Tango steps in a moving-horizon cabaret

Theory, concepts, heuristics and examples ...

Mazen Alamir

CNRS - University of Grenoble-Alpes

Outline

1. Non conventional NMPC Formulations
   - Economic MPC
   - Bringing the infinity closer
2. GPU-related topics
   - Some experiments
   - Use-case
3. Machine Learning-related topics
   - Stochastic NMPC by supervised clustering
   - ML-Model-based monitoring control updating period in NMPC
4. Discussion: Threats & suggestions for a data-dominated future

1 Non conventional NMPC Formulations

2 GPU-related topics

3 Machine Learning-related topics

4 Discussion: Threats & suggestions for a data-dominated future

Economic MPC

Without passivity, cost rotation and other aliens !

G. Pannochia

M.A and G. Pannochia, A new formulation of Economic . . . , Automatica Vol. 125 (2021)

Economic MPC

Without passivity, cost rotation and other aliens !

G. Pannochia

M.A and G. Pannochia, A new formulation of Economic . . . , Automatica Vol. 125 (2021)

Dynamics

$$x^+ = f(x, u)$$

Economic MPC

Without passivity, cost rotation and other aliens !

G. Pannochia

M.A and G. Pannochia, A new formulation of Economic . . . , Automatica Vol. 125 (2021)

Dynamics

$$x^+ = f(x,u)$$

$\|f(x,u) - x\|/\tau$

Stationary set

$$\mathcal{Z} := \Big\{(x,u) \in \mathbb{R}^n \times \mathbb{U} \mid \Delta(x,u) = 0\Big\}$$

Economic MPC

Without passivity, cost rotation and other aliens !

G. Pannochia

M.A and G. Pannochia, A new formulation of Economic . . . , Automatica Vol. 125 (2021)

Dynamics

$$x^+ = f(x,u)$$

$\|f(x,u) - x\|/\tau$

Stationary set

$$\mathcal{Z} := \Big\{(x,u) \in \mathbb{R}^n \times \mathbb{U} \mid \Delta(x,u) = 0\Big\}$$

Optimal Steady pair $z_s \in \mathcal{Z}$

$$\ell_s := \ell(z_s) \leq \ell(z) \quad \forall z \in \mathcal{Z}$$

Economic MPC

Without passivity, cost rotation and other aliens !

G. Pannochia

M.A and G. Pannochia, A new formulation of Economic . . . , Automatica Vol. 125 (2021)

Dynamics

$$x^+ = f(x,u)$$

$\|f(x,u) - x\|/\tau$

Stationary set

$$\mathcal{Z} := \Big\{(x,u) \in \mathbb{R}^n \times \mathbb{U} \mid \Delta(x,u) = 0\Big\}$$

Optimal Steady pair $z_s \in \mathcal{Z}$

$$\ell_s := \ell(z_s) \leq \ell(z) \quad \forall z \in \mathcal{Z}$$

Ideal EMPC

- z_s is **unknown**
- Cost function **should not** involve regulation-like terms ($\|z - z_s\|^2$)
- Rather it only involves ℓ-related terms
- $\rightarrow$ stability proofs involved *weird-to-MPC* tricks such as (strict passivity, rotated cost, etc)

Economic MPC

Without passivity, cost rotation and other aliens !

G. Pannochia

M.A and G. Pannochia, A new formulation of Economic . . . , Automatica Vol. 125 (2021)

Dynamics

$$x^+ = f(x,u) \qquad \|f(x,u)-x\|/\tau$$

Stationary set

$$\mathcal{Z} := \Big\{(x,u) \in \mathbb{R}^n \times \mathbb{U} \mid \Delta(x,u) = 0\Big\}$$

Optimal Steady pair $z_s \in \mathcal{Z}$

$$\ell_s := \ell(z_s) \leq \ell(z) \quad \forall z \in \mathcal{Z}$$

Ideal EMPC

- z_s is **unknown**
- Cost function **should not** involve regulation-like terms ($\|z - z_s\|^2$)
- Rather it only involves ℓ-related terms
- $\rightarrow$ stability proofs involved *weird-to-MPC* tricks such as (strict passivity, rotated cost, etc)

Question Is it possible to derive intuitive and **provably stable** formulations that do not involve such counter-intuitive tricks?

Economic MPC

Without passivity, cost rotation and other aliens !

G. Pannochia

M.A and G. Pannochia, A new formulation of Economic . . . , Automatica Vol. 125 (2021)

Dynamics

$$x^+ = f(x,u)$$

$\|f(x,u) - x\|/\tau$

Stationary set

$$\mathcal{Z} := \Big\{(x,u) \in \mathbb{R}^n \times \mathbb{U} \mid \Delta(x,u) = 0\Big\}$$

Optimal Steady pair $z_s \in \mathcal{Z}$

$$\ell_s := \ell(z_s) \leq \ell(z) \quad \forall z \in \mathcal{Z}$$

Ideal EMPC

- z_s is **unknown**
- Cost function **should not** involve regulation-like terms ($\|z - z_s\|^2$)
- Rather it only involves ℓ-related terms
- $\rightarrow$ stability proofs involved *weird-to-MPC* tricks such as (strict passivity, rotated cost, etc)

Question Is it possible to derive intuitive and **provably stable** formulations that do not involve such counter-intuitive tricks?

$$J(\boldsymbol{u},x) := \gamma\Big[\ell_N^{\boldsymbol{u}} + \alpha\Delta_N^{\boldsymbol{u}}\Big] + \underbrace{\sum_{k=0}^{N-1}\Big[\ell_k^{\boldsymbol{u}} + \alpha\Delta_k^{\boldsymbol{u}}\Big]}_{V(\boldsymbol{u},x)}$$

Cost function

Economic MPC

Without passivity, cost rotation and other aliens !

G. Pannochia

M.A and G. Pannochia, A new formulation of Economic . . . , Automatica Vol. 125 (2021)

Dynamics

$$x^+ = f(x,u)$$

$\|f(x,u) - x\|/\tau$

Stationary set

$$\mathcal{Z} := \Big\{(x,u) \in \mathbb{R}^n \times \mathbb{U} \mid \Delta(x,u) = 0\Big\}$$

Optimal Steady pair $z_s \in \mathcal{Z}$

$$\ell_s := \ell(z_s) \leq \ell(z) \quad \forall z \in \mathcal{Z}$$

Ideal EMPC

$\|z - z_s\|^2$

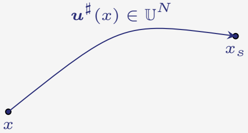
- z_s is **unknown**
- Cost function **should not** involve regulation-like terms
- Rather it only involves ℓ-related terms
- $\rightarrow$ stability proofs involved *weird-to-MPC* tricks such as (strict passivity, rotated cost, etc)

Question Is it possible to derive intuitive and **provably stable** formulations that do not involve such counter-intuitive tricks?

$$J(\boldsymbol{u},x) := \gamma\Big[\ell_N^{\boldsymbol{u}} + \alpha\Delta_N^{\boldsymbol{u}}\Big] + \underbrace{\sum_{k=0}^{N-1}\Big[\ell_k^{\boldsymbol{u}} + \alpha\Delta_k^{\boldsymbol{u}}\Big]}_{V(\boldsymbol{u},x)}$$

Cost function

- N-reachability of z_s $(\forall x_0 \in \mathbb{X}_0)$.
- $V(\boldsymbol{u}^\sharp(x),x) \leq K_s(\|x - x_s\|)$
- $\forall \boldsymbol{u}$, if η is sufficiently small $\rightarrow$ $\boldsymbol{u}^\dagger$ exists



Economic MPC

Without passivity, cost rotation and other aliens !

G. Pannochia

M.A and G. Pannochia, A new formulation of Economic . . . , Automatica Vol. 125 (2021)

Dynamics

$$x^+ = f(x,u)$$

$$\|f(x,u) - x\|/\tau$$

Stationary set

$$\mathcal{Z} := \Big\{(x,u) \in \mathbb{R}^n \times \mathbb{U} \mid \Delta(x,u) = 0\Big\}$$

Optimal Steady pair $z_s \in \mathcal{Z}$

$$\ell_s := \ell(z_s) \leq \ell(z) \quad \forall z \in \mathcal{Z}$$

Ideal EMPC

$$\|z - z_s\|^2$$

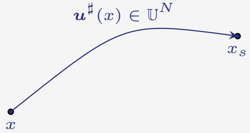
- z_s is **unknown**
- Cost function **should not** involve regulation-like terms
- Rather it only involves ℓ-related terms
- $\rightarrow$ stability proofs involved *weird-to-MPC* tricks such as (strict passivity, rotated cost, etc)

Question Is it possible to derive intuitive and **provably stable** formulations that do not involve such counter-intuitive tricks?

$$J(\boldsymbol{u},x) := \gamma\Big[\ell_N^{\boldsymbol{u}} + \alpha\Delta_N^{\boldsymbol{u}}\Big] + \underbrace{\sum_{k=0}^{N-1}\Big[\ell_k^{\boldsymbol{u}} + \alpha\Delta_k^{\boldsymbol{u}}\Big]}_{V(\boldsymbol{u},x)}$$

Cost function

- N-reachability of z_s $(\forall x_0 \in \mathbb{X}_0)$.
- $V(\boldsymbol{u}^\sharp(x), x) \leq K_s(\|x - x_s\|)$
- $\forall \boldsymbol{u}$, if η is sufficiently small $\rightarrow$ $\boldsymbol{u}^\dagger$ exists



Economic MPC

Without passivity, cost rotation and other aliens !

G. Pannochia

M.A and G. Pannochia, A new formulation of Economic . . . , Automatica Vol. 125 (2021)

Dynamics

$$x^+ = f(x,u)$$

$\|f(x,u) - x\|/\tau$

Stationary set

$$\mathcal{Z} := \Big\{(x,u) \in \mathbb{R}^n \times \mathbb{U} \mid \Delta(x,u) = 0\Big\}$$

Optimal Steady pair $z_s \in \mathcal{Z}$

$$\ell_s := \ell(z_s) \leq \ell(z) \quad \forall z \in \mathcal{Z}$$

Ideal EMPC

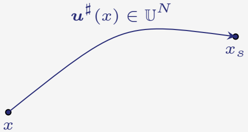
- z_s is **unknown**
- Cost function **should not** involve regulation-like terms ($\|z - z_s\|^2$)
- Rather it only involves ℓ-related terms
- $\rightarrow$ stability proofs involved *weird-to-MPC* tricks such as (strict passivity, rotated cost, etc)

Question Is it possible to derive intuitive and **provably stable** formulations that do not involve such counter-intuitive tricks?

$$J(\boldsymbol{u},x) := \gamma\Big[\ell_N^{\boldsymbol{u}} + \alpha\Delta_N^{\boldsymbol{u}}\Big] + \underbrace{\sum_{k=0}^{N-1}\Big[\ell_k^{\boldsymbol{u}} + \alpha\Delta_k^{\boldsymbol{u}}\Big]}_{V(\boldsymbol{u},x)}$$

Cost function

- N-reachability of z_s $(\forall x_0 \in \mathbb{X}_0)$.
- $V(\boldsymbol{u}^\sharp(x), x) \leq K_s(\|x - x_s\|)$
- $\forall \boldsymbol{u}$, if η is sufficiently small $\rightarrow$ $\boldsymbol{u}^\dagger$ exists



Economic MPC

Without passivity, cost rotation and other aliens !

G. Pannochia

M.A and G. Pannochia, A new formulation of Economic . . . , Automatica Vol. 125 (2021)

Dynamics

$$x^+ = f(x,u)$$

$\|f(x,u) - x\|/\tau$ ↑

Stationary set

$$\mathcal{Z} := \Big\{(x,u) \in \mathbb{R}^n \times \mathbb{U} \mid \Delta(x,u) = 0\Big\}$$

Optimal Steady pair $z_s \in \mathcal{Z}$

$$\ell_s := \ell(z_s) \le \ell(z) \quad \forall z \in \mathcal{Z}$$

Ideal EMPC

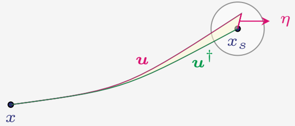
- z_s is **unknown**
- Cost function **should not** involve regulation-like terms ($\|z - z_s\|^2$)
- Rather it only involves ℓ-related terms
- → stability proofs involved *weird-to-MPC* tricks such as (strict passivity, rotated cost, etc)

Question: Is it possible to derive intuitive and **provably stable** formulations that do not involve such counter-intuitive tricks?

$$J(\boldsymbol{u},x) := \gamma\Big[\ell_N^{\boldsymbol{u}} + \alpha\Delta_N^{\boldsymbol{u}}\Big] + \underbrace{\sum_{k=0}^{N-1}\Big[\ell_k^{\boldsymbol{u}} + \alpha\Delta_k^{\boldsymbol{u}}\Big]}_{V(\boldsymbol{u},x)}$$

Cost function

- N-reachability of z_s ($\forall x_0 \in \mathbb{X}_0$).
- $V(\boldsymbol{u}^\sharp(x),x) \le K_s(\|x - x_s\|)$
- $\forall \boldsymbol{u}$, if η is sufficiently small → $\boldsymbol{u}^\dagger$ exists



Economic MPC

Without passivity, cost rotation and other aliens !

