Security of distributed Model Predictive Control under False Data injection

or How I Learned to Stop and Worry about Everything

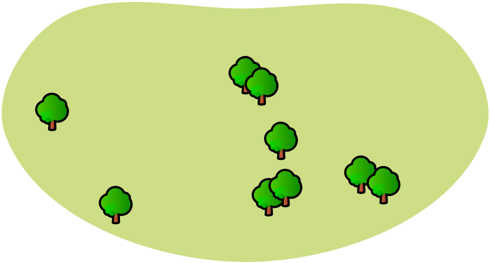
Rafael Accácio NOGUEIRA

December 12, 2022

https://bit.ly/3g3S6X4

Context

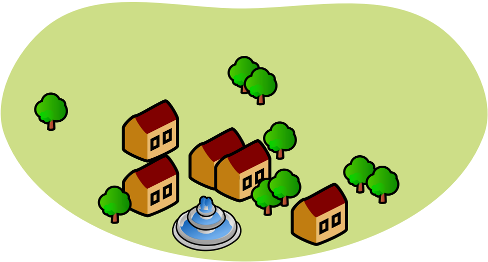
Requirements evolve with time



CentraleSupélec

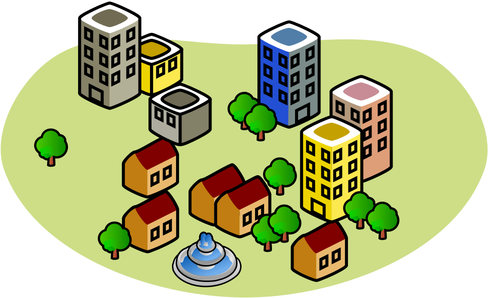
Context

Requirements evolve with time



Context

Requirements evolve with time



Context

Requirements evolve with time

CentraleSupélec

Context

Requirements evolve with time

Context

Requirements evolve with time

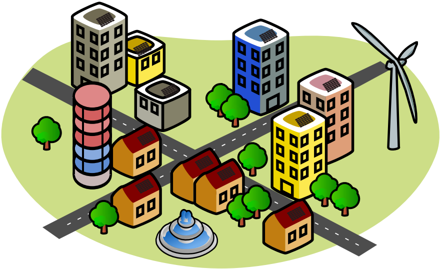
- Electricity Distribution System
- Heat distribution
- Water distribution
- Traffic management

(include your problem here)

CentraleSupélec

Context

Requirements evolve with time



- Electricity Distribution System
- Heat distribution
- Water distribution
- Traffic management

  (include your problem here)

CentraleSupélec

Context

Requirements evolve with time

- Electricity Distribution System
- Heat distribution
- Water distribution
- Traffic management

  (include your problem here)

Context

Requirements evolve with time

- Electricity Distribution System
- Heat distribution
- Water distribution
- Traffic management

  (include your problem here)

Context

Requirements evolve with time

- Multiple systems interacting
- Coupled by constraints
- Optimization objectives
  - Minimize energy consumption
  - Maximize user satisfaction
  - Follow a trajectory
  - ...
- Solution → MPC

CentraleSupélec

Context

Requirements evolve with time

- Multiple systems interacting
- Coupled by constraints
- Optimization objectives
  - Minimize energy consumption
  - Maximize user satisfaction
  - Follow a trajectory
  - ...
- Solution → MPC

Context

Requirements evolve with time

- Multiple systems interacting
- Coupled by constraints
- Optimization objectives
  - Minimize energy consumption
  - Maximize user satisfaction
  - Follow a trajectory
  - ...
- Solution → MPC

Context

Requirements evolve with time

- Multiple systems interacting
- Coupled by constraints
- Optimization objectives
  - Minimize energy consumption
  - Maximize user satisfaction
  - Follow a trajectory

  ...
- Solution → MPC

Context

Requirements evolve with time

- Multiple systems interacting
- Coupled by constraints
- Optimization objectives
  - Minimize energy consumption
  - Maximize user satisfaction
  - Follow a trajectory
  - ...
- Solution → MPC

