single iteration costs τ_c time unites
- $\rightarrow$ Number of iterations during τ_u

$$q(\tau_u) := \left\lfloor \frac{\tau_u}{\tau_c} \right\rfloor$$

Distributing the computation: Intuitive illustration



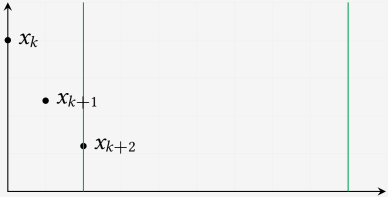


- if a single iteration costs τ_c time unites
- $\rightarrow$ Number of iterations during τ_u

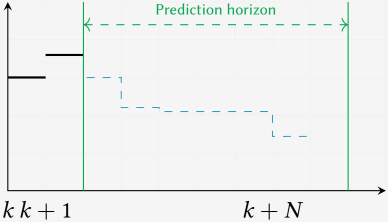
$$q(\tau_u) := \left\lfloor \frac{\tau_u}{\tau_c} \right\rfloor$$

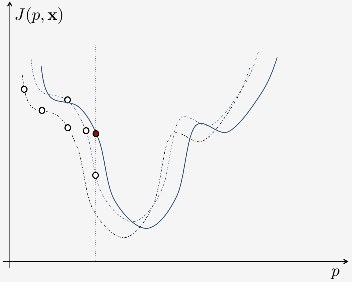
Distributing the computation: Intuitive illustration

Predicted state profile $x(\cdot)$



Control profile $u(\cdot)$



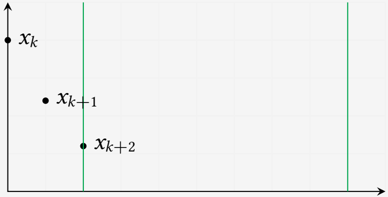


- if a single iteration costs τ_c time unites
- $\rightarrow$ Number of iterations during τ_u

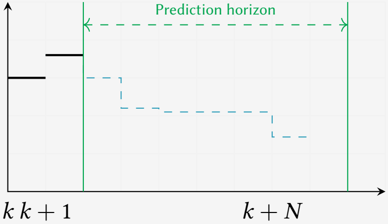
$$q(\tau_u) := \left\lfloor \frac{\tau_u}{\tau_c} \right\rfloor$$

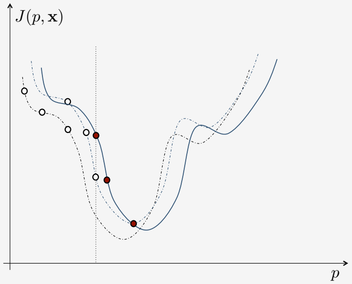
Distributing the computation: Intuitive illustration

Predicted state profile $x(\cdot)$



Control profile $u(\cdot)$



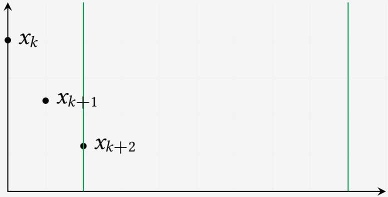


- if a single iteration costs τ_c time unites
- $\rightarrow$ Number of iterations during τ_u

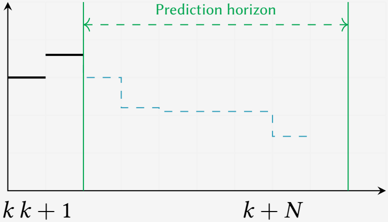
$$q(\tau_u) := \left\lfloor \frac{\tau_u}{\tau_c} \right\rfloor$$

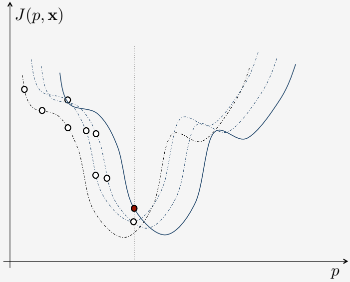
Distributing the computation: Intuitive illustration

Predicted state profile $x(\cdot)$



Control profile $u(\cdot)$



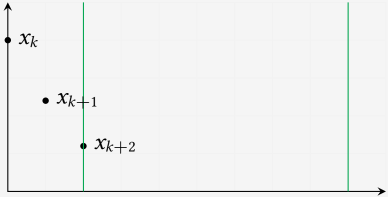


- if a single iteration costs τ_c time unites
- $\rightarrow$ Number of iterations during τ_u

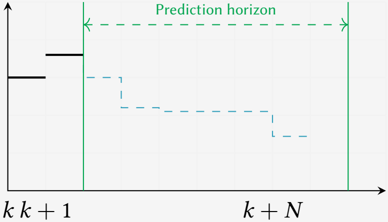
$$q(\tau_u) := \left\lfloor \frac{\tau_u}{\tau_c} \right\rfloor$$

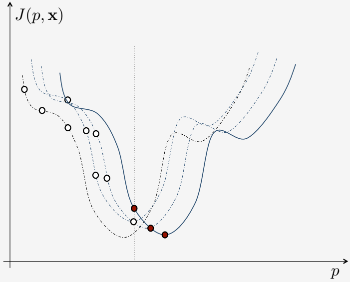
Distributing the computation: Intuitive illustration

Predicted state profile $x(\cdot)$



Control profile $u(\cdot)$



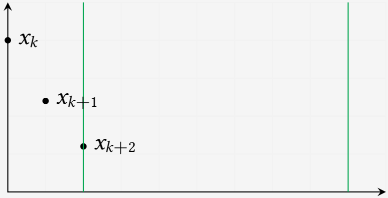


- if a single iteration costs τ_c time unites
- $\rightarrow$ Number of iterations during τ_u

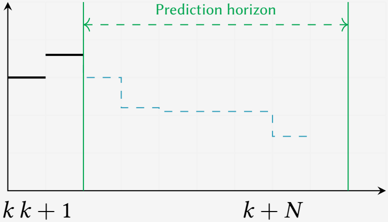
$$q(\tau_u) := \left\lfloor \frac{\tau_u}{\tau_c} \right\rfloor$$

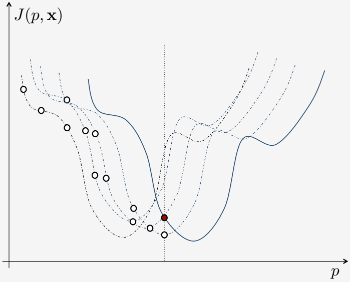
Distributing the computation: Intuitive illustration

Predicted state profile $x(\cdot)$



Control profile $u(\cdot)$



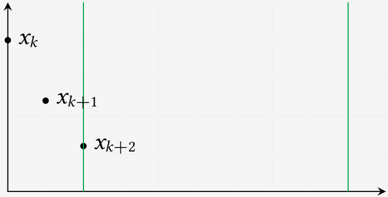


- if a single iteration costs τ_c time unites
- $\rightarrow$ Number of iterations during τ_u

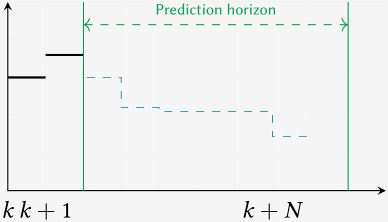
$$q(\tau_u) := \left\lfloor \frac{\tau_u}{\tau_c} \right\rfloor$$

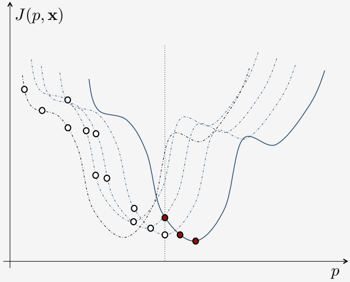
Distributing the computation: Intuitive illustration

Predicted state profile $x(\cdot)$



Control profile $u(\cdot)$



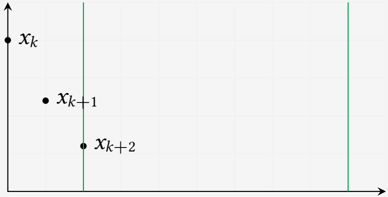


- if a single iteration costs τ_c time unites
- $\rightarrow$ Number of iterations during τ_u

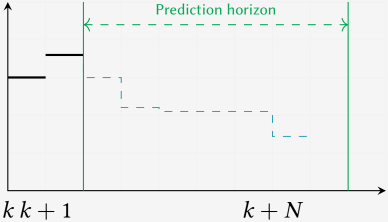
$$q(\tau_u) := \left\lfloor \frac{\tau_u}{\tau_c} \right\rfloor$$

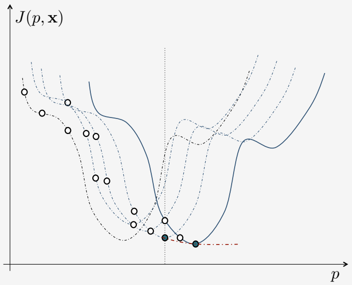
Distributing the computation: Intuitive illustration