G. Pannochia

M.A and G. Pannochia, A new formulation of Economic . . . , Automatica Vol. 125 (2021)

Dynamics

$$x^+ = f(x,u) \qquad \|f(x,u) - x\|/\tau$$

Stationary set

$$\mathcal{Z} := \Big\{(x,u) \in \mathbb{R}^n \times \mathbb{U} \mid \Delta(x,u) = 0\Big\}$$

Optimal Steady pair $z_s \in \mathcal{Z}$

$$\ell_s := \ell(z_s) \leq \ell(z) \quad \forall z \in \mathcal{Z}$$

Ideal EMPC

$\|z - z_s\|^2$

- z_s is **unknown**
- Cost function **should not** involve regulation-like terms
- Rather it only involves ℓ-related terms
- $\rightarrow$ stability proofs involved *weird-to-MPC* tricks such as (strict passivity, rotated cost, etc)

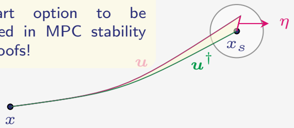
Question Is it possible to derive intuitive and **provably stable** formulations that do not involve such counter-intuitive tricks?

$$J(\boldsymbol{u},x) := \gamma\Big[\ell_N^{\boldsymbol{u}} + \alpha\Delta_N^{\boldsymbol{u}}\Big] + \underbrace{\sum_{k=0}^{N-1}\Big[\ell_k^{\boldsymbol{u}} + \alpha\Delta_k^{\boldsymbol{u}}\Big]}_{V(\boldsymbol{u},x)}$$

Cost function

- N-reachability of z_s $(\forall x_0 \in \mathbb{X}_0)$.
- $V(\boldsymbol{u}^\sharp(x), x) \leq K_s(\|x - x_s\|)$
- $\forall \boldsymbol{u}$, if η is sufficiently small $\rightarrow \boldsymbol{u}^\dagger$ exists

This is a new warm start option to be used in MPC stability proofs!



Economic MPC

Without passivity, cost rotation and other aliens !

G. Pannochia

M.A and G. Pannochia, A new formulation of Economic . . . , Automatica Vol. 125 (2021)

Dynamics

$$x^+ = f(x,u)$$

$\|f(x,u) - x\|/\tau$

Stationary set

$$\mathcal{Z} := \Big\{(x,u) \in \mathbb{R}^n \times \mathbb{U} \mid \Delta(x,u) = 0\Big\}$$

Optimal Steady pair $z_s \in \mathcal{Z}$

$$\ell_s := \ell(z_s) \leq \ell(z) \quad \forall z \in \mathcal{Z}$$

Ideal EMPC

- z_s is **unknown**
- Cost function **should not** involve regulation-like terms ($\|z - z_s\|^2$)
- Rather it only involves ℓ-related terms
- $\rightarrow$ stability proofs involved *weird-to-MPC* tricks such as (strict passivity, rotated cost, etc)

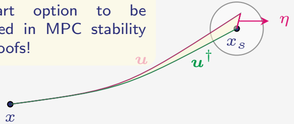
Question Is it possible to derive intuitive and **provably stable** formulations that do not involve such counter-intuitive tricks?

$$J(\boldsymbol{u},x) := \gamma\Big[\ell_N^{\boldsymbol{u}} + \alpha\Delta_N^{\boldsymbol{u}}\Big] + \underbrace{\sum_{k=0}^{N-1}\Big[\ell_k^{\boldsymbol{u}} + \alpha\Delta_k^{\boldsymbol{u}}\Big]}_{V(\boldsymbol{u},x)}$$

Cost function

- N-reachability of z_s ($\forall x_0 \in \mathbb{X}_0$).
- $V(\boldsymbol{u}^\sharp(x), x) \leq K_s(\|x - x_s\|)$
- $\forall \boldsymbol{u}$, if η is sufficiently small $\rightarrow \boldsymbol{u}^\dagger$ exists

This is a new warm start option to be used in MPC stability proofs!



Definition

A trajectory $(\boldsymbol{x},\boldsymbol{u})$ is said to be ϵ-quasi steady optimal if and only if the following conditions hold for all k:

$$|\ell(x_k,u_k) - \ell_s| \leq \epsilon \quad , \quad \Delta(x_k,u_k) \leq \epsilon$$

Economic MPC

Without passivity, cost rotation and other aliens !

G. Pannochia

M.A and G. Pannochia, A new formulation of Economic . . . , Automatica Vol. 125 (2021)

Dynamics

$$x^+ = f(x,u)$$

$\|f(x,u) - x\|/\tau$

Stationary set

$$\mathcal{Z} := \Big\{(x,u) \in \mathbb{R}^n \times \mathbb{U} \mid \Delta(x,u) = 0\Big\}$$

Optimal Steady pair $z_s \in \mathcal{Z}$

$$\ell_s := \ell(z_s) \leq \ell(z) \quad \forall z \in \mathcal{Z}$$

Proposition. For any desired $\epsilon > 0$, $\exists$ **sufficiently high** α, γ s.t the resulting closed-loop trajectory starting at $x_0 \in \mathbb{X}_0$ is asymptotically ϵ-quasi steady optimal.

Moreover, for dynamics that are obtained by time sampling, the size of the terminal region might be further reduced by reducing the sampling period τ.

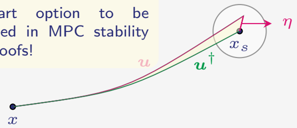
Question Is it possible to derive intuitive and **provably stable** formulations that do not involve such counter-intuitive tricks?

$$J(\boldsymbol{u}, x) := \gamma\Big[\ell_N^{\boldsymbol{u}} + \alpha\Delta_N^{\boldsymbol{u}}\Big] + \underbrace{\sum_{k=0}^{N-1}\Big[\ell_k^{\boldsymbol{u}} + \alpha\Delta_k^{\boldsymbol{u}}\Big]}_{V(\boldsymbol{u},x)}$$

Cost function

- N-reachability of z_s ($\forall x_0 \in \mathbb{X}_0$).
- $V(\boldsymbol{u}^\sharp(x), x) \leq K_s(\|x - x_s\|)$
- $\forall \boldsymbol{u}$, if η is sufficiently small $\rightarrow \boldsymbol{u}^\dagger$ exists

This is a new warm start option to be used in MPC stability proofs!



Definition

A trajectory $(\boldsymbol{x}, \boldsymbol{u})$ is said to be ϵ-quasi steady optimal if and only if the following conditions hold for all k:

$$|\ell(x_k, u_k) - \ell_s| \leq \epsilon \quad , \quad \Delta(x_k, u_k) \leq \epsilon$$

Bringing infinity closer!

MPC with exponentially increasing weight

Cost function

$$J(\boldsymbol{u}, x) := \sum_{k=1}^{N} [k/N]^m \Big[\varphi_m \ell_k^{\boldsymbol{u}} + (1-\varphi_m) V_k^{\boldsymbol{u}}\Big]$$

M.A Stability proof for nonlinear MPC design using monotonically increasing weighting ..., **Automatica 87**, 2018.
M.A Numerical investigation regarding ... Application to the control of real-life cryogenic plant, **NMPC workshop** 2018.

Bringing infinity closer!

MPC with exponentially increasing weight

Cost function

$$J(\boldsymbol{u}, x) := \sum_{k=1}^{N} [k/N]^m \Big[\varphi_m \ell_k^{\boldsymbol{u}} + (1-\varphi_m) V_k^{\boldsymbol{u}}\Big]$$

- ℓ is the stage cost of interest
- V is a **local** controlled Lyapunov function
- $\varphi(0) = 1$ and $\lim_{m\to\infty} \varphi_m = 0$
- **Assumption**: N-reachability over $\mathbb{X}_0$

M.A Stability proof for nonlinear MPC design using monotonically increasing weighting . . . , **Automatica 87**, 2018.
M.A Numerical investigation regarding . . . Application to the control of real-life cryogenic plant, **NMPC workshop** 2018.

Bringing infinity closer!

MPC with exponentially increasing weight

Cost function

$$J(\boldsymbol{u}, x) := \sum_{k=1}^{N} [k/N]^m \Big[\varphi_m \ell_k^{\boldsymbol{u}} + (1-\varphi_m) V_k^{\boldsymbol{u}}\Big]$$

- ℓ is the stage cost of interest
- V is a **local** controlled Lyapunov function
- $\varphi(0) = 1$ and $\lim_{m\to\infty} \varphi_m = 0$
- **Assumption**: N-reachability over $\mathbb{X}_0$

Proposition. For sufficiently high m $\rightarrow$ Closed-loop stability of for any initial state inside $\mathbb{X}_0$.

M.A Stability proof for nonlinear MPC design using monotonically increasing weighting . . . , **Automatica 87**, 2018.
M.A Numerical investigation regarding . . . Application to the control of real-life cryogenic plant, **NMPC workshop** 2018.

Bringing infinity closer!

MPC with exponentially increasing weight

Cost function

$$J(\boldsymbol{u}, x) := \sum_{k=1}^{N} [k/N]^m \Big[\varphi_m \ell_k^{\boldsymbol{u}} + (1-\varphi_m)V_k^{\boldsymbol{u}}\Big]$$

- ℓ is the stage cost of interest
- V is a **local** controlled Lyapunov function
- $\varphi(0) = 1$ and $\lim_{m\to\infty} \varphi_m = 0$
- **Assumption**: N-reachability over $\mathbb{X}_0$

Proposition. For sufficiently high m $\rightarrow$ Closed-loop stability of for any initial state inside $\mathbb{X}_0$.

Questions

Q1. Are we still minimizing the same cost?

Q2. Implication on the computation time!

M.A Stability proof for nonlinear MPC design using monotonically increasing weighting ..., **Automatica 87**, 2018.
M.A Numerical investigation regarding ... Application to the control of real-life cryogenic plant, **NMPC workshop** 2018.

Bringing infinity closer!

MPC with exponentially increasing weight

Cost function

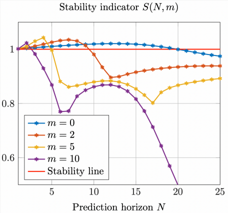
$$J(\boldsymbol{u}, x) := \sum_{k=1}^{N} [k/N]^m \Big[\varphi_m \ell_k^{\boldsymbol{u}} + (1-\varphi_m) V_k^{\boldsymbol{u}}\Big]$$

- ℓ is the stage cost of interest
- V is a **local** controlled Lyapunov function
- $\varphi(0) = 1$ and $\lim_{m\to\infty} \varphi_m = 0$
- **Assumption**: N-reachability over $\mathbb{X}_0$

Proposition. For sufficiently high m $\rightarrow$ Closed-loop stability of for any initial state inside $\mathbb{X}_0$.

Q1

$\dot{x}_1 = x_2, \dot{x}_2 = x_3, \dot{x}_3 = u$



Evolution of the highest module of closed-loop eigenvalues $S(N, m)$

Questions

Q1. Are we still minimizing the same cost?

Q2. Implication on the computation time!

M.A Stability proof for nonlinear MPC design using monotonically increasing weighting . . . , **Automatica 87**, 2018.
M.A Numerical investigation regarding . . . Application to the control of real-life cryogenic plant, **NMPC workshop** 2018.

Bringing infinity closer!

MPC with exponentially increasing weight

Cost function

$$J(\boldsymbol{u}, x) := \sum_{k=1}^{N} [k/N]^m \Big[\varphi_m \ell_k^{\boldsymbol{u}} + (1-\varphi_m) V_k^{\boldsymbol{u}}\Big]$$

- ℓ is the stage cost of interest
- V is a **local** controlled Lyapunov function
- $\varphi(0) = 1$ and $\lim_{m\to\infty} \varphi_m = 0$
- **Assumption**: N-reachability over $\mathbb{X}_0$

Proposition. For sufficiently high m → Closed-loop stability of for any initial state inside $\mathbb{X}_0$.

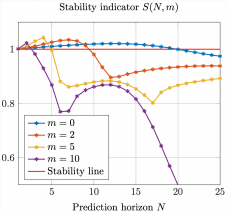
Questions

Q1. Are we still minimizing the same cost?

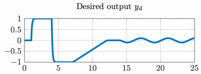
Q2. Implication on the computation time!

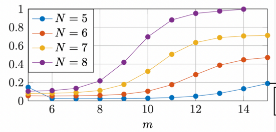
Q1

$\dot{x}_1 = x_2, \dot{x}_2 = x_3, \dot{x}_3 = u$



Evolution of the highest module of closed-loop eigenvalues $S(N,m)$





$$\frac{\Big[\sum_i \ell_i^{cl}\Big]^{\text{Proposed}}}{\Big[\sum_i \ell_i^{cl}\Big]^{\text{term. constr.}}}$$

M.A Stability proof for nonlinear MPC design using monotonically increasing weighting ..., **Automatica 87**, 2018.
M.A Numerical investigation regarding ... Application to the control of real-life cryogenic plant, **NMPC workshop** 2018.

Bringing infinity closer!

MPC with exponentially increasing weight

Cost function

$$J(\boldsymbol{u}, x) := \sum_{k=1}^{N} [k/N]^m \Big[\varphi_m \ell_k^{\boldsymbol{u}} + (1-\varphi_m) V_k^{\boldsymbol{u}}\Big]$$

- ℓ is the stage cost of interest
- V is a **local** controlled Lyapunov function
- $\varphi(0) = 1$ and $\lim_{m\to\infty} \varphi_m = 0$
- **Assumption**: N-reachability over $\mathbb{X}_0$

Proposition. For sufficiently high m → Closed-loop stability of for any initial state inside $\mathbb{X}_0$.

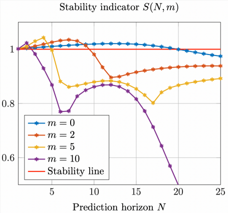
Questions

Q1. Are we still minimizing the same cost?

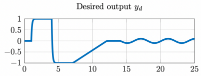
Q2. Implication on the computation time!

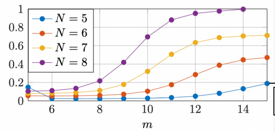
Q1

$\dot{x}_1 = x_2, \dot{x}_2 = x_3, \dot{x}_3 = u$



Evolution of the highest module of closed-loop eigenvalues $S(N, m)$





$$\frac{\left[\sum_i \ell_i^{cl}\right]^{\text{Proposed}}}{\left[\sum_i \ell_i^{cl}\right]^{\text{term. constr.}}}$$

The shorter N, the stronger is the advantage!