CentraleSupélec

Context

Requirements evolve with time

- Multiple systems interacting
- Coupled by constraints
- Optimization objectives
  - Minimize energy consumption
  - Maximize user satisfaction
  - Follow a trajectory

  ...
- Solution → MPC

CentraleSupélec

Context

Requirements evolve with time

- Multiple systems interacting
- Coupled by constraints
- Optimization objectives
  - Minimize energy consumption
  - Maximize user satisfaction
  - Follow a trajectory

  ...
- Solution → MPC

Context

Requirements evolve with time

- Multiple systems interacting
- Coupled by constraints
- Optimization objectives
  - Minimize energy consumption
  - Maximize user satisfaction
  - Follow a trajectory

  ...
- Solution → MPC

CentraleSupélec

Model-based Predictive Control

Find best control sequence using predictions based on a model.

- Objective function to optimize
- System's Model (states and inputs)
- Other constraints to respect (QoS, technical restrictions, ...)

Model-based Predictive Control

Find best control sequence using predictions based on a model.

- Objective function to optimize
- System's Model (states and inputs)
- Other constraints to respect (QoS, technical restrictions, ...)

CentraleSupélec

Model-based Predictive Control

Find best control sequence using predictions based on a model.

- Objective function to optimize
- System's Model (states and inputs)
- Other constraints to respect (QoS, technical restrictions, ...)

CentraleSupélec

Model-based Predictive Control

Find optimal control sequence using predictions based on a model.

- Objective function to optimize
- System's Model (states and inputs)
- Other constraints to respect (QoS, technical restrictions, ...)

CentraleSupélec

Model-based Predictive Control

Find optimal control sequence using predictions based on a model.

- Objective function to optimize
- System's Model (states and inputs)
- Other constraints to respect (QoS, technical restrictions, ...)

$$
\begin{aligned}
\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad & J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k]) \\
\text{subject to} \quad & \left.\begin{aligned}
&\boldsymbol{x}[\xi|k] = f(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \\
&g_i(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \leq 0 \\
&h_j(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) = 0
\end{aligned}\right\}
\begin{aligned}
&\forall \xi \in \{1, \ldots, N\} \\
&\forall i \in \{1, \ldots, m\} \\
&\forall j \in \{1, \ldots, p\}
\end{aligned}
\end{aligned}
$$

Model-based Predictive Control

Find optimal control sequence using predictions based on a model.

- Objective function to optimize
- System's Model (states and inputs)
- Other constraints to respect (QoS, technical restrictions, ...)

$$
\begin{aligned}
\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad & J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k]) \\
\text{subject to} \quad & \left.\begin{aligned} \boldsymbol{x}[\xi|k] &= f(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \\ g_i(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) &\leq 0 \\ h_j(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) &= 0 \end{aligned}\right\} \begin{aligned} &\forall \xi \in \{1, \dots, N\} \\ &\forall i \in \{1, \dots, m\} \\ &\forall j \in \{1, \dots, p\} \end{aligned}
\end{aligned}
$$

CentraleSupélec

Model-based Predictive Control

Find optimal control sequence using predictions based on a model.

- Objective function to optimize
- System's Model (states and inputs)
- Other constraints to respect (QoS, technical restrictions, ...)

$$
\begin{aligned}
\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad & J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k]) \\
\text{subject to} \quad & \left.\begin{aligned} \boldsymbol{x}[\xi|k] &= f(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \\ g_i(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) &\leq 0 \\ h_j(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) &= 0 \end{aligned}\right\} \begin{aligned} &\forall \xi \in \{1, \dots, N\} \\ &\forall i \in \{1, \dots, m\} \\ &\forall j \in \{1, \dots, p\} \end{aligned}
\end{aligned}
$$

Model-based Predictive Control

Find optimal control sequence using predictions based on a model.