Predicted state profile $x(\cdot)$



Control profile $u(\cdot)$



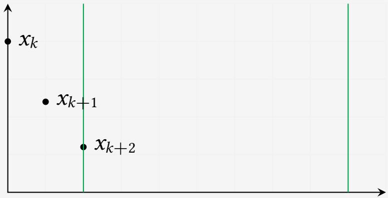


- if a single iteration costs τ_c time unites
- $\rightarrow$ Number of iterations during τ_u

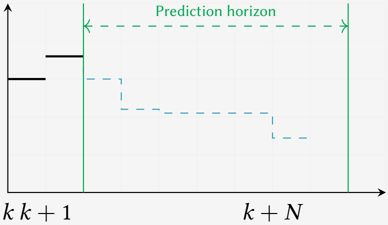
$$q(\tau_u) := \left\lfloor \frac{\tau_u}{\tau_c} \right\rfloor$$

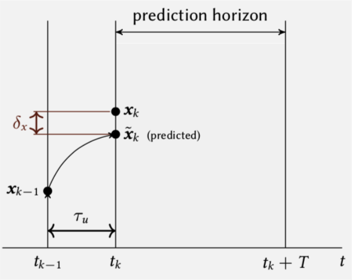
Distributing the computation: Intuitive illustration

Predicted state profile $x(\cdot)$



Control profile $u(\cdot)$





- if a single iteration costs τ_c time unites
- $\rightarrow$ Number of iterations during τ_u

$$q(\tau_u) := \left\lfloor \frac{\tau_u}{\tau_c} \right\rfloor$$

Use variable (adaptive) τ_u !

Updating τ_u is a control problem ...!

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

Updating τ_u is a control problem ...!

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

Updating τ_u is a control problem ...!

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

Updating τ_u is a control problem ...!

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

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

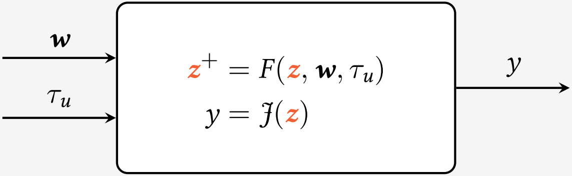
Updating τ_u is a control problem ...!

$$z^+ = F(z, \tau_u)$$
$$y = \mathcal{J}(z)$$

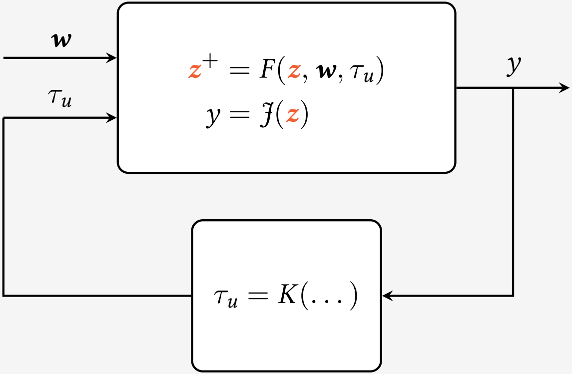
Updating τ_u is a control problem ...!

$$z^+ = F(z, \boldsymbol{w}, \tau_u)$$
$$y = \mathcal{J}(z)$$

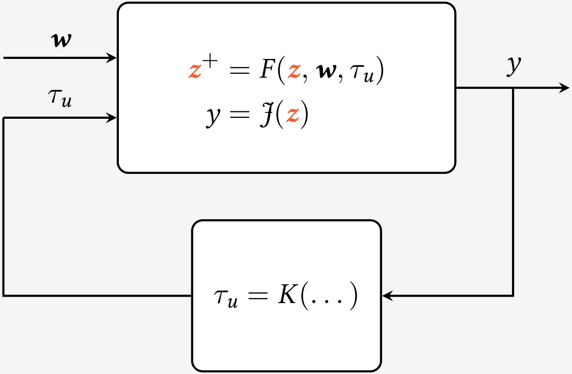
Updating τ_u is a control problem ...!



Updating τ_u is a control problem ...!



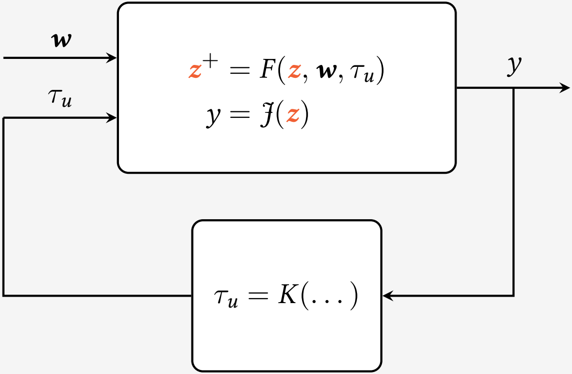
Updating τ_u is a control problem ...!



Complexity: ($5\pm$, $5\times$, $5\div$ and 1 log)

MA, Monitoring Control Updating Period In Fast Gradient-Based NMPC. ECC, Zurich, 2013
MA. A State-Dependent Updating Period For Certified Real-Time Model Predictive Control. IEEE TAC Vol. 62, Issue 5, 2017.
Bonne et al. Experimental investigation of control updating period monitoring in industrial PLC-based fast MPC: Application to the constrained control of a cryogenic refrigerator. Journal of Control Theory and technology. Vol. 15, Number 2, pages 92-108, 2017.

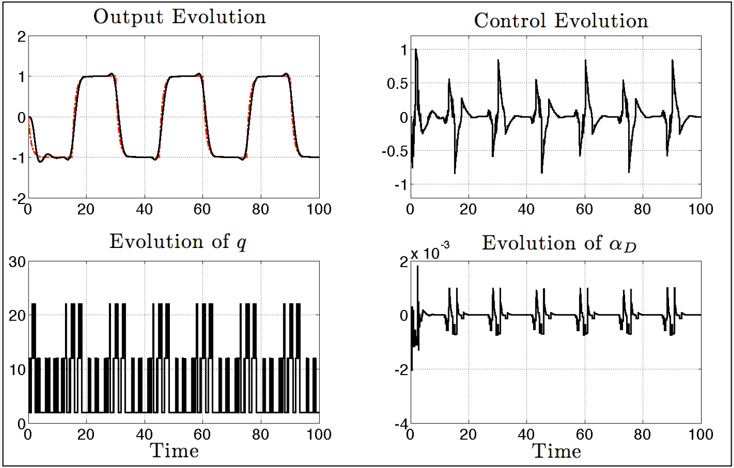
Updating τ_u is a control problem …!



Complexity: ($5\pm$, $5\times$, $5\div$ and 1 log)

MA, Monitoring Control Updating Period In Fast Gradient-Based NMPC. ECC, Zurich, 2013
MA. A State-Dependent Updating Period For Certified Real-Time Model Predictive Control. IEEE TAC Vol. 62, Issue 5, 2017.
Bonne et al. Experimental investigation of control updating period monitoring in industrial PLC-based fast MPC: Application to the constrained control of a cryogenic refrigerator. Journal of Control Theory and technology. Vol. 15, Number 2, pages 92-108, 2017.

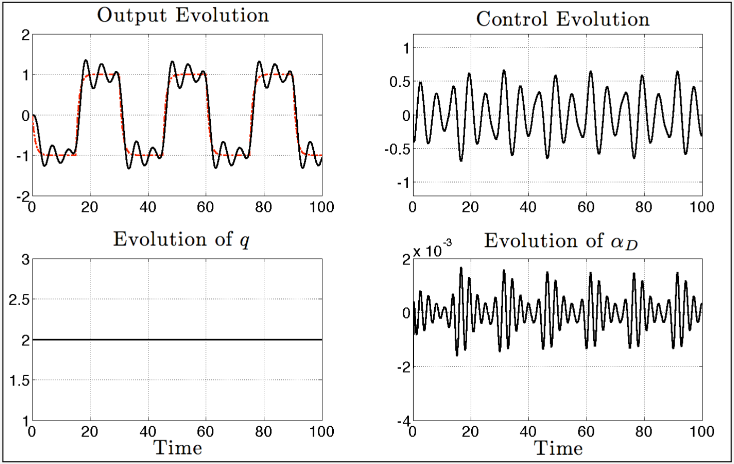
Updating τ_u is a control problem ...!: Illustrating a heuristic on a triple integrator



...