M.A Stability proof for nonlinear MPC design using monotonically increasing weighting . . . , **Automatica 87**, 2018.
M.A Numerical investigation regarding . . . Application to the control of real-life cryogenic plant, **NMPC workshop** 2018.

Bringing infinity closer!

MPC with exponentially increasing weight

Cost function

$$J(\boldsymbol{u}, x) := \sum_{k=1}^{N} [k/N]^m \Big[\varphi_m \ell_k^{\boldsymbol{u}} + (1-\varphi_m) V_k^{\boldsymbol{u}}\Big]$$

- ℓ is the stage cost of interest
- V is a **local** controlled Lyapunov function
- $\varphi(0) = 1$ and $\lim_{m\to\infty} \varphi_m = 0$
- **Assumption**: N-reachability over $\mathbb{X}_0$

Proposition. For sufficiently high m $\rightarrow$ Closed-loop stability of for any initial state inside $\mathbb{X}_0$.

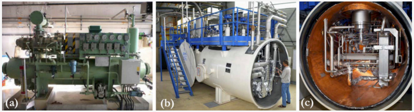
Questions

Q1. Are we still minimizing the same cost?

Q2. Implication on the computation time!

Q1-Q2

Cryogenic station of CEA-INAC-SBT



$$x^+ = Ax + Bu + Gw$$
$$y = Cx + Du \quad (x, u, y, w) \in \mathbb{R}^{24} \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}$$

M.A Stability proof for nonlinear MPC design using monotonically increasing weighting . . . , **Automatica 87**, 2018.
M.A Numerical investigation regarding . . . Application to the control of real-life cryogenic plant, **NMPC workshop** 2018.

Bringing infinity closer!

MPC with exponentially increasing weight

Cost function

$$J(\boldsymbol{u}, x) := \sum_{k=1}^{N} [k/N]^m \Big[\varphi_m \ell_k^{\boldsymbol{u}} + (1-\varphi_m) V_k^{\boldsymbol{u}}\Big]$$

- ℓ is the stage cost of interest
- V is a **local** controlled Lyapunov function
- $\varphi(0) = 1$ and $\lim_{m\to\infty} \varphi_m = 0$
- **Assumption**: N-reachability over $\mathbb{X}_0$

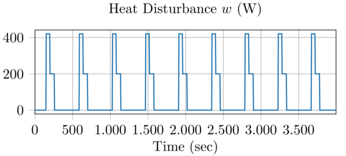
Proposition. For sufficiently high m $\rightarrow$ Closed-loop stability of for any initial state inside $\mathbb{X}_0$.

Questions

Q1. Are we still minimizing the same cost?

Q2. Implication on the computation time!

Q1-Q2



$$x^+ = Ax + Bu + Gw$$
$$y = Cx + Du \quad (x, u, y, w) \in \mathbb{R}^{24} \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}$$

M.A Stability proof for nonlinear MPC design using monotonically increasing weighting . . . , **Automatica 87**, 2018.
M.A Numerical investigation regarding . . . Application to the control of real-life cryogenic plant, **NMPC workshop** 2018.

Bringing infinity closer!

MPC with exponentially increasing weight

Cost function

$$J(\boldsymbol{u}, x) := \sum_{k=1}^{N} [k/N]^m \Big[\varphi_m \ell_k^{\boldsymbol{u}} + (1-\varphi_m) V_k^{\boldsymbol{u}}\Big]$$

- ℓ is the stage cost of interest
- V is a **local** controlled Lyapunov function
- $\varphi(0) = 1$ and $\lim_{m\to\infty} \varphi_m = 0$
- **Assumption**: N-reachability over $\mathbb{X}_0$

Proposition. For sufficiently high m $\rightarrow$ Closed-loop stability of for any initial state inside $\mathbb{X}_0$.

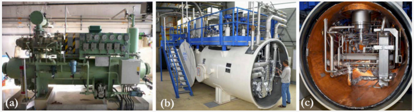
Questions

Q1. Are we still minimizing the same cost?

Q2. Implication on the computation time!

Q1-Q2

Cryogenic station of CEA-INAC-SBT



$$x^+ = Ax + Bu + Gw$$
$$y = Cx + Du \quad (x, u, y, w) \in \mathbb{R}^{24} \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}$$

M.A Stability proof for nonlinear MPC design using monotonically increasing weighting . . . , **Automatica 87**, 2018.
M.A Numerical investigation regarding . . . Application to the control of real-life cryogenic plant, **NMPC workshop** 2018.

Bringing infinity closer!

MPC with exponentially increasing weight

Cost function

$$J(\boldsymbol{u}, x) := \sum_{k=1}^{N} [k/N]^m \left[\varphi_m \ell_k^{\boldsymbol{u}} + (1-\varphi_m)V_k^{\boldsymbol{u}}\right]$$

- ℓ is the stage cost of interest
- V is a **local** controlled Lyapunov function
- $\varphi(0) = 1$ and $\lim_{m\to\infty} \varphi_m = 0$
- **Assumption**: N-reachability over $\mathbb{X}_0$

Proposition. For sufficiently high m $\to$ Closed-loop stability of for any initial state inside $\mathbb{X}_0$.

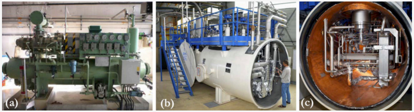
Questions

Q1. Are we still minimizing the same cost?

Q2. Implication on the computation time!

Q1-Q2

Cryogenic station of CEA-INAC-SBT



$$x^+ = Ax + Bu + Gw$$
$$y = Cx + Du \quad (x, u, y, w) \in \mathbb{R}^{24} \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}$$

$$\varepsilon_f \|x_N\|^2 + \sum_{k=1}^{N} [k/N]^m \left[\|y_i\|^2_{Q_y} + \varepsilon\|x_i\|^2\right]$$

Controller Settings	ϵ_f	m	ϵ	N
Proposed Controller	0	3	0.01	10
Controller 1	0	0	0.01	10
Controller 2	100	0	0.01	10

M.A Stability proof for nonlinear MPC design using monotonically increasing weighting . . . , **Automatica 87**, 2018.
M.A Numerical investigation regarding . . . Application to the control of real-life cryogenic plant, **NMPC workshop** 2018.

Bringing infinity closer!

MPC with exponentially increasing weight

Cost function

$$J(\boldsymbol{u}, x) := \sum_{k=1}^{N} [k/N]^m \Big[\varphi_m \ell_k^{\boldsymbol{u}} + (1-\varphi_m) V_k^{\boldsymbol{u}}\Big]$$

- ℓ is the stage cost of interest
- V is a **local** controlled Lyapunov function
- $\varphi(0) = 1$ and $\lim_{m\to\infty} \varphi_m = 0$
- **Assumption**: N-reachability over $\mathbb{X}_0$

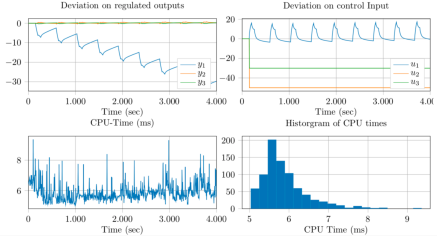
Proposition. For sufficiently high m → Closed-loop stability of for any initial state inside $\mathbb{X}_0$.

Questions

Q1. Are we still minimizing the same cost?

Q2. Implication on the computation time!

Q1-Q2 **Controller 1**



$$\varepsilon_f \|x_N\|^2 + \sum_{k=1}^{N} [k/N]^m \Big[\|y_i\|^2_{Q_y} + \varepsilon \|x_i\|^2\Big]$$

Controller Settings	ϵ_f	m	ϵ	N
Proposed Controller	0	3	0.01	10
Controller 1	0	0	0.01	10
Controller 2	100	0	0.01	10

M.A Stability proof for nonlinear MPC design using monotonically increasing weighting . . . , **Automatica 87**, 2018.
M.A Numerical investigation regarding . . . Application to the control of real-life cryogenic plant, **NMPC workshop** 2018.

Bringing infinity closer!

MPC with exponentially increasing weight

Cost function

$$J(\boldsymbol{u}, x) := \sum_{k=1}^{N} [k/N]^m \Big[\varphi_m \ell_k^{\boldsymbol{u}} + (1-\varphi_m) V_k^{\boldsymbol{u}}\Big]$$

- ℓ is the stage cost of interest
- V is a **local** controlled Lyapunov function
- $\varphi(0) = 1$ and $\lim_{m\to\infty} \varphi_m = 0$
- **Assumption**: N-reachability over $\mathbb{X}_0$

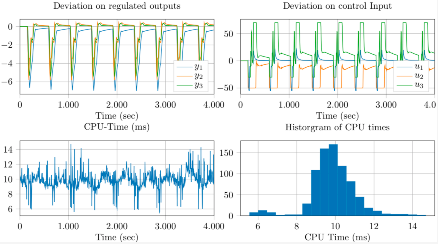
Proposition. For sufficiently high m $\rightarrow$ Closed-loop stability of for any initial state inside $\mathbb{X}_0$.

Questions

Q1. Are we still minimizing the same cost?

Q2. Implication on the computation time!

Q1-Q2 **Controller 2**



$$\varepsilon_f \|x_N\|^2 + \sum_{k=1}^{N} [k/N]^m \Big[\|y_i\|^2_{Q_y} + \varepsilon \|x_i\|^2\Big]$$

Controller Settings	ϵ_f	m	ϵ	N
Proposed Controller	0	3	0.01	10
Controller 1	0	0	0.01	10
Controller 2	100	0	0.01	10

M.A Stability proof for nonlinear MPC design using monotonically increasing weighting . . . , **Automatica 87**, 2018.
M.A Numerical investigation regarding . . . Application to the control of real-life cryogenic plant, **NMPC workshop** 2018.
M.A A new contraction-based NMPC formulation without stability-related constraints. **Automatica**, 2017.

Bringing infinity closer!

MPC with exponentially increasing weight

Cost function

$$J(\boldsymbol{u}, x) := \sum_{k=1}^{N} [k/N]^m \Big[\varphi_m \ell_k^{\boldsymbol{u}} + (1-\varphi_m) V_k^{\boldsymbol{u}}\Big]$$

- ℓ is the stage cost of interest
- V is a **local** controlled Lyapunov function
- $\varphi(0) = 1$ and $\lim_{m\to\infty} \varphi_m = 0$
- **Assumption**: N-reachability over $\mathbb{X}_0$

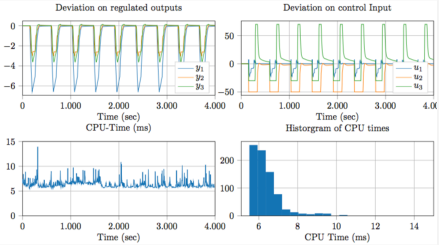
Proposition. For sufficiently high m $\rightarrow$ Closed-loop stability of for any initial state inside $\mathbb{X}_0$.

Questions

Q1. Are we still minimizing the same cost?

Q2. Implication on the computation time!

Q1-Q2 **Proposed**



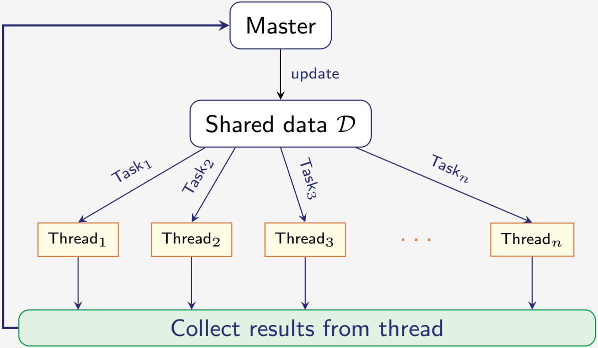
$$\varepsilon_f \|x_N\|^2 + \sum_{k=1}^{N} [k/N]^m \Big[\|y_i\|^2_{Q_y} + \varepsilon \|x_i\|^2\Big]$$

Controller Settings	ϵ_f	m	ϵ	N
Proposed Controller	0	3	0.01	10
Controller 1	0	0	0.01	10
Controller 2	100	0	0.01	10

M.A Stability proof for nonlinear MPC design using monotonically increasing weighting . . . , **Automatica 87**, 2018.
M.A Numerical investigation regarding . . . Application to the control of real-life cryogenic plant, **NMPC workshop** 2018.
M.A A new contraction-based NMPC formulation without stability-related constraints. **Automatica**, 2017.

Distributing the computation

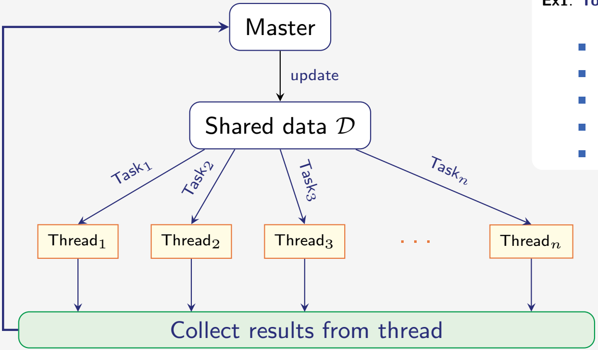
GPU, FPGA ...



$n \to 10^7$

Distributing the computation

GPU, FPGA ...