- Objective function to optimize
- System's Model (states and inputs)
- Other constraints to respect (QoS, technical restrictions, ...)

$$
\begin{aligned}
\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad & J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k]) \\
\text{subject to} \quad & \left.\begin{aligned}
\boldsymbol{x}[\xi|k] &= f(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \\
g_i(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) &\leq 0 \\
h_j(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) &= 0
\end{aligned}\right\} \begin{aligned} &\forall \xi \in \{1, \dots, N\} \\ &\forall i \in \{1, \dots, m\} \\ &\forall j \in \{1, \dots, p\} \end{aligned}
\end{aligned}
$$

CentraleSupélec

Model-based Predictive Control

Find optimal control sequence using predictions based on a model.

- Objective function to optimize
- System's Model (states and inputs)
- Other constraints to respect (QoS, technical restrictions, ...)

$$
\begin{aligned}
\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad & J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k]) \\
\text{subject to} \quad & \left.\begin{aligned}
\boldsymbol{x}[\xi|k] &= f(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \\
g_i(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) &\leqslant 0 \\
h_j(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) &= 0
\end{aligned}\right\} \begin{aligned}
&\forall \xi \in \{1, \dots, N\} \\
&\forall i \in \{1, \dots, m\} \\
&\forall j \in \{1, \dots, p\}
\end{aligned}
\end{aligned}
$$

CentraleSupélec

Model-based Predictive Control

Find optimal control sequence using predictions based on a model.

- Objective function to optimize
- System's Model (states and inputs)
- Other constraints to respect (QoS, technical restrictions, ...)

$$
\begin{aligned}
\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad & J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k]) \\
\text{subject to} \quad & \left.\begin{aligned} \boldsymbol{x}[\xi|k] &= f(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \\ g_i(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) &\leqslant 0 \\ h_j(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) &= 0 \end{aligned}\right\} \begin{aligned} &\forall \xi \in \{1, \dots, N\} \\ &\forall i \in \{1, \dots, m\} \\ &\forall j \in \{1, \dots, p\} \end{aligned}
\end{aligned}
$$

CentraleSupélec

Model-based Predictive Control

Find optimal control sequence using predictions based on a model.

- Objective function to optimize
- System's Model (states and inputs)
- Other constraints to respect (QoS, technical restrictions, ...)

$$
\begin{aligned}
\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad & J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k]) \\
\text{subject to} \quad & \left.\begin{aligned}
\boldsymbol{x}[\xi|k] &= f(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \\
g_i(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) &\leqslant 0 \\
h_j(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) &= 0
\end{aligned}\right\}
\begin{aligned}
&\forall \xi \in \{1, \dots, N\} \\
&\forall i \in \{1, \dots, m\} \\
&\forall j \in \{1, \dots, p\}
\end{aligned}
\end{aligned}
$$

Model-based Predictive Control

Find optimal control sequence using predictions based on a model.

- Objective function to optimize
- System's Model (states and inputs)
- Other constraints to respect (QoS, technical restrictions, ...)

$$
\begin{aligned}
\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad & J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k]) \\
\text{subject to} \quad & \left.\begin{aligned}
\boldsymbol{x}[\xi|k] &= f(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \\
g_i(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) &\leqslant 0 \\
h_j(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) &= 0
\end{aligned}\right\}
\begin{aligned}
&\forall \xi \in \{1, \dots, N\} \\
&\forall i \in \{1, \dots, m\} \\
&\forall j \in \{1, \dots, p\}
\end{aligned}
\end{aligned}
$$

CentraleSupélec

Model-based Predictive Control

Find optimal control sequence using predictions based on a model.

- Objective function to optimize
- System's Model (states and inputs)
- Other constraints to respect (QoS, technical restrictions, ...)