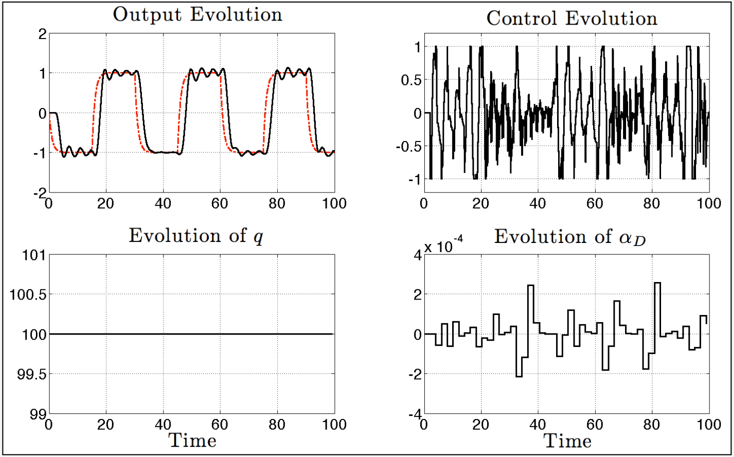
MA, Monitoring Control Updating Period In Fast Gradient-Based NMPC. ECC, Zurich, 2013

Updating τ_u is a control problem . . . !: Illustrating a heuristic on a triple integrator



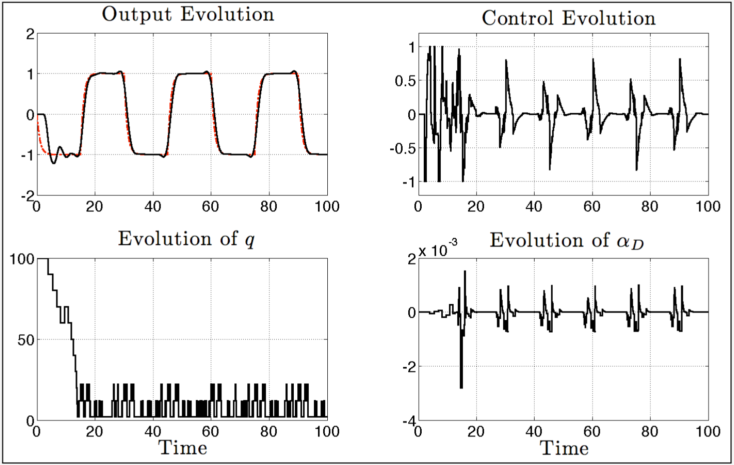
MA, Monitoring Control Updating Period In Fast Gradient-Based NMPC. ECC, Zurich, 2013

Updating τ_u is a control problem ...!: Illustrating a heuristic on a triple integrator



MA, Monitoring Control Updating Period In Fast Gradient-Based NMPC. ECC, Zurich, 2013

Updating τ_u is a control problem . . . !: Illustrating a heuristic on a triple integrator

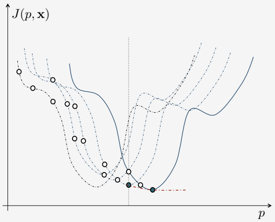


. . .

MA, Monitoring Control Updating Period In Fast Gradient-Based NMPC. ECC, Zurich, 2013

Case study 6

Updating rate as decision variable



Take away

- Consider adapting sampling period
- $\rightarrow$ Tune a state dependent sampling:
  - On-line heuristic
  - Machine learning
  - probabilistic certification

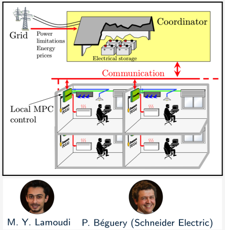
Case study 6

Smart Building

Smart building management: problem statement

(Illustration: courtesy of M. Y. Lamoudi)

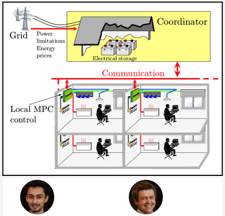
Lamoudi et al. A Distributed-in-time NMPC-Based Mechanism For Resource Sharing Problems. In Distributed MPC Made Easy. J. Maestre and R. R. Negenborn (Eds). Springer Verlag. (2012)



Smart building management: problem statement

(Illustration: courtesy of M. Y. Lamoudi)

Lamoudi et al. A Distributed-in-time NMPC-Based Mechanism For Resource Sharing Problems. In Distributed MPC Made Easy. J. Maestre and R. R. Negenborn (Eds). Springer Verlag. (2012)



M. Y. Lamoudi   P. Béguery (Schneider Electric)

Objective

1) Consumption monitoring

$$\boldsymbol{P}_b^+ + \sum_{\ell \in Z} \boldsymbol{P}_\ell \leq \bar{\boldsymbol{P}}_g$$

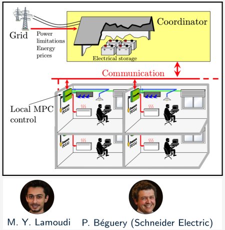
2) Optimize comfort:

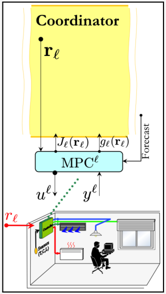
$$\mathcal{J} := \sum_{\ell \in Z} \mathcal{J}_\ell$$

Smart building management: problem statement

(Illustration: courtesy of M. Y. Lamoudi)

Lamoudi et al. A Distributed-in-time NMPC-Based Mechanism For Resource Sharing Problems. In Distributed MPC Made Easy. J. Maestre and R. R. Negenborn (Eds). Springer Verlag. (2012)





Objective

1) Consumption monitoring

$$\boldsymbol{P}_b^+ + \sum_{\ell \in Z} \boldsymbol{P}_\ell \le \bar{\boldsymbol{P}}_g$$

$J_\ell(\boldsymbol{r}_\ell)$ optimal cost | $\boldsymbol{r}_\ell$

$$g_\ell(\boldsymbol{r}_\ell) = \partial J_\ell / \partial r_\ell(\boldsymbol{r}_\ell)$$

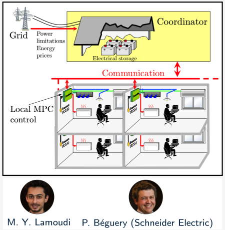
2) Optimize comfort:

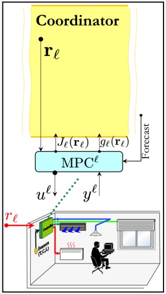
$$J := \sum_{\ell \in Z} J_\ell$$

Smart building management: problem statement

(Illustration: courtesy of M. Y. Lamoudi)

Lamoudi et al. A Distributed-in-time NMPC-Based Mechanism For Resource Sharing Problems. In Distributed MPC Made Easy. J. Maestre and R. R. Negenborn (Eds). Springer Verlag. (2012)





Objective

1) Consumption monitoring

$$\boldsymbol{P}_b^+ + \sum_{\ell \in Z} \boldsymbol{P}_\ell \le \bar{\boldsymbol{P}}_g$$

2) Optimize comfort:

$$\mathcal{J} := \sum_{\ell \in Z} \mathcal{J}_\ell$$

$\mathcal{J}_\ell(\boldsymbol{r}_\ell)$ optimal cost | $\boldsymbol{r}_\ell$

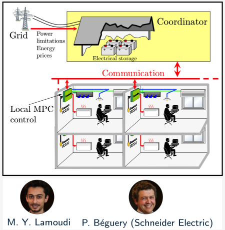
$$g_\ell(\boldsymbol{r}_\ell) = \partial \mathcal{J}_\ell / \partial r_\ell(\boldsymbol{r}_\ell)$$

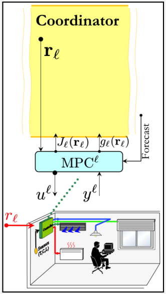
The coordinator **totally ignores** the expressions of local costs $\mathcal{J}_\ell$, the local models and the local states dimensions and values !

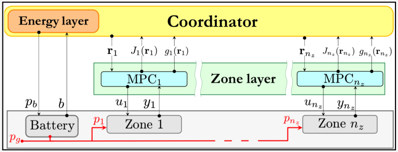
Smart building management: problem statement

(Illustration: courtesy of M. Y. Lamoudi)

Lamoudi et al. A Distributed-in-time NMPC-Based Mechanism For Resource Sharing Problems. In Distributed MPC Made Easy. J. Maestre and R. R. Negenborn (Eds). Springer Verlag. (2012)







Objective

1) Consumption monitoring

$$P_b^+ + \sum_{\ell \in Z} P_\ell \leq \bar{P}_g$$

2) Optimize comfort:

$$\mathcal{J} := \sum_{\ell \in Z} \mathcal{J}_\ell$$

$\mathcal{J}_\ell(r_\ell)$ optimal cost | r_ℓ

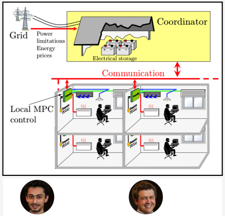
$$g_\ell(r_\ell) = \partial \mathcal{J}_\ell / \partial r_\ell (r_\ell)$$

The coordinator **totally ignores** the expressions of local costs $\mathcal{J}_\ell$, the local models and the local states dimensions and values !