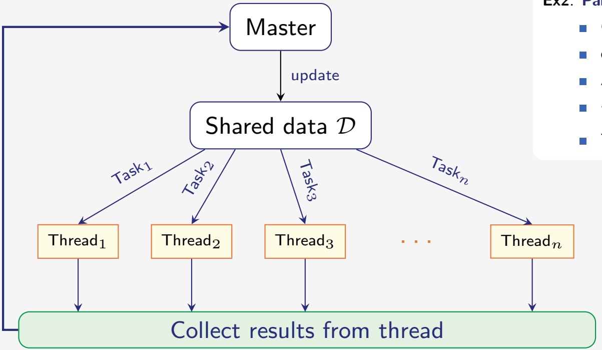
$n \to 10^7$

Ex1. Toy example on Google-Colab

- $2D$ discrete time dynamics
- simulate $N = 20$ steps
- for ≈ 500000 initial states
- time CPU $\approx$ 2 min.
- **GPU-time $\approx$ 400 μ-sec**

Distributing the computation

GPU, FPGA ...



$n \to 10^7$

Ex2. Parametrized stochastic MPC

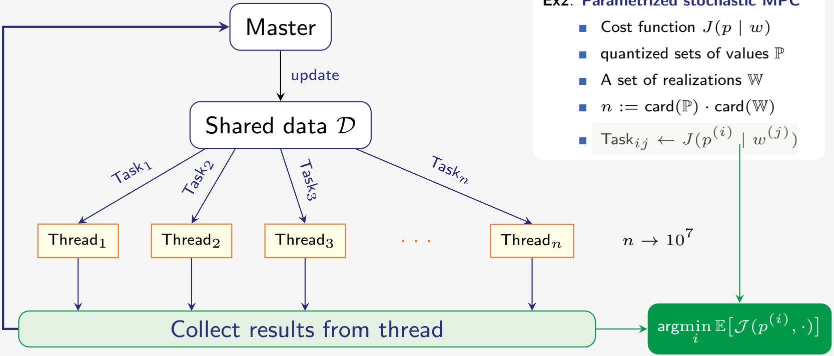
- Cost function $J(p \mid w)$
- quantized sets of values $\mathbb{P}$
- A set of realizations $\mathbb{W}$
- $n :=$ card$(\mathbb{P}) \cdot$ card$(\mathbb{W})$
- Task$_{ij} \leftarrow J(p^{(i)} \mid w^{(j)})$

Distributing the computation

GPU, FPGA ...

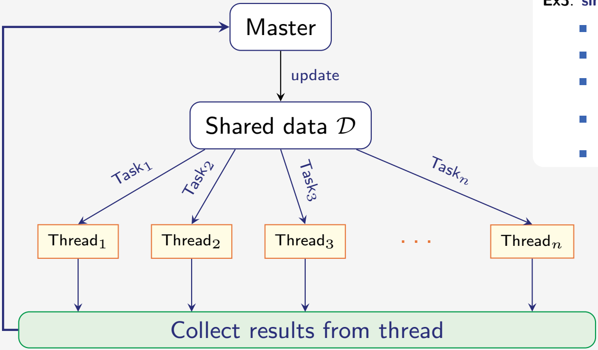
Ex2. Parametrized stochastic MPC

- Cost function $J(p \mid w)$
- quantized sets of values $\mathbb{P}$
- A set of realizations $\mathbb{W}$
- $n :=$ card$(\mathbb{P}) \cdot$ card$(\mathbb{W})$
- Task$_{ij} \leftarrow J(p^{(i)} \mid w^{(j)})$



DISTRIBUTING THE COMPUTATION

GPU, FPGA ...



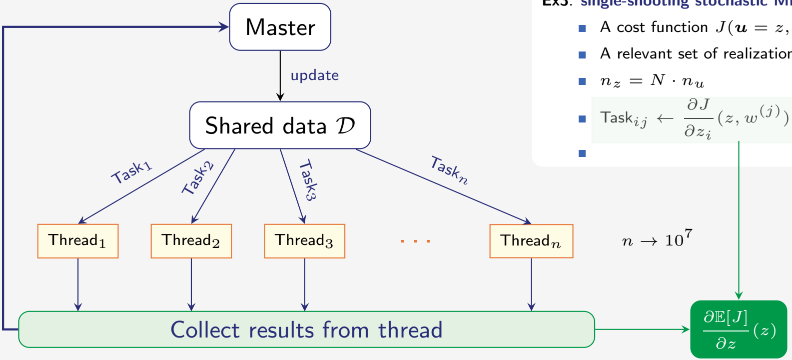
Ex3. single-shooting stochastic MPC

- A cost function $J(\boldsymbol{u} = z, w)$
- A relevant set of realizations $\mathbb{W}$
- $n_z = N \cdot n_u$
- $\text{Task}_{ij} \leftarrow \dfrac{\partial J}{\partial z_i}(z, w^{(j)})$
-

$n \to 10^7$

Distributing the computation

GPU, FPGA ...



Ex3. single-shooting stochastic MPC

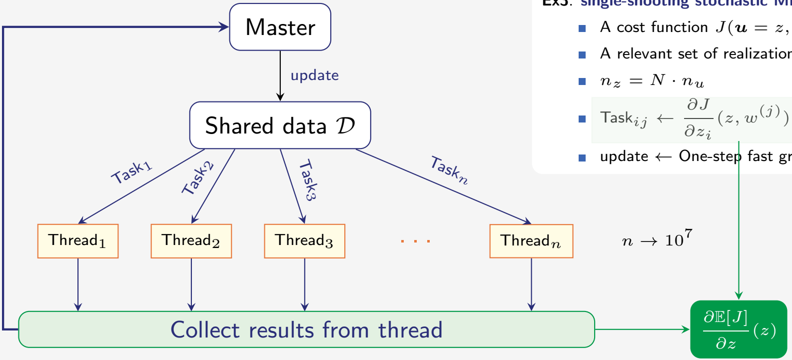
- A cost function $J(\boldsymbol{u} = z, w)$
- A relevant set of realizations $\mathbb{W}$
- $n_z = N \cdot n_u$
- $\text{Task}_{ij} \leftarrow \dfrac{\partial J}{\partial z_i}(z, w^{(j)})$
-

Distributing the computation

GPU, FPGA ...

Ex3. single-shooting stochastic MPC

- A cost function $J(\boldsymbol{u} = z, w)$
- A relevant set of realizations $\mathbb{W}$
- $n_z = N \cdot n_u$
- $\text{Task}_{ij} \leftarrow \dfrac{\partial J}{\partial z_i}(z, w^{(j)})$
- update $\leftarrow$ One-step fast gradient!

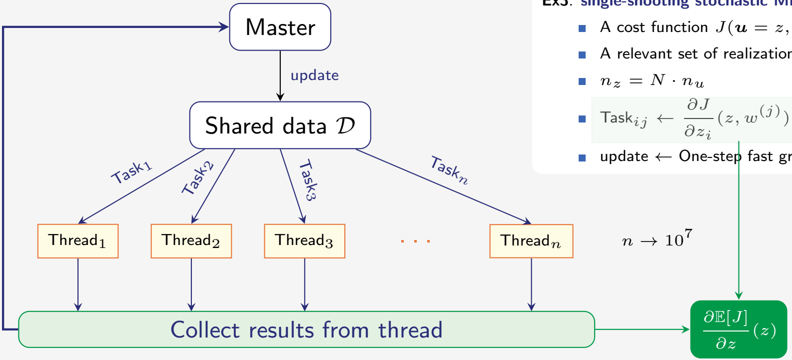


Distributing the computation

GPU, FPGA ...

Ex3. single-shooting stochastic MPC

- A cost function $J(\boldsymbol{u} = z, w)$
- A relevant set of realizations $\mathbb{W}$
- $n_z = N \cdot n_u$
- $\text{Task}_{ij} \leftarrow \dfrac{\partial J}{\partial z_i}(z, w^{(j)})$
- update ← One-step fast gradient!



https://colab.research.google.com/drive/1CJl4IaQ1s23D5Wn_6P-uIIzIgEfsr5ji?usp=sharing

```
Number of scenarios = 20000
Prediction Horizon = 20
gpu computation time = 0.0002663135528564453
Number of individual tasks = 800000
```

```
t1 = time.time()

Grad(x0_gpu, U_gpu, W_gpu, J_gpu, G_gpu, mode_gpu,
     grid=(256, 16, 1), block=(256, 1, 1))
t2 = time.time()
```

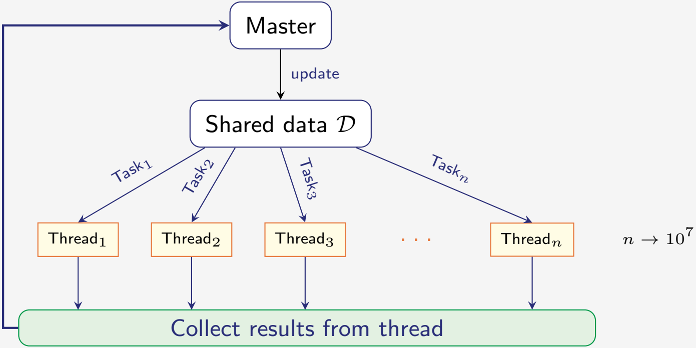
Combined therapy

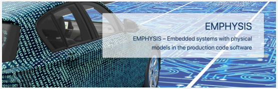
$n_x = 4, n_w = 13$

$n_u = 2, N = 20$

Distributing the computation

GPU, FPGA . . .

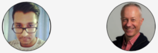




GPU-Based NMPC of a semi-active suspension

Distributing the computation

GPU-Based MPC of a semi-active suspension.



K. M. M. Rathai O. Sename

Half-Car SA-suspension

$$m_s \ddot{z}_s = -F_{s,\ell}(sa_\ell) - F_{s,r}(sa_r)$$

$$I_x \ddot{\theta} = \ell_\ell F_{s,\ell}(sa_\ell) - \ell_r F_{s,r}(sa_r)$$

$$m_{us,\ell} \ddot{z}_{us,\ell} = F_{t,\ell} - F_{s,\ell}(sa_\ell)$$

$$m_{us,r} \ddot{z}_{us,r} = F_{t,r} - F_{s,r}(sa_r)$$

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

$$F_{s,i} = -k_{s,i}(z_{s,i}(\theta) - z_{us,i}) + sa_i$$

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

Road profile ↲

K. M. M. Rathai et al., GPU-based parametrized NMPC scheme . . . IEEE Control Systems Letters, Vol. 3, Number 3, 2019

Distributing the computation

GPU-Based MPC of a semi-active suspension.


K. M. M. Rathai


O. Sename

Half-Car SA-suspension

$$m_s\ddot{z}_s = -F_{s,\ell}(sa_\ell) - F_{s,r}(sa_r)$$
$$I_x\ddot{\theta} = \ell_\ell F_{s,\ell}(sa_\ell) - \ell_r F_{s,r}(sa_r)$$
$$m_{us,\ell}\ddot{z}_{us,\ell} = F_{t,\ell} - F_{s,\ell}(sa_\ell)$$
$$m_{us,r}\ddot{z}_{us,r} = F_{t,r} - F_{s,r}(sa_r)$$

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

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

Road profile

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

K. M. M. Rathai et al., GPU-based parametrized NMPC scheme . . . IEEE Control Systems Letters, Vol. 3, Number 3, 2019

Distributing the computation

GPU-Based MPC of a semi-active suspension.

K. M. M. Rathai

O. Sename

Half-Car SA-suspension

$$m_s\ddot{z}_s = -F_{s,\ell}(sa_\ell) - F_{s,r}(sa_r)$$

$$I_x\ddot{\theta} = \ell_\ell F_{s,\ell}(sa_\ell) - \ell_r F_{s,r}(sa_r)$$

$$m_{us,\ell}\ddot{z}_{us,\ell} = F_{t,\ell} - F_{s,\ell}(sa_\ell)$$

$$m_{us,r}\ddot{z}_{us,r} = F_{t,r} - F_{s,r}(sa_r)$$

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

$$F_{s,i} = -k_{s,i}(z_{s,i}(\theta) - z_{us,i}) + sa_i$$

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

Road profile

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

Control objective

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

K. M. M. Rathai et al., GPU-based parametrized NMPC scheme . . . IEEE Control Systems Letters, Vol. 3, Number 3, 2019

Distributing the computation

GPU-Based MPC of a semi-active suspension.


K. M. M. Rathai


O. Sename

Half-Car SA-suspension

$$m_s\ddot{z}_s = -F_{s,\ell}(sa_\ell) - F_{s,r}(sa_r)$$

$$I_x\ddot{\theta} = \ell_\ell F_{s,\ell}(sa_\ell) - \ell_r F_{s,r}(sa_r)$$

$$m_{us,\ell}\ddot{z}_{us,\ell} = F_{t,\ell} - F_{s,\ell}(sa_\ell)$$

$$m_{us,r}\ddot{z}_{us,r} = F_{t,r} - F_{s,r}(sa_r)$$

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

$$F_{s,i} = -k_{s,i}(z_{s,i}(\theta) - z_{us,i}) + sa_i$$

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

Road profile

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

Control objective

- Comfort $(\downarrow |\ddot{z}_s|)$
- Ride handling $(\downarrow |\ddot{\theta}^2|)$
- 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.

K. M. M. Rathai et al., GPU-based parametrized NMPC scheme . . . IEEE Control Systems Letters, Vol. 3, Number 3, 2019

Distributing the computation

GPU-Based MPC of a semi-active suspension.

K. M. M. Rathai

O. Sename

Half-Car SA-suspension

$$m_s \ddot{z}_s = -F_{s,\ell}(sa_\ell) - F_{s,r}(sa_r)$$
$$I_x \ddot{\theta} = \ell_\ell F_{s,\ell}(sa_\ell) - \ell_r F_{s,r}(sa_r)$$
$$m_{us,\ell}\ddot{z}_{us,\ell} = F_{t,\ell} - F_{s,\ell}(sa_\ell)$$
$$m_{us,r}\ddot{z}_{us,r} = F_{t,r} - F_{s,r}(sa_r)$$

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

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

Road profile

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

Control objective

- Comfort $(\downarrow |\ddot{z}_s|)$
- Ride handling $(\downarrow |\ddot{\theta}^2|)$
- 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.