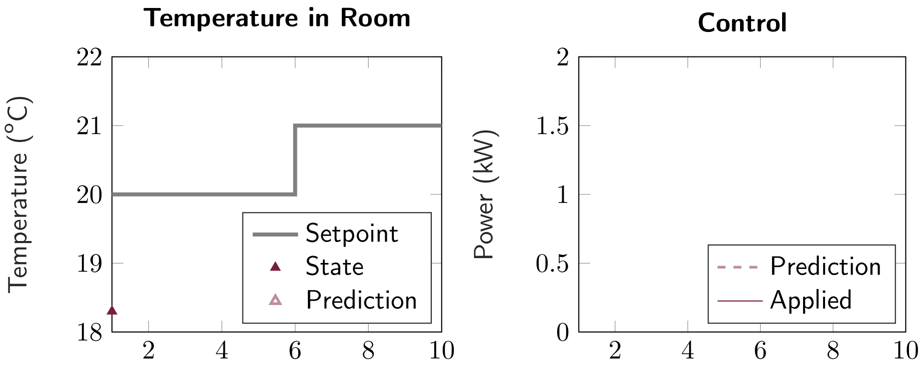
$$
\begin{aligned}
\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad & J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k]) \\
\text{subject to} \quad & \left.\begin{aligned}
\boldsymbol{x}[\xi|k] = f(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \\
g_i(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \leqslant 0 \\
h_j(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) = 0
\end{aligned}\right\} \begin{aligned}
&\forall \xi \in \{1, \dots, N\} \\
&\forall i \in \{1, \dots, m\} \\
&\forall j \in \{1, \dots, p\}
\end{aligned}
\end{aligned}
$$