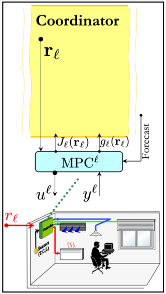
Smart building management: problem statement

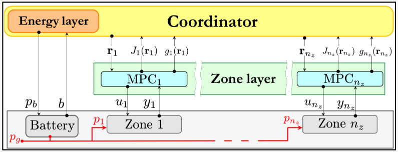
(Illustration: courtesy of M. Y. Lamoudi)

Lamoudi et al. A Distributed-in-time NMPC-Based Mechanism For Resource Sharing Problems. In Distributed MPC Made Easy. J. Maestre and R. R. Negenborn (Eds). Springer Verlag. (2012)



M. Y. Lamoudi P. Béguery (Schneider Electric)





Coordinator problem

How to allocate resources $\{\boldsymbol{r}_\ell\}_{\ell\in Z}$

Objective

1) Consumption monitoring

$$\boldsymbol{P}_b^+ + \sum_{\ell\in Z} \boldsymbol{P}_\ell \le \bar{\boldsymbol{P}}_g$$

2) Optimize comfort:

$$\mathcal{J} := \sum_{\ell\in Z} \mathcal{J}_\ell$$

$\mathcal{J}_\ell(\boldsymbol{r}_\ell)$ optimal cost | $\boldsymbol{r}_\ell$

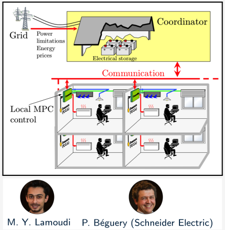
$$g_\ell(\boldsymbol{r}_\ell) = \partial\mathcal{J}_\ell/\partial r_\ell(\boldsymbol{r}_\ell)$$

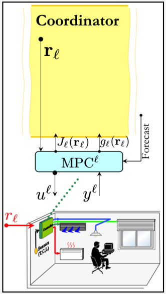
The coordinator **totally ignores** the expressions of local costs $\mathcal{J}_\ell$, the local models and the local states dimensions and values !

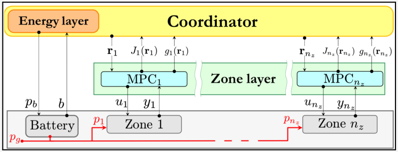
Smart building management: problem statement

(Illustration: courtesy of M. Y. Lamoudi)

Lamoudi et al. A Distributed-in-time NMPC-Based Mechanism For Resource Sharing Problems. In Distributed MPC Made Easy. J. Maestre and R. R. Negenborn (Eds). Springer Verlag. (2012)







Coordinator problem

How to allocate resources $\{r_\ell\}_{\ell \in Z}$

Objective

1) Consumption monitoring

$$P_b^+ + \sum_{\ell \in Z} P_\ell \leq \bar{P}_g$$

2) Optimize comfort:

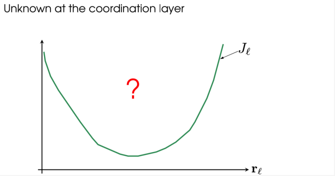
$$\mathcal{J} := \sum_{\ell \in Z} \mathcal{J}_\ell$$

$\mathcal{J}_\ell(r_\ell)$ optimal cost | r_ℓ

$$g_\ell(r_\ell) = \partial \mathcal{J}_\ell / \partial r_\ell(r_\ell)$$

The coordinator **totally ignores** the expressions of local costs $\mathcal{J}_\ell$, the local models and the local states dimensions and values !

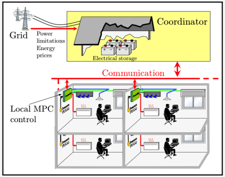
The Bundle Method



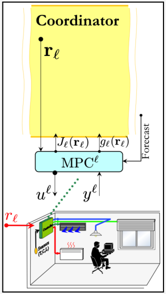
Smart building management: problem statement

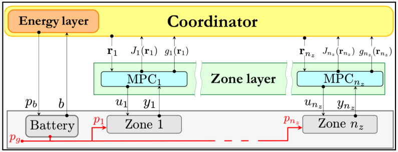
(Illustration: courtesy of M. Y. Lamoudi)

Lamoudi et al. A Distributed-in-time NMPC-Based Mechanism For Resource Sharing Problems. In Distributed MPC Made Easy. J. Maestre and R. R. Negenborn (Eds). Springer Verlag. (2012)



M. Y. Lamoudi P. Béguery (Schneider Electric)





Coordinator problem

How to allocate resources $\{r_\ell\}_{\ell \in Z}$

Objective

1) Consumption monitoring

$$P_b^+ + \sum_{\ell \in Z} P_\ell \leq \bar{P}_g$$

2) Optimize comfort:

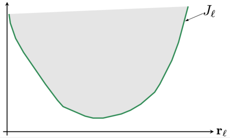
$$\mathcal{J} := \sum_{\ell \in Z} \mathcal{J}_\ell$$

$\mathcal{J}_\ell(r_\ell)$ optimal cost | r_ℓ

$$g_\ell(r_\ell) = \partial \mathcal{J}_\ell / \partial r_\ell(r_\ell)$$

The coordinator **totally ignores** the expressions of local costs $\mathcal{J}_\ell$, the local models and the local states dimensions and values !

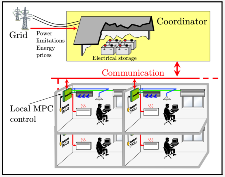
The Bundle Method



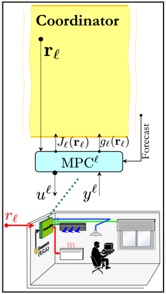
Smart building management: problem statement

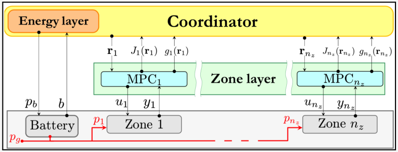
(Illustration: courtesy of M. Y. Lamoudi)

Lamoudi et al. A Distributed-in-time NMPC-Based Mechanism For Resource Sharing Problems. In Distributed MPC Made Easy. J. Maestre and R. R. Negenborn (Eds). Springer Verlag. (2012)



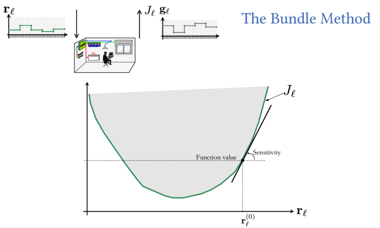
M. Y. Lamoudi P. Béguery (Schneider Electric)





Coordinator problem

How to allocate resources $\{r_\ell\}_{\ell\in Z}$



Objective

1) Consumption monitoring

$$P_b^+ + \sum_{\ell\in Z} P_\ell \leq \bar{P}_g$$

2) Optimize comfort:

$$\mathcal{J} := \sum_{\ell\in Z} \mathcal{J}_\ell$$

$\mathcal{J}_\ell(\boldsymbol{r}_\ell)$ optimal cost | $\boldsymbol{r}_\ell$

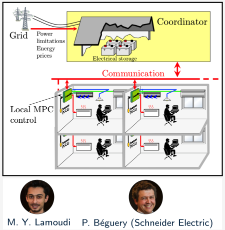
$$g_\ell(\boldsymbol{r}_\ell) = \partial\mathcal{J}_\ell/\partial r_\ell(\boldsymbol{r}_\ell)$$

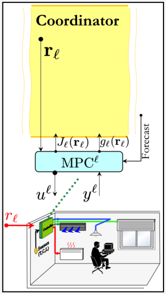
The coordinator **totally ignores** the expressions of local costs $\mathcal{J}_\ell$, the local models and the local states dimensions and values !

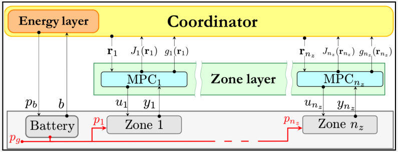
Smart building management: problem statement

(Illustration: courtesy of M. Y. Lamoudi)

Lamoudi et al. A Distributed-in-time NMPC-Based Mechanism For Resource Sharing Problems. In Distributed MPC Made Easy. J. Maestre and R. R. Negenborn (Eds). Springer Verlag. (2012)







Objective

1) Consumption monitoring

$$P_b^+ + \sum_{\ell \in Z} P_\ell \leq \bar{P}_g$$

2) Optimize comfort:

$$\mathcal{J} := \sum_{\ell \in Z} \mathcal{J}_\ell$$

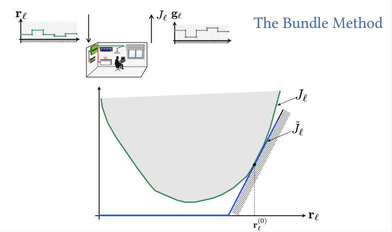
$\mathcal{J}_\ell(\boldsymbol{r}_\ell)$ optimal cost | $\boldsymbol{r}_\ell$

$$g_\ell(\boldsymbol{r}_\ell) = \partial \mathcal{J}_\ell / \partial r_\ell(\boldsymbol{r}_\ell)$$

The coordinator **totally ignores** the expressions of local costs $\mathcal{J}_\ell$, the local models and the local states dimensions and values !