$\dot{x} = f(x, u, w)$, $(x, u, w) \in \mathbb{R}^8 \times \mathbb{R}^2 \times \mathbb{R}^2$, $\tau_s \leq$ 5msec

K. M. M. Rathai et al., GPU-based parametrized NMPC scheme ... IEEE Control Systems Letters, Vol. 3, Number 3, 2019

Distributing the computation

GPU-Based MPC of a semi-active suspension.


K. M. M. Rathai


O. Sename

Half-Car SA-suspension

$$m_s\ddot{z}_s = -F_{s,\ell}(sa_\ell) - F_{s,r}(sa_r)$$
$$I_x\ddot{\theta} = \ell_\ell F_{s,\ell}(sa_\ell) - \ell_r F_{s,r}(sa_r)$$
$$m_{us,\ell}\ddot{z}_{us,\ell} = F_{t,\ell} - F_{s,\ell}(sa_\ell)$$
$$m_{us,r}\ddot{z}_{us,r} = F_{t,r} - F_{s,r}(sa_r)$$

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

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

Road profile

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

Control objective

- Comfort $(\downarrow |\ddot{z}_s|)$
- Ride handling $(\downarrow |\ddot{\theta}^2|)$
- 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.

$$\dot{x} = f(x, u, w),\ (x, u, w) \in \mathbb{R}^8 \times \mathbb{R}^2 \times \mathbb{R}^2\ ,\ \tau_s \leq 5\text{msec}$$

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

Matlab/simulink, Intel Core i7 PC

K. M. M. Rathai et al., GPU-based parametrized NMPC scheme . . . IEEE Control Systems Letters, Vol. 3, Number 3, 2019

Distributing the computation

GPU-Based MPC of a semi-active suspension.


K. M. M. Rathai


O. Sename

Half-Car SA-suspension

$$m_s \ddot{z}_s = -F_{s,\ell}(sa_\ell) - F_{s,r}(sa_r)$$
$$I_x \ddot{\theta} = \ell_\ell F_{s,\ell}(sa_\ell) - \ell_r F_{s,r}(sa_r)$$
$$m_{us,\ell}\ddot{z}_{us,\ell} = F_{t,\ell} - F_{s,\ell}(sa_\ell)$$
$$m_{us,r}\ddot{z}_{us,r} = F_{t,r} - F_{s,r}(sa_r)$$

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

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

Road profile

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

Control objective

- Comfort $(\downarrow |\ddot{z}_s|)$
- Ride handling $(\downarrow |\ddot{\theta}^2|)$
- 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.

$\dot{x} = f(x, u, w)$, $(x, u, w) \in \mathbb{R}^8 \times \mathbb{R}^2 \times \mathbb{R}^2$, $\tau_s \leq$ 5msec

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

Matlab/simulink, Intel Core i7 PC

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

NVIDIA GTX 1050 / 768 CUDA cores

K. M. M. Rathai et al., GPU-based parametrized NMPC scheme . . . IEEE Control Systems Letters, Vol. 3, Number 3, 2019

Distributing the computation

GPU-Based MPC of a semi-active suspension.


K. M. M. Rathai

O. Sename

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

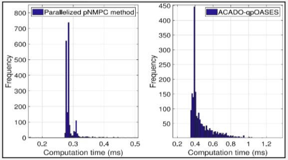
TABLE I
ACADO-qpOASES NMPC Controller

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)

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

Control objective

- Comfort ($\downarrow |\ddot{z}_s|$)
- Ride handling ($\downarrow |\ddot{\theta}^2|$)
- 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.

$$\dot{x} = f(x, u, w),\ (x, u, w) \in \mathbb{R}^8 \times \mathbb{R}^2 \times \mathbb{R}^2\ ,\ \tau_s \leq 5\text{msec}$$

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

Matlab/simulink, Intel Core i7 PC

GPU/Simulation based solver

- Constant profile
- 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

NVIDIA GTX 1050 / 768 CUDA cores

K. M. M. Rathai et al., GPU-based parametrized NMPC scheme . . . IEEE Control Systems Letters, Vol. 3, Number 3, 2019

Stochastic MPC using supervised clustering

Distributing the data building over the real-life time

Dynamics	$\dot{x} = f(x, u, w)$
Cost	$J(\boldsymbol{u}\vert(x, w))$
Constraints	$g(\boldsymbol{u}\vert(x, w))$
w**-statistics**	$\mathcal{W}$

M.A On the use of supervised clustering in stochastic NMPC design. IEEE-TAC. Volume 65, Issue 12, 2020.

Stochastic MPC using supervised clustering

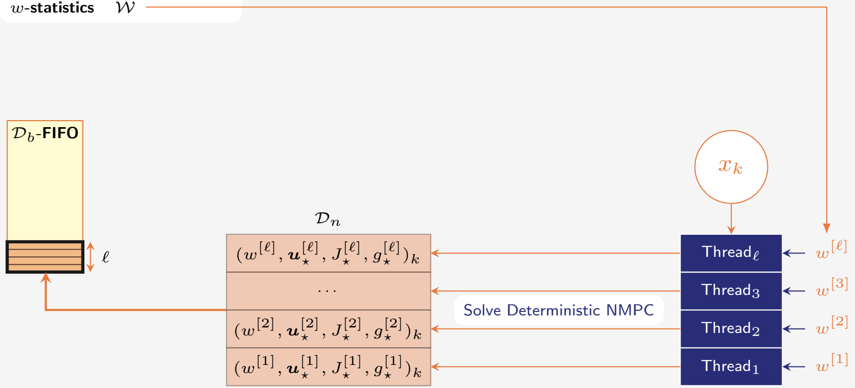
Distributing the data building over the real-life time

Dynamics $\dot{x} = f(x, u, w)$

Cost $J(\boldsymbol{u}|(x, w))$

Constraints $g(\boldsymbol{u}|(x, w))$

w**-statistics** $\mathcal{W}$

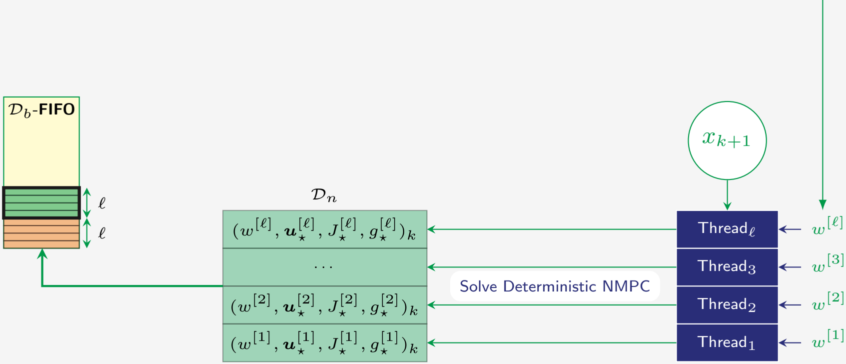


M.A On the use of supervised clustering in stochastic NMPC design. IEEE-TAC. Volume 65, Issue 12, 2020.

Stochastic MPC using supervised clustering

Distributing the data building over the real-life time

Dynamics	$\dot{x} = f(x, u, w)$
Cost	$J(\boldsymbol{u}\vert(x, w))$
Constraints	$g(\boldsymbol{u}\vert(x, w))$
w-statistics	$\mathcal{W}$

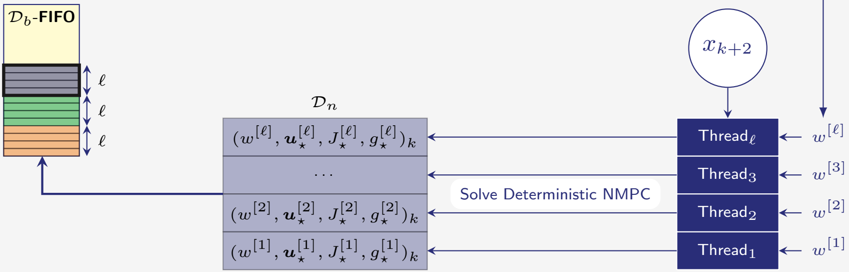


M.A On the use of supervised clustering in stochastic NMPC design. IEEE-TAC. Volume 65, Issue 12, 2020.

Stochastic MPC using supervised clustering

Distributing the data building over the real-life time

Dynamics	$\dot{x} = f(x, u, w)$
Cost	$J(\boldsymbol{u}\|(x, w))$
Constraints	$g(\boldsymbol{u}\|(x, w))$
w**-statistics**	$\mathcal{W}$



M.A On the use of supervised clustering in stochastic NMPC design. IEEE-TAC. Volume 65, Issue 12, 2020.

Stochastic MPC using supervised clustering

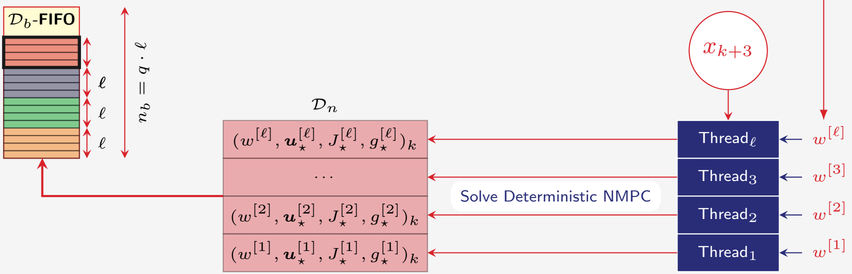
Distributing the data building over the real-life time

Dynamics $\dot{x} = f(x, u, w)$

Cost $J(\boldsymbol{u}|(x, w))$

Constraints $g(\boldsymbol{u}|(x, w))$

w**-statistics** $\mathcal{W}$

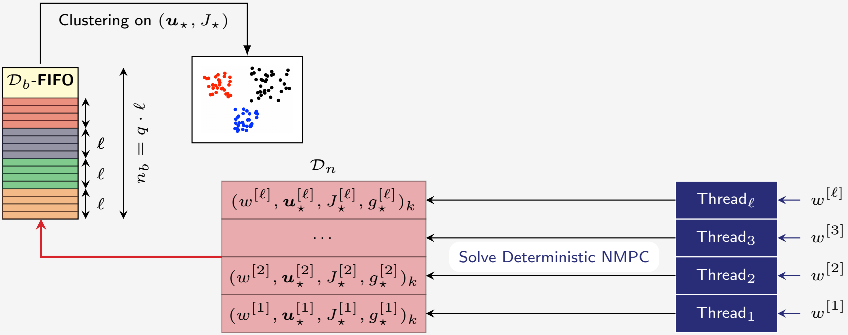


M.A On the use of supervised clustering in stochastic NMPC design. IEEE-TAC. Volume 65, Issue 12, 2020.

STOCHASTIC MPC USING SUPERVISED CLUSTERING

Distributing the data building over the real-life time

Dynamics	$\dot{x} = f(x, u, w)$
Cost	$J(\boldsymbol{u}\|(x, w))$
Constraints	$g(\boldsymbol{u}\|(x, w))$
w**-statistics**	$\mathcal{W}$

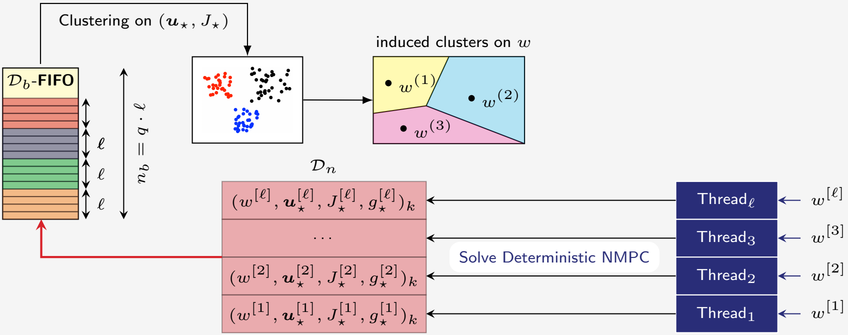


M.A On the use of supervised clustering in stochastic NMPC design. IEEE-TAC. Volume 65, Issue 12, 2020.

Stochastic MPC using supervised clustering

Distributing the data building over the real-life time

Dynamics	$\dot{x} = f(x, u, w)$
Cost	$J(\boldsymbol{u}\|(x, w))$
Constraints	$g(\boldsymbol{u}\|(x, w))$
w**-statistics**	$\mathcal{W}$



M.A On the use of supervised clustering in stochastic NMPC design. IEEE-TAC. Volume 65, Issue 12, 2020.

Stochastic MPC using supervised clustering

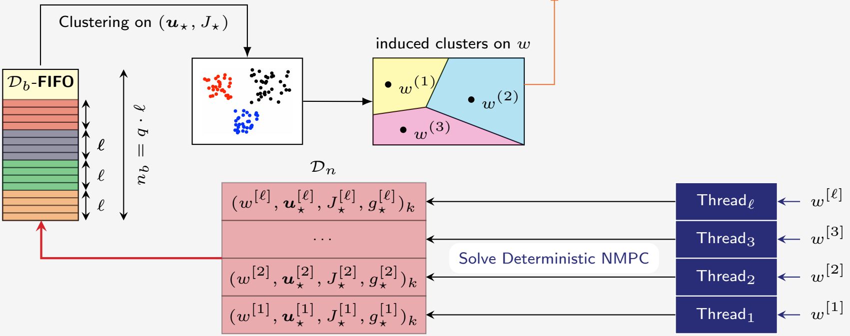
Distributing the data building over the real-life time

Dynamics	$\dot{x} = f(x, u, w)$
Cost	$J(\boldsymbol{u}\|(x, w))$
Constraints	$g(\boldsymbol{u}\|(x, w))$
w-statistics	$\mathcal{W}$

$$\min_{\boldsymbol{u},\mu\geq 0} \sum_{i=1}^{n_{cl}} p_i \cdot J(\boldsymbol{u}\ |(x, w^{(i)}))$$

under

$$g(\boldsymbol{u}\ |(x, w^{(i)})) + \frac{1-\epsilon}{\epsilon}\sigma_g^{(i)} \leq \mu \qquad i = 1, \dots, n_{cl}$$



M.A On the use of supervised clustering in stochastic NMPC design. IEEE-TAC. Volume 65, Issue 12, 2020.