CentraleSupélec

Model Predictive Control

In a nutshell

Find optimal control sequence

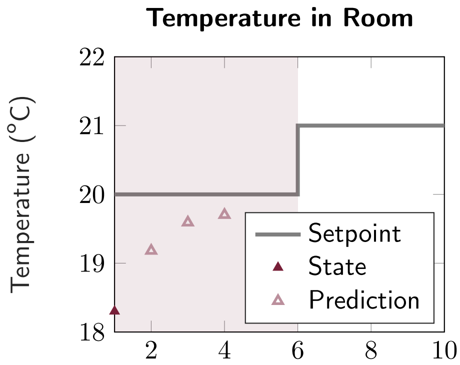


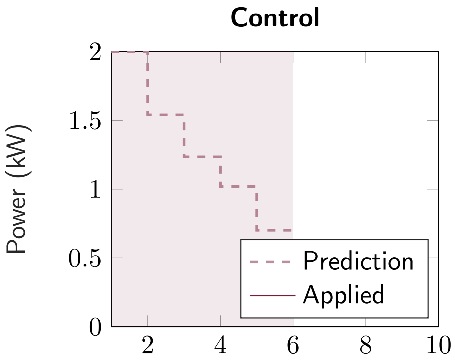
CentraleSupélec

Model Predictive Control

In a nutshell

Find optimal control sequence

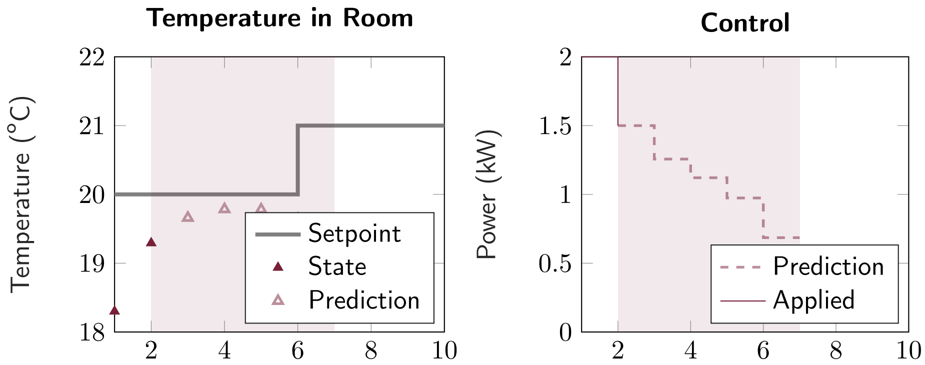




Model Predictive Control

In a nutshell

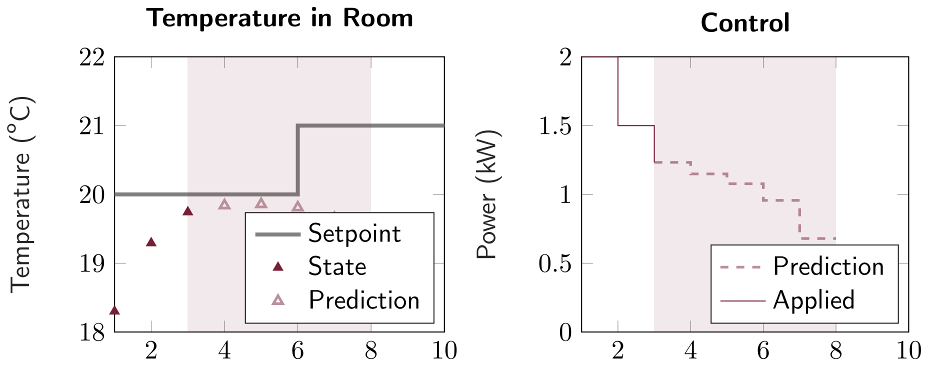
Find optimal control sequence, apply first element



Model Predictive Control

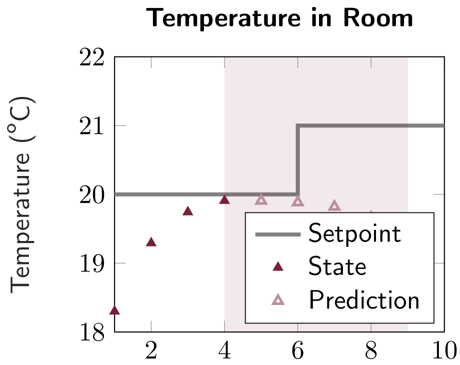
In a nutshell

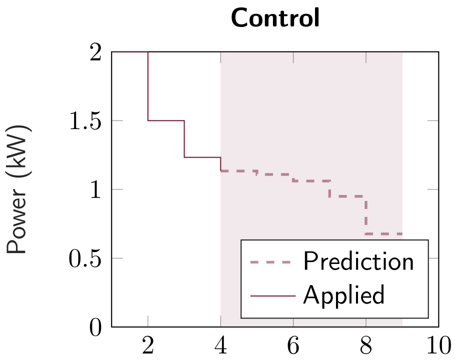
Find optimal control sequence, apply first element, rinse repeat



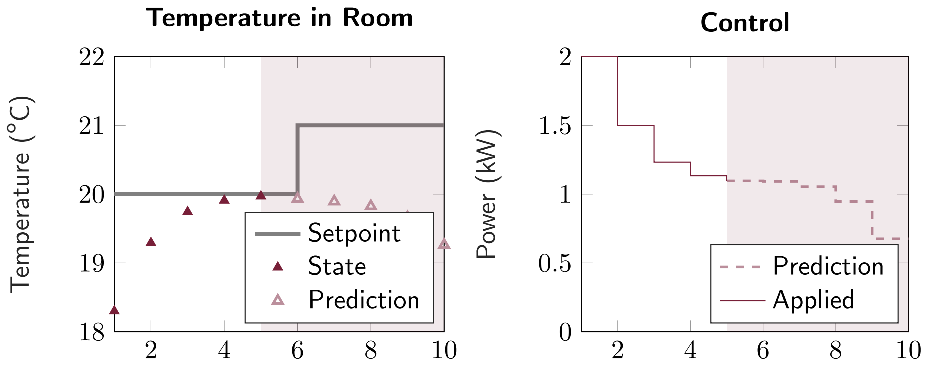
In a nutshell

Find optimal control sequence, apply first element, rinse repeat → Receding Horizon

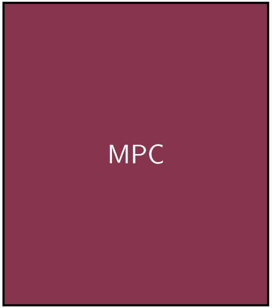




Find optimal control sequence, apply first element, rinse repeat → Receding Horizon