Coordinator problem

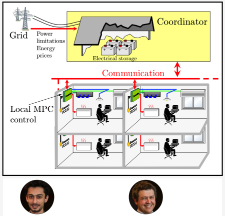
How to allocate resources $\{\boldsymbol{r}_\ell\}_{\ell \in Z}$



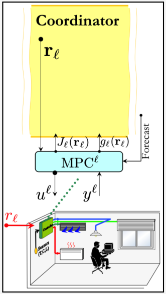
Smart building management: problem statement

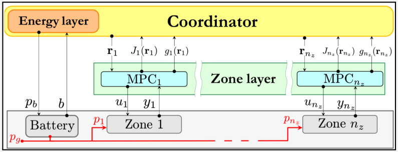
(Illustration: courtesy of M. Y. Lamoudi)

Lamoudi et al. A Distributed-in-time NMPC-Based Mechanism For Resource Sharing Problems. In Distributed MPC Made Easy. J. Maestre and R. R. Negenborn (Eds). Springer Verlag. (2012)



M. Y. Lamoudi   P. Béguery (Schneider Electric)





Objective

1) Consumption monitoring

$$P_b^+ + \sum_{\ell\in Z} P_\ell \le \bar{P}_g$$

2) Optimize comfort:

$$\mathcal{J} := \sum_{\ell\in Z} \mathcal{J}_\ell$$

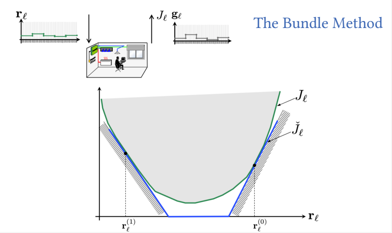
$\mathcal{J}_\ell(r_\ell)$ optimal cost | r_ℓ

$$g_\ell(r_\ell) = \partial\mathcal{J}_\ell/\partial r_\ell(r_\ell)$$

The coordinator **totally ignores** the expressions of local costs $\mathcal{J}_\ell$, the local models and the local states dimensions and values !

Coordinator problem

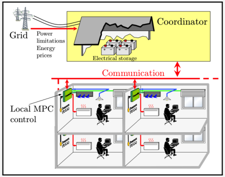
How to allocate resources $\{r_\ell\}_{\ell\in Z}$



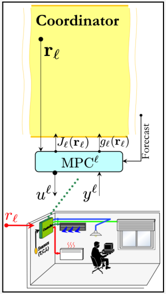
Smart building management: problem statement

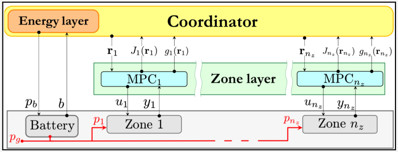
(Illustration: courtesy of M. Y. Lamoudi)

Lamoudi et al. A Distributed-in-time NMPC-Based Mechanism For Resource Sharing Problems. In Distributed MPC Made Easy. J. Maestre and R. R. Negenborn (Eds). Springer Verlag. (2012)



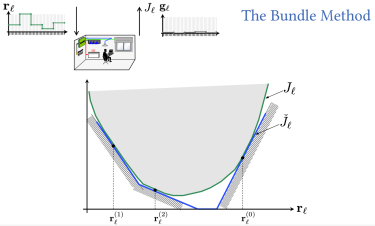
M. Y. Lamoudi   P. Béguery (Schneider Electric)





Coordinator problem

How to allocate resources $\{r_\ell\}_{\ell \in Z}$



Objective

1) Consumption monitoring

$$P_b^+ + \sum_{\ell \in Z} P_\ell \leq \bar{P}_g$$

2) Optimize comfort:

$$\mathcal{J} := \sum_{\ell \in Z} \mathcal{J}_\ell$$

$\mathcal{J}_\ell(r_\ell)$ optimal cost | r_ℓ

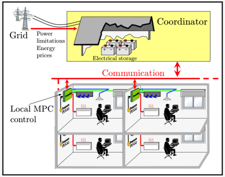
$$g_\ell(r_\ell) = \partial \mathcal{J}_\ell / \partial r_\ell(r_\ell)$$

The coordinator **totally ignores** the expressions of local costs $\mathcal{J}_\ell$, the local models and the local states dimensions and values !

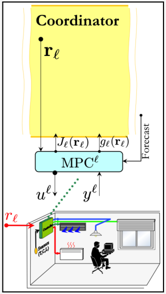
Smart building management: problem statement

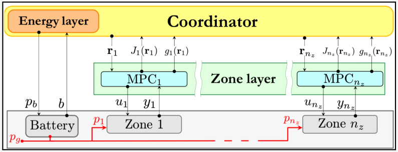
(Illustration: courtesy of M. Y. Lamoudi)

Lamoudi et al. A Distributed-in-time NMPC-Based Mechanism For Resource Sharing Problems. In Distributed MPC Made Easy. J. Maestre and R. R. Negenborn (Eds). Springer Verlag. (2012)



M. Y. Lamoudi   P. Béguery (Schneider Electric)





Objective

1) Consumption monitoring

$$P_b^+ + \sum_{\ell \in Z} P_\ell \leq \bar{P}_g$$

2) Optimize comfort:

$$\mathcal{J} := \sum_{\ell \in Z} \mathcal{J}_\ell$$

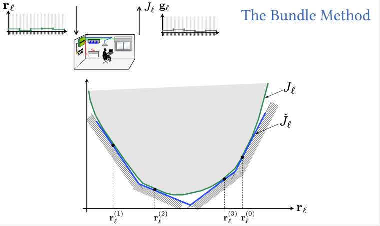
$\mathcal{J}_\ell(r_\ell)$ optimal cost | r_ℓ

$$g_\ell(r_\ell) = \partial \mathcal{J}_\ell / \partial r_\ell(r_\ell)$$

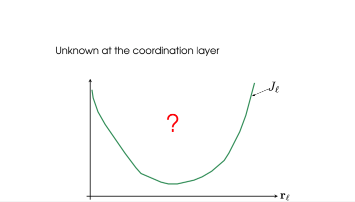
The coordinator **totally ignores** the expressions of local costs $\mathcal{J}_\ell$, the local models and the local states dimensions and values !

Coordinator problem

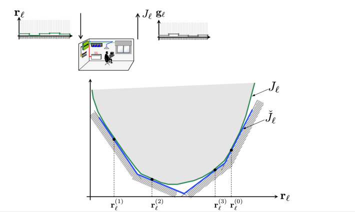
How to allocate resources $\{r_\ell\}_{\ell \in Z}$



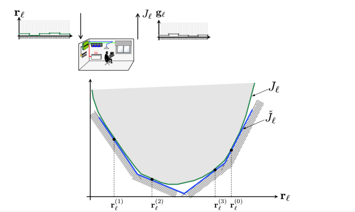
Smart building management: The Bundle method distributed in time



Smart building management: The Bundle method distributed in time

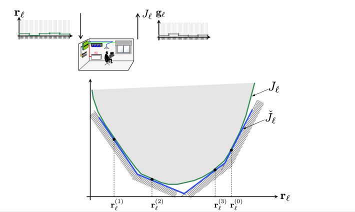


Smart building management: The Bundle method distributed in time



$$\check{\mathcal{J}}_\ell(r_\ell) := \max_{j\in\{1,\dots,s_{max}\}} \left[\mathcal{J}_\ell^{(j)} + \langle g_\ell^{(j)}, r_\ell - r_\ell^{(j)}\rangle\right]$$

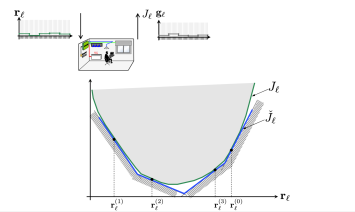
Smart building management: The Bundle method distributed in time



$$\check{J}_\ell(r_\ell) := \max_{j\in\{1,\dots,s_{max}\}} \left[J_\ell^{(j)} + \langle g_\ell^{(j)}, r_\ell - r_\ell^{(j)} \rangle \right]$$

- s_{max} is the # iterations coordinator/local controllers per updating period $[t_k, t_k + \tau_u]$.