Stochastic MPC using supervised clustering

Distributing the data building over the real-life time

Dynamics	$\dot{x} = f(x, u, w)$
Cost	$J(\boldsymbol{u}\vert(x, w))$
Constraints	$g(\boldsymbol{u}\vert(x, w))$
w**-statistics**	$\mathcal{W}$

$$\min_{\boldsymbol{u},\mu\geq 0} \sum_{i=1}^{n_{cl}} p_i \cdot J(\boldsymbol{u}\ \vert(x, w^{(i)}))$$

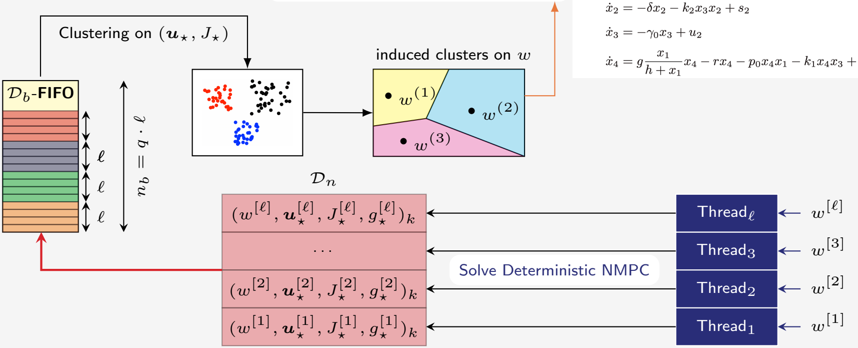
under

$$g(\boldsymbol{u}\ \vert(x, w^{(i)})) + \frac{1-\epsilon}{\epsilon}\sigma_g^{(i)} \leq \mu \qquad i = 1, \dots, n_{cl}$$

x_1 tumor cell population;
x_2 circulating lymphocytes population;
x_3 chemotherapy drug concentration;
x_4 effector immune cell population;
u_1 rate of introduction of immunotherapy drug;
u_2 rate of introduction of chemotherapy drug.

and the dynamics is given by

$$\dot{x}_1 = ax_1(1 - bx_1) - c_1x_4x_1 - k_3x_3x_1$$
$$\dot{x}_2 = -\delta x_2 - k_2x_3x_2 + s_2$$
$$\dot{x}_3 = -\gamma_0 x_3 + u_2$$
$$\dot{x}_4 = g\frac{x_1}{h + x_1}x_4 - rx_4 - p_0x_4x_1 - k_1x_4x_3 + s_1u_1.$$



M.A On the use of supervised clustering in stochastic NMPC design. IEEE-TAC. Volume 65, Issue 12, 2020.

Stochastic MPC using supervised clustering

Distributing the data building over the real-life time

Dynamics	$\dot{x} = f(x, u, w)$
Cost	$J(\boldsymbol{u}\|(x, w))$
Constraints	$g(\boldsymbol{u}\|(x, w))$
w-statistics	$\mathcal{W}$

$$\min_{\boldsymbol{u},\mu\geq 0} \sum_{i=1}^{n_{cl}} p_i \cdot J(\boldsymbol{u}\ |(x, w^{(i)}))$$

under

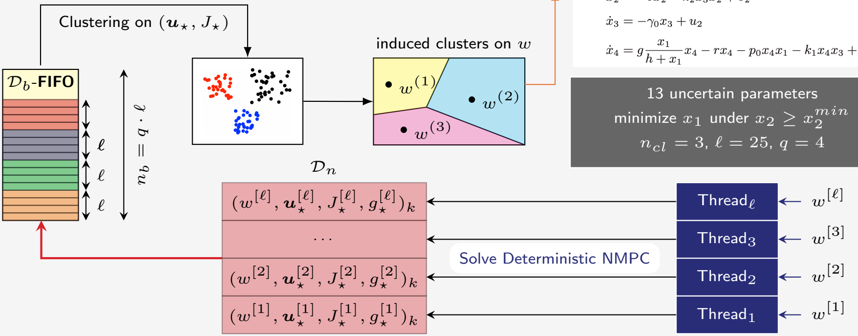
$$g(\boldsymbol{u}\ |(x, w^{(i)})) + \frac{1-\epsilon}{\epsilon}\sigma_g^{(i)} \leq \mu$$
$$i = 1, \dots, n_{cl}$$

x_1 tumor cell population;
x_2 circulating lymphocytes population;
x_3 chemotherapy drug concentration;
x_4 effector immune cell population;
u_1 rate of introduction of immunotherapy drug;
u_2 rate of introduction of chemotherapy drug.

and the dynamics is given by

$$\dot{x}_1 = ax_1(1 - bx_1) - c_1x_4x_1 - k_3x_3x_1$$
$$\dot{x}_2 = -\delta x_2 - k_2x_3x_2 + s_2$$
$$\dot{x}_3 = -\gamma_0x_3 + u_2$$
$$\dot{x}_4 = g\frac{x_1}{h + x_1}x_4 - rx_4 - p_0x_4x_1 - k_1x_4x_3 + s_1u_1.$$

13 uncertain parameters
minimize x_1 under $x_2 \geq x_2^{min}$
$n_{cl} = 3$, $\ell = 25$, $q = 4$



M.A On the use of supervised clustering in stochastic NMPC design. IEEE-TAC. Volume 65, Issue 12, 2020.

Stochastic MPC using supervised clustering

Distributing the data building over the real-life time

Dynamics $\dot{x} = f(x, u, w)$

Cost $J(\boldsymbol{u}|(x, w))$

Constraints $g(\boldsymbol{u}|(x, w))$

w**-statistics** $\mathcal{W}$

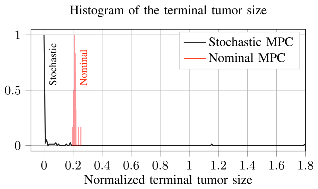
→

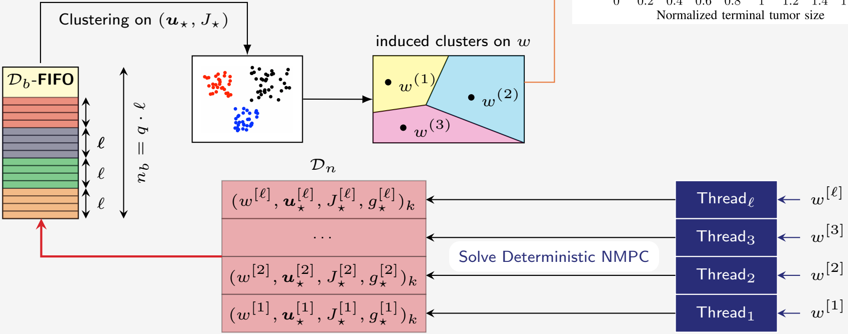
$$\min_{\boldsymbol{u}, \mu \geq 0} \sum_{i=1}^{n_{cl}} p_i \cdot J(\boldsymbol{u} \,|(x, w^{(i)}))$$

under

$$g(\boldsymbol{u} \,|(x, w^{(i)})) + \frac{1-\epsilon}{\epsilon}\sigma_g^{(i)} \leq \mu$$

$$i = 1, \dots, n_{cl}$$





M.A On the use of supervised clustering in stochastic NMPC design. IEEE-TAC. Volume 65, Issue 12, 2020.

Stochastic MPC using supervised clustering

Distributing the data building over the real-life time

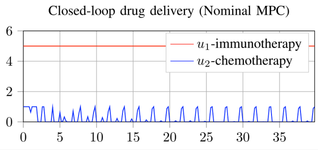
Dynamics	$\dot{x} = f(x, u, w)$
Cost	$J(\boldsymbol{u}\|(x, w))$
Constraints	$g(\boldsymbol{u}\|(x, w))$
w-statistics	$\mathcal{W}$

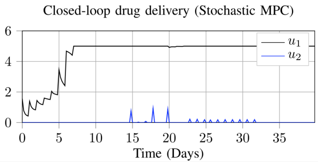
$$\min_{\boldsymbol{u},\mu\geq 0} \sum_{i=1}^{n_{cl}} p_i \cdot J(\boldsymbol{u}\ |(x, w^{(i)}))$$

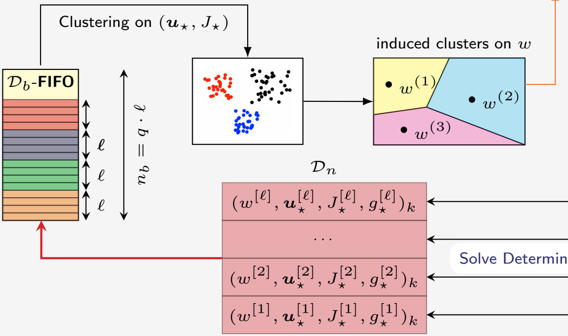
under

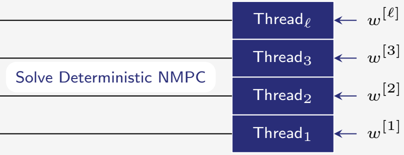
$$g(\boldsymbol{u}\ |(x, w^{(i)})) + \frac{1-\epsilon}{\epsilon}\sigma_g^{(i)} \leq \mu$$

$$i = 1, \dots, n_{cl}$$









M.A On the use of supervised clustering in stochastic NMPC design. IEEE-TAC. Volume 65, Issue 12, 2020.

Stochastic MPC using supervised clustering

Distributing the data building over the real-life time

Dynamics $\dot{x} = f(x, u, w)$

Cost $J(\boldsymbol{u}|(x, w))$

Constraints $g(\boldsymbol{u}|(x, w))$

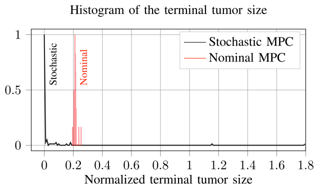
w**-statistics** $\mathcal{W}$

$$\min_{\boldsymbol{u},\mu \geq 0} \sum_{i=1}^{n_{cl}} p_i \cdot J(\boldsymbol{u} \,|(x, w^{(i)}))$$

under

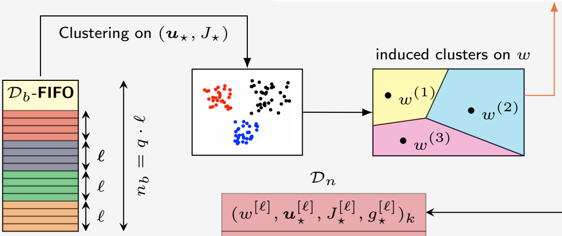
$$g(\boldsymbol{u} \,|(x, w^{(i)})) + \frac{1-\epsilon}{\epsilon}\sigma_g^{(i)} \leq \mu$$

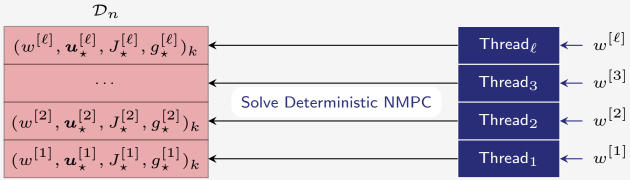
$$i = 1, \dots, n_{cl}$$



Statistics of the Computation Times (in ms) of a Single Solution of the Optimization Problem. (CasADi [4] (Python Version) on a MacBookPro 2.9 GHz Intel Core i7)

	Mean (ms)	**Standard deviation** (ms)
Nominal	270	76
Stochastic	383	330

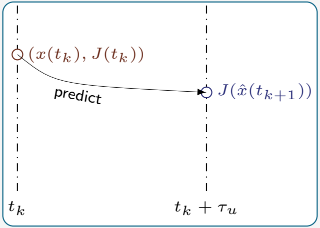




M.A On the use of supervised clustering in stochastic NMPC design. IEEE-TAC. Volume 65, Issue 12, 2020.