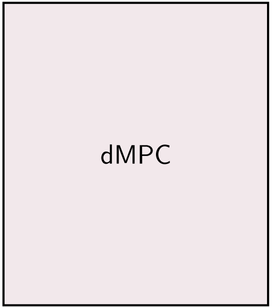
- Problem: Complexity depends on N, m, p and sizes of $\boldsymbol{x}$ and $\boldsymbol{u}$
- Solution: Divide and Conquer[1]



1

Distributed Model Predictive Control

- Problem: Complexity depends on N, m, p and sizes of $\boldsymbol{x}$ and $\boldsymbol{u}$
- Solution: Divide and Conquer[1]

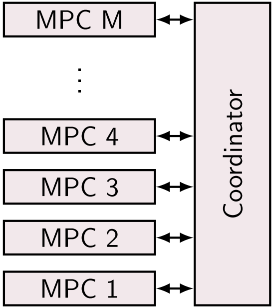


[1] *Distributed Model Predictive Control made easy*

CentraleSupélec

Distributed Model Predictive Control

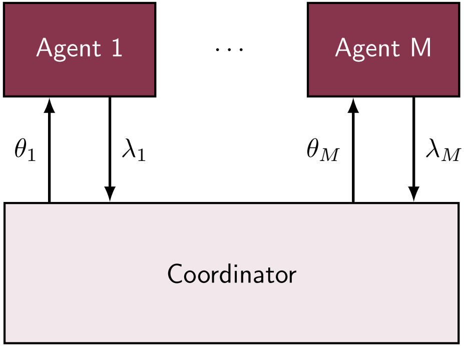
- Problem: Complexity depends on N, m, p and sizes of $\boldsymbol{x}$ and $\boldsymbol{u}$
- Solution: Divide and Conquer[1]



[1] *Distributed Model Predictive Control made easy*

Distributed Model Predictive Control

Optimization Frameworks

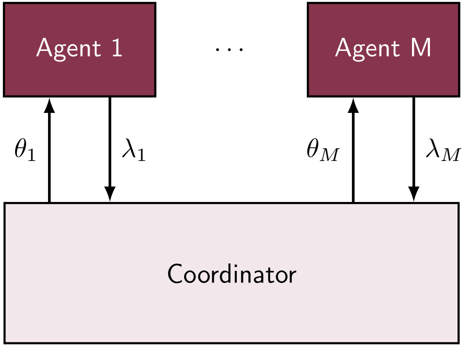


- Agents solve local problems
- Variables are updated

$\left.\right\}$ Until Convergence

Distributed Model Predictive Control

Optimization Frameworks

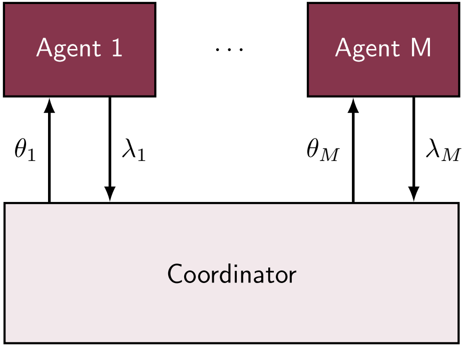


- Agents solve local problems
- Variables are updated

Until Convergence

Distributed Model Predictive Control

Optimization Frameworks



- Agents solve local problems
- Variables are updated

$\Big\}$ Until Convergence

Distributed Model Predictive Control

Negotiation works if agents comply.

But what if some agents are ill-intentioned and attack the system?

- What are the consequences of an attack?
- Can we mitigate the effects?

Distributed Model Predictive Control

Negotiation works if agents comply.
But what if some agents are ill-intentioned and attack the system?

- What are the consequences of an attack?
- Can we mitigate the effects?

CentraleSupélec

Distributed Model Predictive Control

Negotiation works if agents comply.
But what if some agents are ill-intentioned and attack the system?

- What are the consequences of an attack?
- Can we mitigate the effects?

CentraleSupélec

Distributed Model Predictive Control

Negotiation works if agents comply.
But what if some agents are ill-intentioned and attack the system?

- What are the consequences of an attack?
- Can we mitigate the effects?