Smart building management: The Bundle method distributed in time

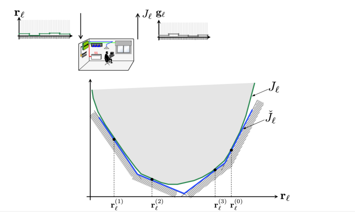


$$\check{\mathcal{J}}_\ell(r_\ell) := \max_{j\in\{1,\dots,s_{max}\}}\Big[\mathcal{J}_\ell^{(j)} + \langle g_\ell^{(j)}, r_\ell - r_\ell^{(j)}\rangle\Big]$$

- s_{max} is the # iterations coordinator/local controllers per updating period $[t_k, t_k+\tau_u]$.
- Model built using data contained in the Bundle:

$$\mathcal{B}_\ell^{(k)} := \Big\{r_\ell^{(j)}, \mathcal{J}_\ell^{(j)}, g_\ell^{(j)}\Big\}$$

Smart building management: The Bundle method distributed in time



$$\check{J}_\ell(r_\ell) := \max_{j\in\{1,\dots,s_{max}\}} \Big[J_\ell^{(j)} + \langle g_\ell^{(j)}, r_\ell - r_\ell^{(j)} \rangle \Big]$$

- s_{max} is the # iterations coordinator/local controllers per updating period $[t_k, t_k + \tau_u]$.
- Model built using data contained in the Bundle:

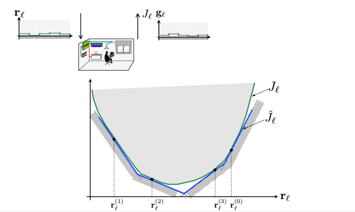
$$\mathcal{B}_\ell^{(k)} := \Big\{ r_\ell^{(j)}, J_\ell^{(j)}, g_\ell^{(j)} \Big\}$$

Use past bundles:

$$\mathcal{B}_\ell := \Big\{ \mathcal{B}_\ell^{(k)}, \dots, B_\ell^{(k-n_{\mathcal{B}})} \Big\}$$

[with a forgetting tricks not discussed here]

Smart building management: The Bundle method distributed in time



$$\check{\mathcal{J}}_\ell(r_\ell) := \max_{j\in\{1,\dots,s_{max}\}} \left[\mathcal{J}_\ell^{(j)} + \langle g_\ell^{(j)}, r_\ell - r_\ell^{(j)}\rangle\right]$$

- s_{max} is the # iterations coordinator/local controllers per updating period $[t_k, t_k + \tau_u]$.
- Model built using data contained in the Bundle:

$$\mathcal{B}_\ell^{(k)} := \left\{r_\ell^{(j)}, \mathcal{J}_\ell^{(j)}, g_\ell^{(j)}\right\}$$

Use past bundles:

$$\mathcal{B}_\ell := \left\{\mathcal{B}_\ell^{(k)}, \dots, B_\ell^{(k-n_{\mathcal{B}})}\right\}$$

[with a forgetting tricks not discussed here]

Two meta-parameters:

- s_{max}

  $\nearrow s_{max} \Rightarrow \nearrow \tau_u \Rightarrow \nearrow$ quality

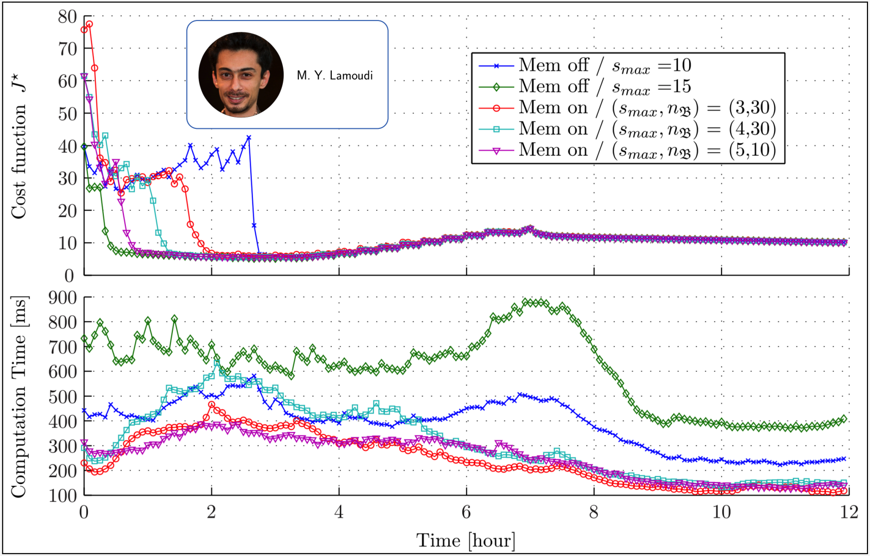
  but $\searrow$ reactivity.

- $n_{\mathcal{B}}$

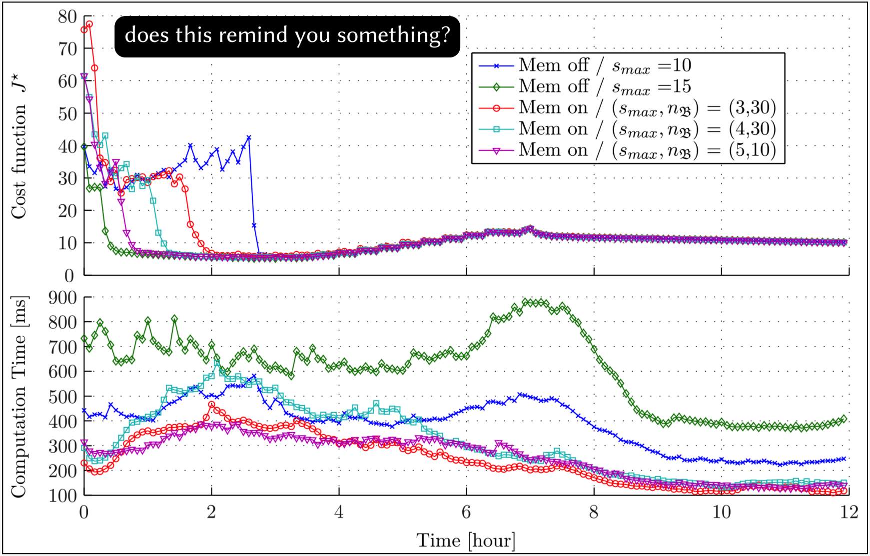
  $\nearrow n_{\mathcal{B}} \Rightarrow \nearrow$ quality

  but $\nearrow$ risk of out-of-date data.

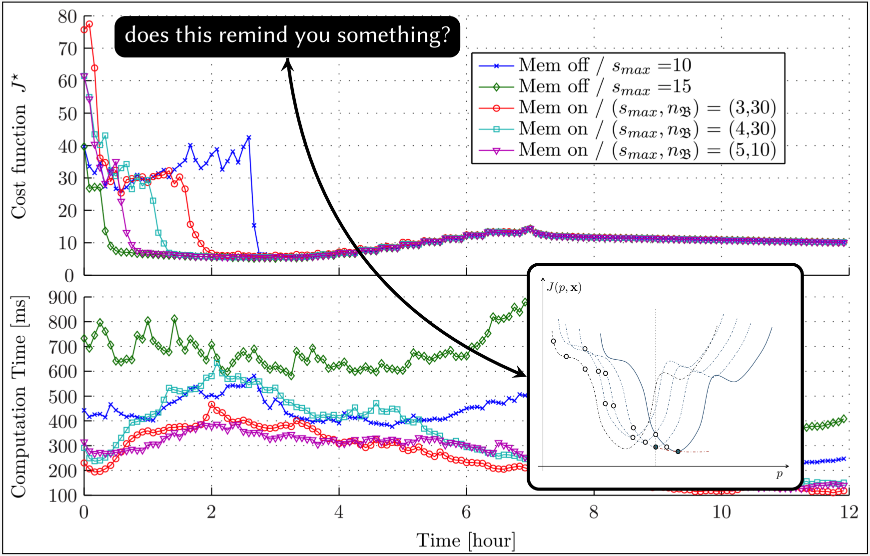
Smart building management: impact of meta-parameters s_{max} and $n_{\mathcal{B}}$ (Results: courtesy of M. Y. Lamoudi)



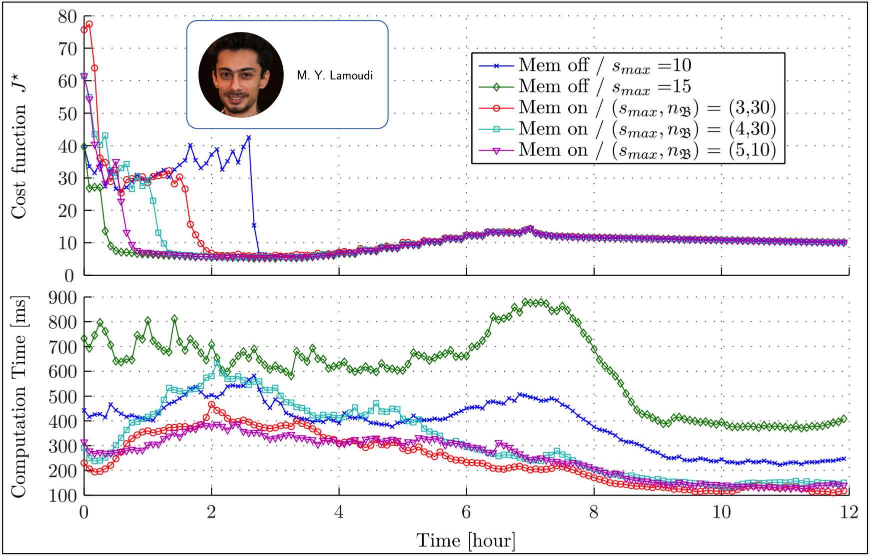
Smart building management: impact of meta-parameters s_{max} and $n_{\mathcal{B}}$ (Results: courtesy of M. Y. Lamoudi)