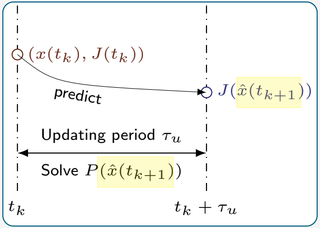
Monitoring Control Updating Period

New ML-induced opportunities (preliminary)



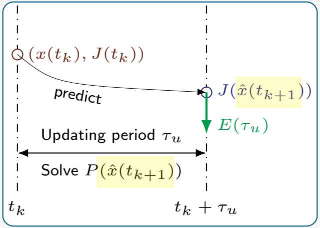
Monitoring Control Updating Period

New ML-induced opportunities (preliminary)



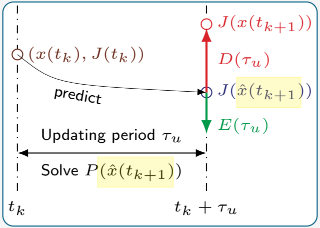
Monitoring Control Updating Period

New ML-induced opportunities (preliminary)



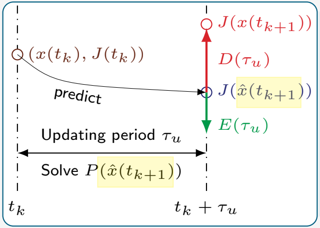
Monitoring Control Updating Period

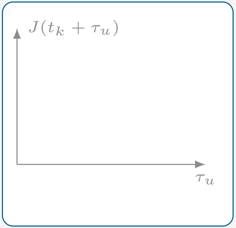
New ML-induced opportunities (preliminary)



Monitoring Control Updating Period

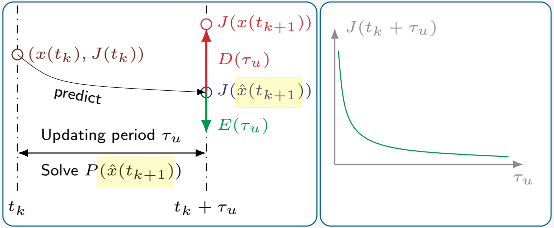
New ML-induced opportunities (preliminary)





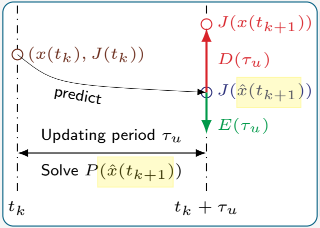
Monitoring Control Updating Period

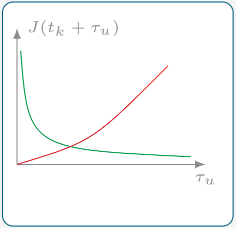
New ML-induced opportunities (preliminary)



Monitoring Control Updating Period

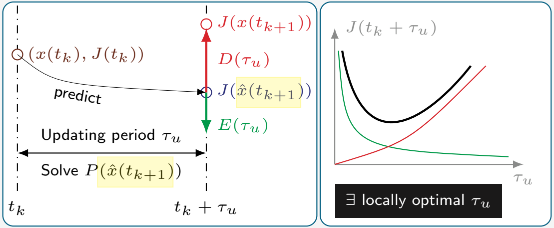
New ML-induced opportunities (preliminary)





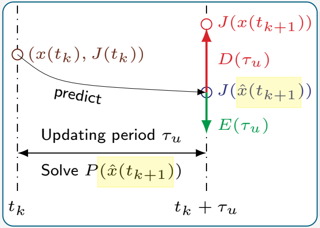
Monitoring Control Updating Period

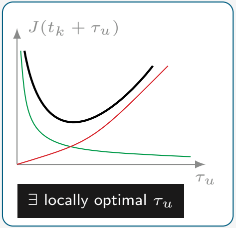
New ML-induced opportunities (preliminary)

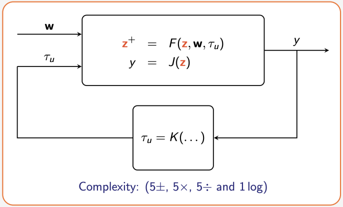


Monitoring Control Updating Period

New ML-induced opportunities (preliminary)



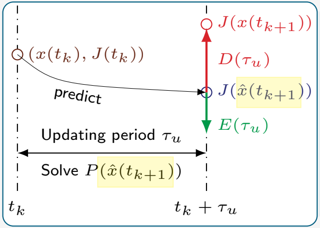


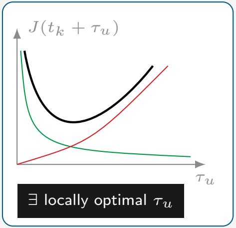


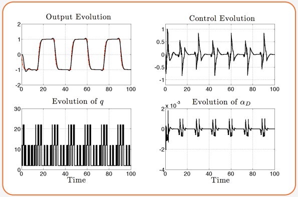
M.A. Monitoring Control Updating Period In Fast Gradient-Based NMPC. ECC2013, 2013

Monitoring Control Updating Period

New ML-induced opportunities (preliminary)





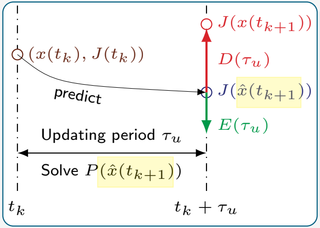


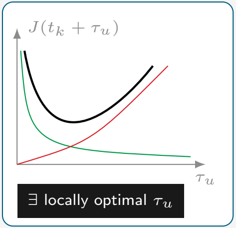
$\tau_u := q \times$ cpu of a single iteration

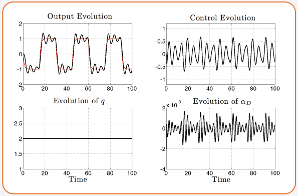
M.A. Monitoring Control Updating Period In Fast Gradient-Based NMPC. ECC2013, 2013

Monitoring Control Updating Period

New ML-induced opportunities (preliminary)





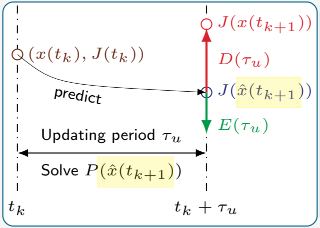


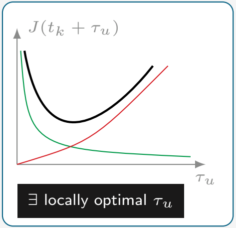
$\tau_u := q \times$ cpu of a single iteration

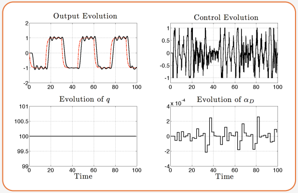
M.A. Monitoring Control Updating Period In Fast Gradient-Based NMPC. ECC2013, 2013

Monitoring Control Updating Period

New ML-induced opportunities (preliminary)





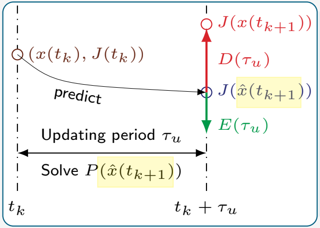


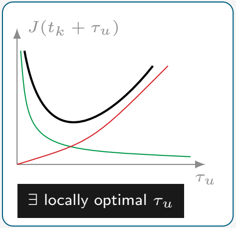
$\tau_u := q \times$ cpu of a single iteration

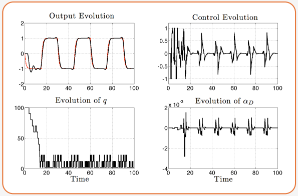
M.A. Monitoring Control Updating Period In Fast Gradient-Based NMPC. ECC2013, 2013

Monitoring Control Updating Period

New ML-induced opportunities (preliminary)





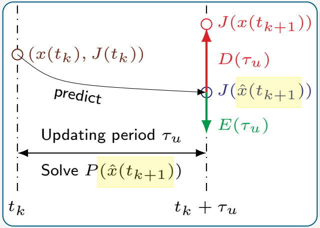


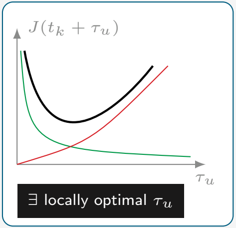
$\tau_u := q \times$ cpu of a single iteration

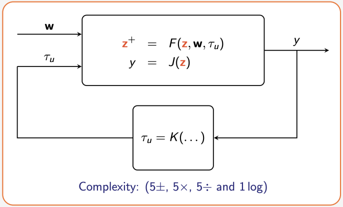
M.A. Monitoring Control Updating Period In Fast Gradient-Based NMPC. ECC2013, 2013

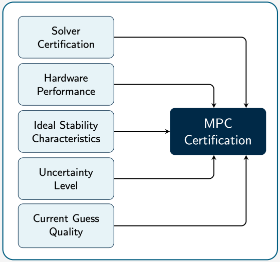
Monitoring Control Updating Period

New ML-induced opportunities (preliminary)





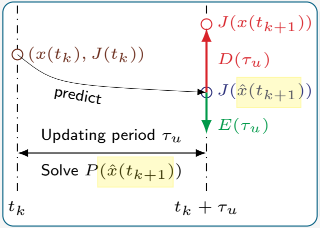


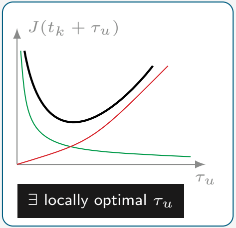


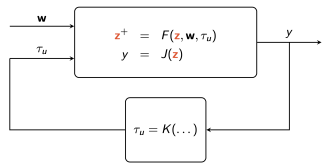
M. Alamir. A State-Dependent Updating Period For Certified Real-Time MPC. IEEE TAC, 2017.

Monitoring Control Updating Period

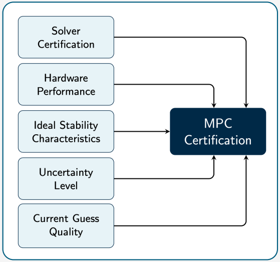
New ML-induced opportunities (preliminary)

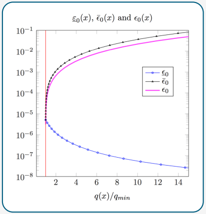






Complexity: (5±, 5×, 5÷ and 1 log)





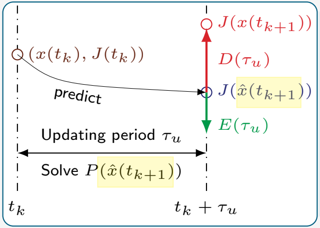
ϵ_0: required precision

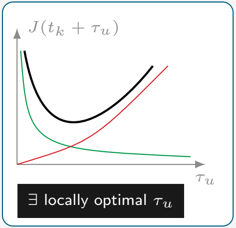
$q(x) \leftarrow$ stage cost

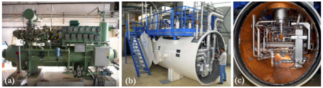
M. Alamir. A State-Dependent Updating Period For Certified Real-Time MPC. IEEE TAC, 2017.

Monitoring Control Updating Period

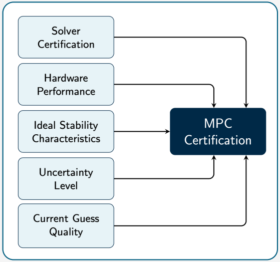
New ML-induced opportunities (preliminary)

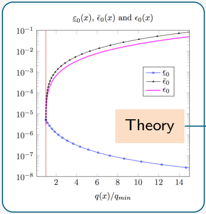






Simulate CL-NMPC
Fast Gradient
Warm start
Random (τ_u, w)





check

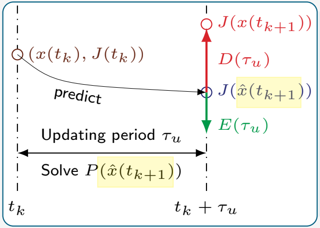
ϵ_0: required precision

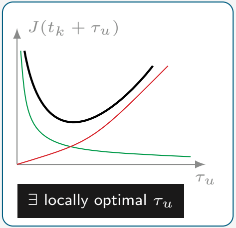
$q(x) \leftarrow$ stage cost

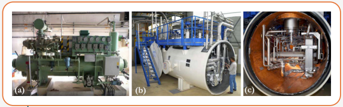
M. Alamir. A State-Dependent Updating Period For Certified Real-Time MPC. IEEE TAC, 2017.

Monitoring Control Updating Period

New ML-induced opportunities (preliminary)





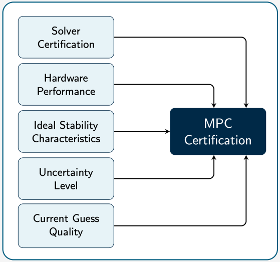


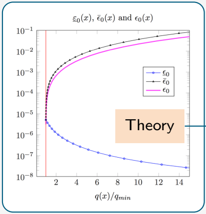
- Simulate CL-NMPC
- Fast Gradient
- Warm start
- Random (τ_u, w)

Learning data:

$$X = (J, \tau_u, w)$$

$$y = \frac{J^+}{J+1}$$





check

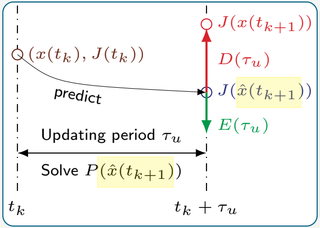
ϵ_0: required precision

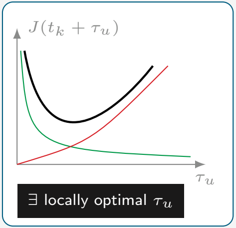
$q(x) \leftarrow$ stage cost

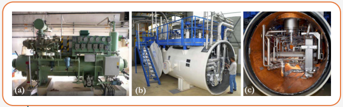
M. Alamir. A State-Dependent Updating Period For Certified Real-Time MPC. IEEE TAC, 2017.

Monitoring Control Updating Period

New ML-induced opportunities (preliminary)





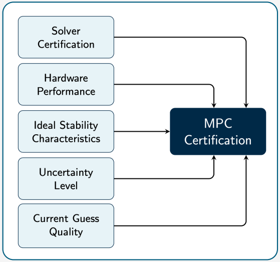


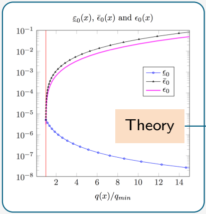
- Simulate CL-NMPC
- Fast Gradient
- Warm start
- Random (τ_u, w)

Learning data:

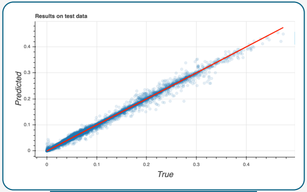
$$X = (J, \tau_u, w)$$

$$y = \frac{J^+}{J+1}$$





check



$J^+ = \text{ML}(J(x), \tau_u, w)$

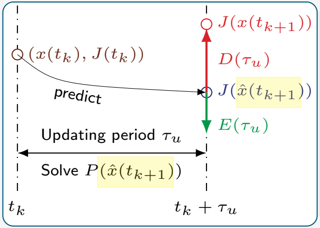
ϵ_0: required precision

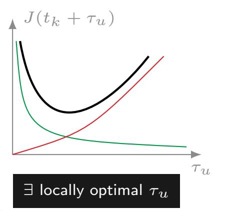
$q(x) \leftarrow$ stage cost

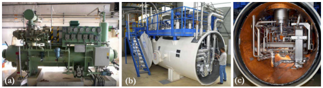
M. Alamir. A State-Dependent Updating Period For Certified Real-Time MPC. IEEE TAC, 2017.