CentraleSupélec

State of art

	Decomposition	Present vulnerabilities?	Resilient/Robust	Detection	Mitigation
[Vel+17a] [Mae+21]	Dual	Yes	Robust (Scenario)	NA	NA
[Vel+17b] [Vel+18]	Dual	Yes	Robust (f-robust)	NA	NA
[CMI18]	Jacobi-Gauß	Yes	–	–	–
[Ana+18] [Ana+19] [Ana+20]	Dual	Yes	Resilient	Analyt./Learn.	Disconnect (Robustness)
Our	Primal	Yes	Resilient	Active Analyt./Learn.	Data reconstruction

CentraleSupélec

State of art

	Decomposition	Present vulnerabilities?	Resilient/Robust	Detection	Mitigation
[Vel+17a] [Mae+21]	Dual	Yes	Robust (Scenario)	NA	NA
[Vel+17b] [Vel+18]	Dual	Yes	Robust (f-robust)	NA	NA
[CMI18]	Jacobi-Gauß	Yes	–	–	–
[Ana+18] [Ana+19] [Ana+20]	Dual	Yes	Resilient	Analyt./Learn.	Disconnect (Robustness)
Our	Primal	Yes	Resilient	Active Analyt./Learn.	Data reconstruction

CentraleSupélec

Outline

CentraleSupélec

Outline

1. Vulnerabilities in distributed MPC based on Primal Decomposition
2. Resilient Primal Decomposition-based dMPC for deprived systems
3. Resilient Primal Decomposition-based dMPC using Artificial Scarcity

CentraleSupélec

Outline

1. Vulnerabilities in distributed MPC based on Primal Decomposition
2. Resilient Primal Decomposition-based dMPC for deprived systems
3. Resilient Primal Decomposition-based dMPC using Artificial Scarcity

CentraleSupélec

Outline

CentraleSupélec

Distributed Model Predictive Control

Primal Decomposition | Quantity Decomposition | Resource Allocation

Decompose original problem using primal problem

Distributed Model Predictive Control

Primal Decomposition | Quantity Decomposition | Resource Allocation

Decompose original problem using primal problem

$$
\begin{aligned}
\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad & J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k]) \\
\text{subject to} \quad & \left.\begin{aligned}
\boldsymbol{x}[\xi|k] = f(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \\
g_i(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \leqslant 0 \\
h_j(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) = 0
\end{aligned}\right\}
\begin{aligned}
&\forall \xi \in \{1, \dots, N\} \\
&\forall i \in \{1, \dots, m\} \\
&\forall j \in \{1, \dots, p\}
\end{aligned}
\end{aligned}
$$

CentraleSupélec

Distributed Model Predictive Control

Primal Decomposition | Quantity Decomposition | Resource Allocation

Decompose original problem using primal problem

$$
\begin{aligned}
\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad & \sum_{i\in\mathcal{M}}\sum_{\xi\in\mathcal{N}}\left[\|\boldsymbol{v}_i[\xi|k]\|^2_{Q_i}+\|\boldsymbol{u}_i[\xi-1|k]\|^2_{R_i}\right] \\
\text{subject to} \quad & \left.\begin{aligned} \boldsymbol{x}[\xi|k] &= f(\boldsymbol{x}[\xi-1|k],\boldsymbol{u}[\xi-1|k]) \\ g_i(\boldsymbol{x}[\xi-1|k],\boldsymbol{u}[\xi-1|k]) &\leqslant 0 \\ h_j(\boldsymbol{x}[\xi-1|k],\boldsymbol{u}[\xi-1|k]) &= 0 \end{aligned}\right\} \begin{aligned} &\forall\xi\in\{1,\dots,N\} \\ &\forall i\in\{1,\dots,m\} \\ &\forall j\in\{1,\dots,p\} \end{aligned}
\end{aligned}
$$