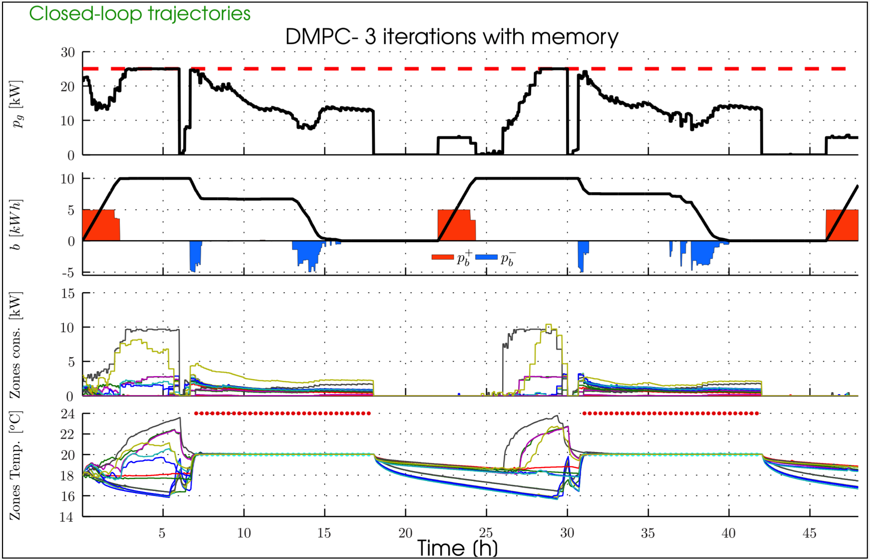
Smart building management: impact of meta-parameters s_{max} and $n_{\mathcal{B}}$ (Results: courtesy of M. Y. Lamoudi)



Smart building management: impact of meta-parameters s_{max} and $n_{\mathcal{B}}$ (Results: courtesy of M. Y. Lamoudi)



Smart building management: impact of meta-parameters s_{max} and $n_{\mathcal{B}}$ (Results: courtesy of M. Y. Lamoudi)



Case study 6

Smart Building

Take away

- Remember past computation
- ...with a forgetting factor

Scalability: Dual aggregated Bundle

(Results: courtesy of P. Pflaum)

- District composed of **1000** buildings of different size and inertia
- Prediction horizon $t_{horizon} = 24\text{h}$, sampling period $\tau = 15\text{min}$
- Number of decision variables $\approx$ **800000**
- Time to solve one building MPC problem: $t_{Subsys} = 75ms$
- $t_{Coordinator} = \mathbf{79}ms$ (aggregated bundle)
- Around **30** iterations required in closed loop
- Building MPC problems can be solved in parallel


Peter Pflaum

Total computation time to solve the problem:

$$t_{total} = n_{iter} \cdot (t_{Subsys} + t_{Coordinator}) = 30 \cdot (75[ms] + 79[ms]) = \mathbf{4.6[s]}$$

(Inter(R) Core(TM) i7-3540M CPU @ 3.00 GHz, 16,0 Go RAM, Gurobi 6.0 for LPs and QPs)

Pflaum et al. Scalability study for a hierarchical NMPC scheme for resource sharing problems. ECC2015, Linz, Austria, 2015.
Pflaum et al. Comparison of a primal and a dual decomposition for distributed MPC in smart districts. IEEE Symposium SmartGridComm, Venice, Italy, 2014.

Keywords involved in MPC Time shortage

Time shortage

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

Formulation

Prediction horizon
Constraints
Parametrization

Suboptimality

Uniform sampling
State-dependent

Architecture

Centralized
Hierarchical
Parallel

Hardware

PLC
Work Station
8-bit processor

Algorithm

IP
SQP
Gradient

Simplification

- Parametrization
- Observer
- Formulation

Distribution over time

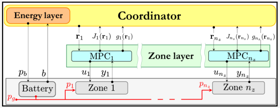
- State-dependent sampling
- Co-design S/H

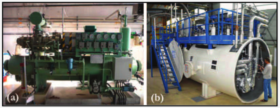
Distribution over space

- Fixed-point Hierarchical MPC
- GPU-based MPC

Distributing computation in space

Hierarchical MPC



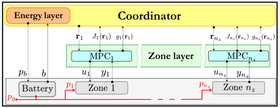


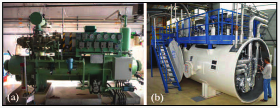
Lamoudi et al. A Distributed-in-time NMPC-Based Mechanism For Resource Sharing Problems. In Distributed MPC Made Easy. J. Maestre and R. R. Negenborn (Eds). **Springer Verlag**. (2012)

Alamir et al. Fixed-Point based hierarchical MPC control design for a cryogenic refrigerator. **Journal of Process Control**, Vol 58, pages 117-130, 2017.

Distributing computation in space

Hierarchical MPC





Lamoudi et al. A Distributed-in-time NMPC-Based Mechanism For Resource Sharing Problems. In Distributed MPC Made Easy. J. Maestre and R. R. Negenborn (Eds). **Springer Verlag**. (2012)

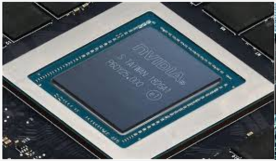
Alamir et al. Fixed-Point based hierarchical MPC control design for a cryogenic refrigerator. **Journal of Process Control**, Vol 58, pages 117-130, 2017.

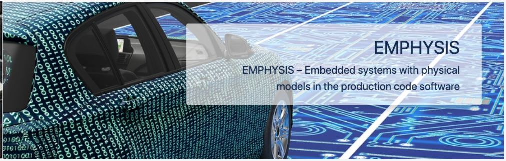
GPU-Based NMPC

Rathai et al., GPU-based parametrized NMPC scheme for control of half car vehicle with semi-active suspension system. **IEEE Control Systems Letters**, Vol. 3, Number 3, pages 631-636, 2019

Case study 7

GPU-based MPC of a semi-active suspension





Control of a semi-active suspension: problem statement


K. M. M. Rathai


O. Sename

Control objective

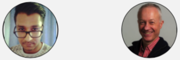
- Comfort ($\downarrow |\ddot{z}_s|$)
- Ride handling ($\downarrow |\theta^2|$)
- Meet box constraints on x/u

Equation of a half car SA-suspension

$$
\begin{aligned}
m_s\ddot{z}_s &= -F_{s,\ell} - F_{s,r} \\
I_x\ddot{\theta} &= \ell_\ell F_{s,\ell} - \ell_r F_{s,r} \\
m_{us,\ell}\ddot{z}_{us,\ell} &= F_{t,\ell} - F_{s,\ell} \\
m_{us,r}\ddot{z}_{us,r} &= F_{t,r} - F_{s,r}
\end{aligned}
$$

(ℓ) left, (r) right, (s) chassis, (t) wheel

Control of a semi-active suspension: problem statement



K. M. M. Rathai   O. Sename

Control objective

- Comfort ($\downarrow |\ddot{z}_s|$)
- Ride handling ($\downarrow |\theta^2|$)
- Meet box constraints on x/u

Equation of a half car SA-suspension

$$\begin{aligned} m_s\ddot{z}_s &= -F_{s,\ell} - F_{s,r} \\ I_x\ddot{\theta} &= \ell_\ell F_{s,\ell} - \ell_r F_{s,r} \\ m_{us,\ell}\ddot{z}_{us,\ell} &= F_{t,\ell} - F_{s,\ell} \\ m_{us,r}\ddot{z}_{us,r} &= F_{t,r} - F_{s,r} \end{aligned}$$

(ℓ) left, (r) right, (s) chassis, (t) wheel

$$F_{s,i} = -k_{s,i}(z_{s,i}(\theta) - z_{us,i}) + sa_i \quad i \in \{\ell, r\}$$
$$F_{t,i} = -k_{t,i}(z_{us,i} - z_{r,i}) \quad i \in \{\ell, r\}$$

Road profiles: disturbance

Control of a semi-active suspension: problem statement


K. M. M. Rathai


O. Sename

Control objective

- Comfort ($\downarrow |\ddot{z}_s|$)
- Ride handling ($\downarrow |\theta^2|$)
- Meet box constraints on x/u