Monitoring Control Updating Period

New ML-induced opportunities (preliminary)





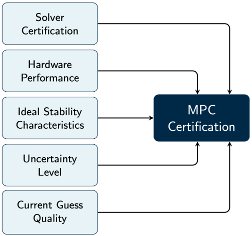


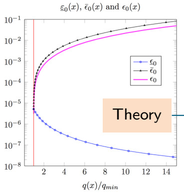
- Simulate CL-NMPC
- Fast Gradient
- Warm start
- Random (τ_u, w)

Learning data:

$$X = (J, \tau_u, w)$$

$$y = \frac{J^+}{J+1}$$





ϵ_0: required precision

$q(x) \leftarrow$ stage cost

check

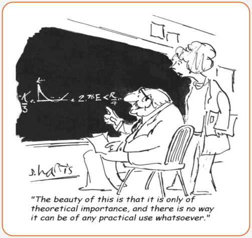
$$J(t_{k+1}) = \textbf{ML}(J(t_k), \tau_u, w)$$

$$t_{k+1} = t_k + \tau_u$$

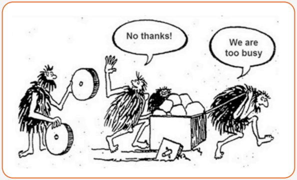
$$\min_{\tau_u(\cdot)} \mathbb{E}(J(T_P)) \text{ s.t. } \begin{cases} \tau_u^{(i)} > 0 \\ \sum_i \tau_u^{(i)} = T_p \end{cases}$$

M. Alamir. A State-Dependent Updating Period For Certified Real-Time MPC. IEEE TAC, 2017.

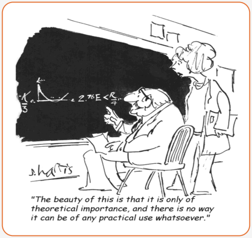
Conclusion



Control

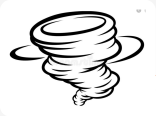


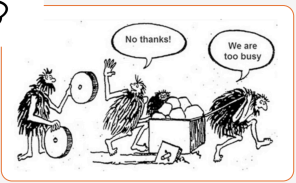
Conclusion



Control

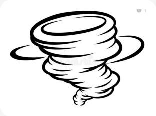
Machine Learning

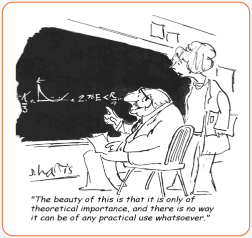




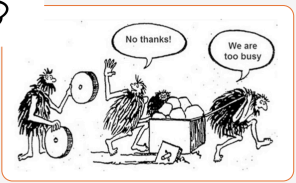
Conclusion

Machine Learning





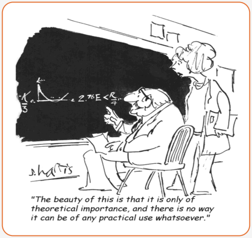
Control



Computational technologies

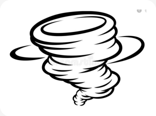
GPU, FPGA, Containers, . . .

Conclusion



Control

Machine Learning

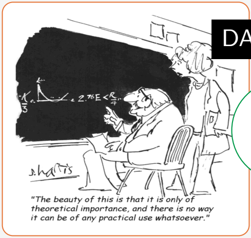


AI-Autonomous X

Computational technologies
GPU, FPGA, Containers, . . .

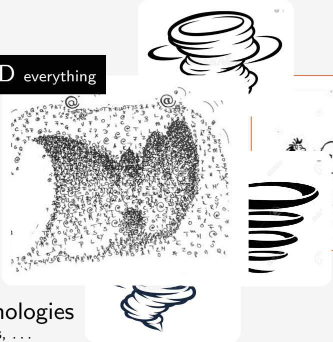
Conclusion

Machine Learning



DATA-BASED everything

Control



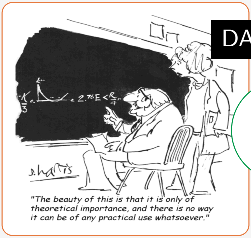
AI-Autonomous X

Computational technologies
GPU, FPGA, Containers, ...

Conclusion

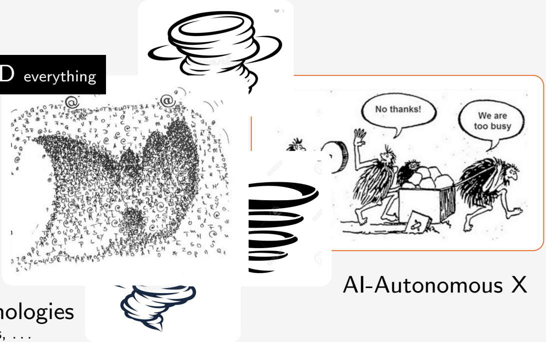
Business as usual is not an option!

Machine Learning



DATA-BASED everything

Control



AI-Autonomous X

Computational technologies
GPU, FPGA, Containers, . . .

Threat: Recycling old recipes . . .

Finding a solution to a real-life problem with provable qualities

$\neq$

Exhibiting sufficient conditions for such a solution to exist.

Threat: Recycling old recipes . . .

Finding a solution to a real-life problem with provable qualities	$\neq$	Exhibiting sufficient conditions for such a solution to exist.

- Once such **conditional** claims are published . . .
- Nobody recalls the **unrealistic** supporting assumptions
- The absence of proofs is opposed to excellent heuristics that would efficiently solve **infinitely wider class** of real-life problems
- This was already nuisible in the past
- It is even more today as other actors (including giants) are working towards efficient and generic solutions

Threat: Recycling old recipes ...

Finding a solution to a real-life problem with provable qualities	$\neq$	Exhibiting sufficient conditions for such a solution to exist.

- Once such **conditional** claims are published ...
- Nobody recalls the **unrealistic** supporting assumptions
- The absence of proofs is opposed to excellent heuristics that would efficiently solve **infinitely wider class** of real-life problems
- This was already nuisible in the past
- It is even more today as other actors (including giants) are working towards efficient and generic solutions

Threat: Recycling old recipes . . .

Finding a solution to a real-life problem with provable qualities

$\neq$

Exhibiting sufficient conditions for such a solution to exist.

- Once such **conditional** claims are published . . .
- Nobody recalls the **unrealistic** supporting assumptions
- The absence of proofs is opposed to excellent heuristics that would efficiently solve **infinitely wider class** of real-life problems
- This was already nuisible in the past
- It is even more today as other actors (including giants) are working towards efficient and generic solutions

Threat: Recycling old recipes . . .

Finding a solution to a real-life problem with provable qualities

$\neq$

Exhibiting sufficient conditions for such a solution to exist.

- Once such **conditional** claims are published . . .
- Nobody recalls the **unrealistic** supporting assumptions
- The absence of proofs is opposed to excellent heuristics that would efficiently solve **infinitely wider class** of real-life problems
- This was already nuisible in the past
- It is even more today as other actors (including giants) are working towards efficient and generic solutions

Threat: Recycling old recipes ...

Finding a solution to a real-life problem with provable qualities	$\neq$	Exhibiting sufficient conditions for such a solution to exist.

- Once such **conditional** claims are published ...
- Nobody recalls the **unrealistic** supporting assumptions
- The absence of proofs is opposed to excellent heuristics that would efficiently solve **infinitely wider class** of real-life problems
- This was already nuisible in the past
- It is even more today as other actors (including giants) are working towards efficient and generic solutions

Threat: Recycling old recipes . . .

Finding a solution to a real-life problem with provable qualities	$\neq$	Exhibiting sufficient conditions for such a solution to exist.

- Once such **conditional** claims are published . . .
- Nobody recalls the **unrealistic** supporting assumptions
- The absence of proofs is opposed to excellent heuristics that would efficiently solve **infinitely wider class** of real-life problems
- This was already nuisible in the past
- It is even more today as other actors (including giants) are working towards efficient and generic solutions

Suggestions . . .

1. Avoid problems for which we already got solutions (including models)
2. Build shared **Blind**, **really challenging & realistic benchmarks**
3. Systematically **evaluate the cost of the learning phase** (ex. RL)
4. Invest on **automated generation** of **physically meaningful** and **control-oriented** models
5. **Share codes!**

Suggestions . . .

1. Avoid problems for which we already got solutions (including models)
2. Build shared **Blind**, **really challenging & realistic benchmarks**
3. Systematically **evaluate the cost of the learning phase** (ex. RL)
4. Invest on **automated generation** of **physically meaningful** and **control-oriented** models
5. **Share codes!**

Suggestions . . .

1. Avoid problems for which we already got solutions (including models)
2. Build shared Blind, **really challenging & realistic benchmarks**
3. Systematically **evaluate the cost of the learning phase** (ex. RL)
4. Invest on **automated generation** of **physically meaningful** and **control-oriented** models
5. **Share codes!**

Suggestions . . .

1. Avoid problems for which we already got solutions (including models)
2. Build shared Blind, **really challenging & realistic benchmarks**
3. Systematically **evaluate the cost of the learning phase** (ex. RL)
4. Invest on **automated generation** of **physically meaningful** and **control-oriented** models
5. **Share codes!**

Suggestions ...

1. Avoid problems for which we already got solutions (including models)
2. Build shared **Blind**, **really challenging & realistic benchmarks**
3. Systematically **evaluate the cost of the learning phase** (ex. RL)
4. Invest on **automated generation** of **physically meaningful** and **control-oriented** models
5. **Share codes!**

Suggestions . . .

1. Avoid problems for which we already got solutions (including models)
2. Build shared **Blind**, **really challenging & realistic benchmarks**
3. Systematically **evaluate the cost of the learning phase** (ex. RL)
4. Invest on **automated generation** of **physically meaningful** and **control-oriented** models
5. **Share codes!**

Take home message . . .

We have assets & strengths!

Let's avoid to get lost in Random Forests! :-)

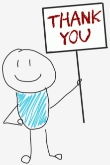
Keep critical and focus on the right fights!

Take home message . . .

We have assets & strengths!

Let's avoid to get lost in Random Forests! :-)

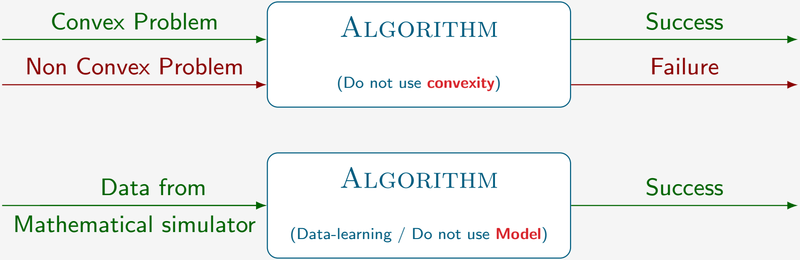
Keep critical and focus on the right fights!



Conjecture

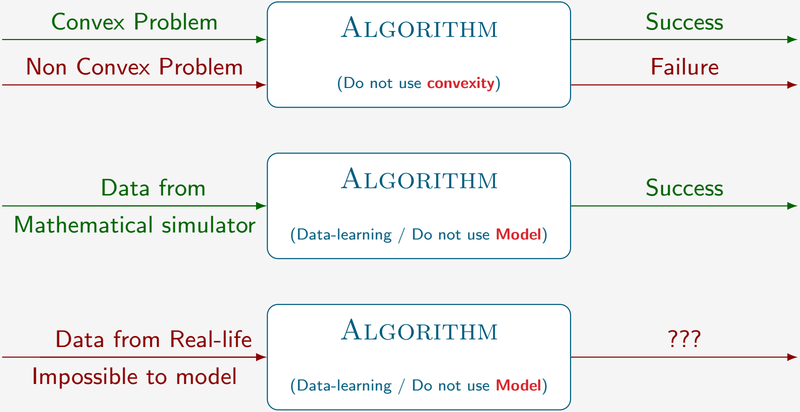

Convex Problem
Non Convex Problem
Algorithm
(Do not use convexity)
Success
Failure

Conjecture

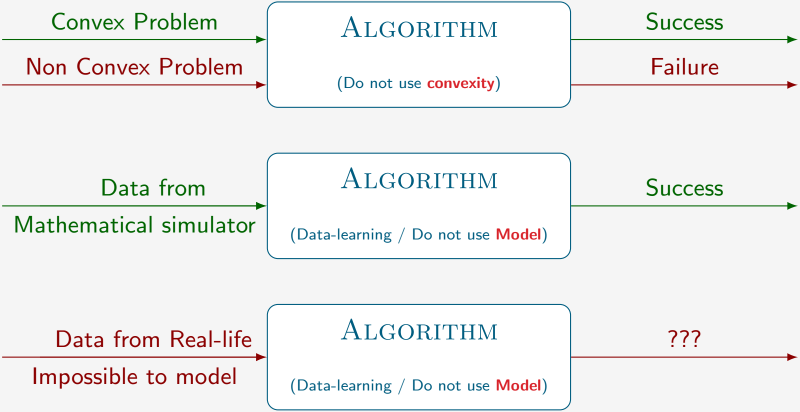


Convex Problem
Non Convex Problem
ALGORITHM
(Do not use convexity)
Success
Failure
Data from
Mathematical simulator
ALGORITHM
(Data-learning / Do not use Model)
Success

Conjecture


Convex Problem
Non Convex Problem
ALGORITHM
(Do not use convexity)
Success
Failure
Data from
Mathematical simulator
ALGORITHM
(Data-learning / Do not use Model)
Success
Data from Real-life
Impossible to model
ALGORITHM
(Data-learning / Do not use Model)
???

Conjecture



What if (Success) $\Leftrightarrow$ ($\exists$ model)?