Equation of a half car SA-suspension

$$
\begin{aligned}
m_s \ddot{z}_s &= -F_{s,\ell} - F_{s,r} \\
I_x \ddot{\theta} &= \ell_\ell F_{s,\ell} - \ell_r F_{s,r} \\
m_{us,\ell} \ddot{z}_{us,\ell} &= F_{t,\ell} - F_{s,\ell} \\
m_{us,r} \ddot{z}_{us,r} &= F_{t,r} - F_{s,r}
\end{aligned}
$$

$$
sa_i = k_0 z_{d,i} + c_0 \dot{z}_{d,i} + f_c \phi_i \tanh(a_1 \dot{z}_{d,i} + a_2 z_{d,i})
$$

(ℓ) left, (r) right, (s) chassis, (t) wheel

$$
\begin{aligned}
F_{s,i} &= -k_{s,i}(z_{s,i}(\theta) - z_{us,i}) + sa_i \quad i \in \{\ell, r\} \\
F_{t,i} &= -k_{t,i}(z_{us,i} - z_{r,i}) \quad i \in \{\ell, r\}
\end{aligned}
$$

Road profiles: disturbance

Control of a semi-active suspension: problem statement

K. M. M. Rathai

O. Sename

Control objective

- Comfort ($\downarrow |\ddot{z}_s|$)
- Ride handling ($\downarrow |\theta^2|$)
- Meet box constraints on x/u

Manipulated variables

- $u := (\phi_\ell, \phi_r) \in \mathbb{E}^2 \subset \mathbb{R}^2$
- $\mathbb{E}$ is a quantized (discrete) set.

Equation of a half car SA-suspension

$$\begin{aligned} m_s\ddot{z}_s &= -F_{s,\ell} - F_{s,r} \\ I_x\ddot{\theta} &= \ell_\ell F_{s,\ell} - \ell_r F_{s,r} \\ m_{us,\ell}\ddot{z}_{us,\ell} &= F_{t,\ell} - F_{s,\ell} \\ m_{us,r}\ddot{z}_{us,r} &= F_{t,r} - F_{s,r} \end{aligned}$$

$$sa_i = k_0 z_{d,i} + c_0 \dot{z}_{d,i} + f_c \phi_i \tanh(a_1 \dot{z}_{d,i} + a_2 z_{d,i})$$

(ℓ) left, (r) right, (s) chassis, (t) wheel

$$F_{s,i} = -k_{s,i}(z_{s,i}(\theta) - z_{us,i}) + sa_i \quad i \in \{\ell, r\}$$
$$F_{t,i} = -k_{t,i}(z_{us,i} - z_{r,i}) \quad i \in \{\ell, r\}$$

Road profiles: disturbance

Control of a semi-active suspension: problem statement


K. M. M. Rathai


O. Sename

Control objective

- Comfort ($\downarrow |\ddot{z}_s|$)
- Ride handling ($\downarrow |\theta^2|$)
- Meet box constraints on x/u

Manipulated variables

- $u := (\phi_\ell, \phi_r) \in \mathbb{E}^2 \subset \mathbb{R}^2$
- $\mathbb{E}$ is a quantized (discrete) set.

To summarize

- $\dot{x} = f(x, u, d)$
- $(x, u, d) \in \mathbb{R}^8 \times \mathbb{R}^2 \times \mathbb{R}^2$
- Sampling period $\leq$ 5ms

Equation of a half car SA-suspension

$$\begin{aligned} m_s\ddot{z}_s &= -F_{s,\ell} - F_{s,r} \\ I_x\ddot{\theta} &= \ell_\ell F_{s,\ell} - \ell_r F_{s,r} \\ m_{us,\ell}\ddot{z}_{us,\ell} &= F_{t,\ell} - F_{s,\ell} \\ m_{us,r}\ddot{z}_{us,r} &= F_{t,r} - F_{s,r} \end{aligned}$$

$$sa_i = k_0 z_{d,i} + c_0 \dot{z}_{d,i} + f_c \phi_i \tanh(a_1 \dot{z}_{d,i} + a_2 z_{d,i})$$

(ℓ) left, (r) right, (s) chassis, (t) wheel

$$F_{s,i} = -k_{s,i}(z_{s,i}(\theta) - z_{us,i}) + sa_i \quad i \in \{\ell, r\}$$
$$F_{t,i} = -k_{t,i}(z_{us,i} - z_{r,i}) \quad i \in \{\ell, r\}$$

Road profiles: disturbance

Control of SA-suspension: two competing designs

State of the art general NLP solver

- ACADO-qpOases / multiple shooting
- Symbolic automatic differentiation
- 4-th order Runge-Kutta integrator
- Code-generation based
- Standard p.w.c control profile
- projection of the quantized set
- Monitor the # iterations N_s

GPU/Simulation based solver

- Constant profile
- $\text{Card}(\mathbb{E})^2$ possibilities
- **Each** is simulated by a single thread
- **If** $\exists$ admissible solutions:

  Take the cost minimizer
- **otherwise**:

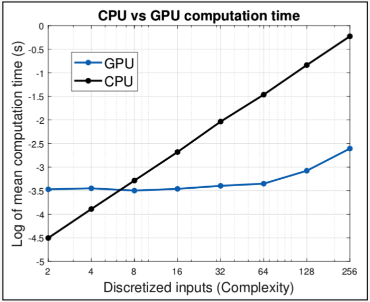
  Minimize the constraint violation

Sampling period = 5 ms
Prediction horizon length $N = 3$
$\text{Card}(\mathbb{E}) = 8^2 = 64$
MATLAB/SIMULINK, Intel Core i7 PC
GPU: NVIDIA GTX 1050 Ti with 768 CUDA cores

Rathai et al. GPU-Based parameterized NMPC scheme for control of half car vehicle with semi-active suspension system. **IEEE Control System Letters**, vol 3, No 3, 2019.

Results

Rathai et al. GPU-Based parameterized NMPC scheme for control of half car vehicle with semi-active suspension system. **IEEE Control System Letters**, vol 3, No 3, 2019.



(Courtesy K. M. M. Rathai)

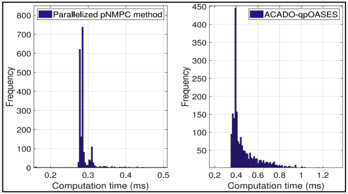
TABLE I
ACADO-QPOASES NMPC CONTROLLER

N_s	FA	Mean CT (ms)	Max CT (ms)	NCLO
5	✗	0.70	1.6	—
10	✗	0.95	2.2	—
15	✓	1.3	2.9	0.6484
20	✓	1.5	3.0	0.6412
25	✓	1.9	3.9	0.6317

TABLE II
PARALLELIZED PNMPC METHOD

$\{n_{\phi_l}, n_{\phi_r}\}$	FA	Mean CT (ms)	Max CT (ms)	NCLO
{2, 2}	✓	0.35	0.62	0.4679
{4, 4}	✓	0.35	0.60	0.4640
{8, 8}	✓	0.35	0.61	0.4646
{16, 16}	✓	0.36	0.55	0.4588
{32, 32}	✓	0.41	0.67	0.4568

(Courtesy K. M. M. Rathai)

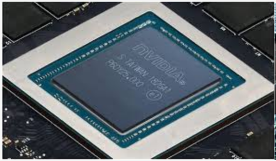


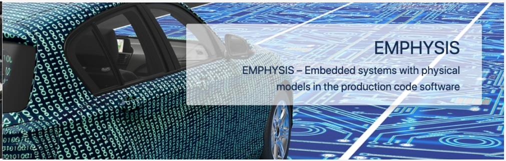
(Courtesy K. M. M. Rathai)

(FA) Feasibility
(CT) Computation Time
(NCLO) Normalized Closed-Loop Objective
(N_s) number of Newton iterations
(n_ϕ) card($\mathbb{E}$)

Case study 7

GPU-based MPC of a semi-active suspension





Take away

- Using GPU
- Consider simulation-based optimization
- Re-think/Re-invente efficient heuristics
- New culture to adopt...!

Conclusion

- Time shortage is inherent to competition-based market
- $\Rightarrow$ **it is and will be present endlessly**

Conclusion

- Time shortage is inherent to competition-based market
- ⇒ **it is and will be present endlessly**

$\begin{bmatrix} \text{Despite of} \\ \text{thanks to} \end{bmatrix}$ tremendous achievements

MPC is more alive than ever

waiting for your innovation ...!

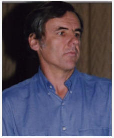
Acknowledgement

Peter Pflaum

M. Yacine Lamoudi

Rachid Amari

Guy Bornard



Claude Le pape

Paolo Tona

André Murilo

Gabriel Buche

François Bonne

Olivier Sename

K. M. M. Rathai

Patrick Béguery

Patrick Bonnay

Schneider Electric

CNRS

IFPEN

UGA

CEA

ANR

Gipsa-lab