CentraleSupélec

Distributed Model Predictive Control

Primal Decomposition | Quantity Decomposition | Resource Allocation

E.g. $v_i = w_i - x_i$

Decompose original problem using primal problem

$$\begin{aligned}
\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad & \sum_{i\in\mathcal{M}} \sum_{\xi\in\mathcal{N}} \left[\|\boldsymbol{v}_i[\xi|k]\|^2_{Q_i} + \|\boldsymbol{u}_i[\xi-1|k]\|^2_{R_i} \right] \\
\text{subject to} \quad & \left.\begin{aligned} \boldsymbol{x}[\xi|k] &= f(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \\ g_i(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) &\leqslant 0 \\ h_j(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) &= 0 \end{aligned}\right\} \begin{aligned} &\forall \xi \in \{1,\dots,N\} \\ &\forall i \in \{1,\dots,m\} \\ &\forall j \in \{1,\dots,p\} \end{aligned}
\end{aligned}$$

Distributed Model Predictive Control

Decompose original problem using primal problem

$$\begin{aligned}
\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad & \sum_{i\in\mathcal{M}} \sum_{\xi\in\mathcal{N}} \left[\|\boldsymbol{v}_i[\xi|k]\|^2_{Q_i} + \|\boldsymbol{u}_i[\xi-1|k]\|^2_{R_i} \right] \\
\text{subject to} \quad & \left.\begin{array}{c} \boldsymbol{x}[\xi|k] = A\boldsymbol{x}[\xi-1|k] + B\boldsymbol{u}[\xi-1|k] \\ g_i(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \leqslant 0 \\ h_j(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) = 0 \end{array}\right\} \begin{array}{l} \forall \xi \in \{1,\dots,N\} \\ \forall i \in \{1,\dots,m\} \\ \forall j \in \{1,\dots,p\} \end{array}
\end{aligned}$$

Distributed Model Predictive Control

Primal Decomposition | Quantity Decomposition | Resource Allocation

Decompose original problem using primal problem

$$\begin{aligned}
\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad & \sum_{i\in\mathcal{M}}\sum_{\xi\in\mathcal{N}}\left[\|\boldsymbol{v}_i[\xi|k]\|^2_{Q_i}+\|\boldsymbol{u}_i[\xi-1|k]\|^2_{R_i}\right] \\
\text{subject to} \quad & \left.\begin{aligned}\boldsymbol{x}[\xi|k] = A_i\boldsymbol{x}[\xi-1|k]+B_i\boldsymbol{u}[\xi-1|k] \\ \sum_{i\in\mathcal{M}}\Gamma_i\boldsymbol{u}_i[\xi|k]\leqslant\boldsymbol{u}_{\max}\end{aligned}\right\} \forall\xi\in\mathcal{N}
\end{aligned}$$

CentraleSupélec

Distributed Model Predictive Control

Primal Decomposition | Quantity Decomposition | Resource Allocation

Decompose original problem using primal problem

$$\begin{aligned} \underset{\boldsymbol{U}_1[k],\dots,\boldsymbol{U}_M[k]}{\text{minimize}} \quad & \sum_{i\in\mathcal{M}} \left[\frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|^2_{H_i} + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k]\right] \\ \text{subject to} \quad & \sum_{i\in\mathcal{M}} \left[\bar{\Gamma}_i\boldsymbol{U}_i[k]\right] \preceq \boldsymbol{U}_{\mathsf{max}} \end{aligned}$$

Distributed Model Predictive Control

Decompose original problem using primal problem

$$\begin{aligned}
&\underset{\boldsymbol{U}_1[k]}{\text{minimize}} && \frac{1}{2}\left\|\boldsymbol{U}_1[k]\right\|_{H_1}^2 + \boldsymbol{f}_1[k]^T\boldsymbol{U}_1[k] \\
&\text{subject to} && \bar{\Gamma}_1\boldsymbol{U}_1[k] \preceq \boldsymbol{\theta}_1[k] : \boldsymbol{\lambda}_1[k] \\
& && \vdots \\
&\underset{\boldsymbol{U}_M[k]}{\text{minimize}} && \frac{1}{2}\left\|\boldsymbol{U}_M[k]\right\|_{H_M}^2 + \boldsymbol{f}_M[k]^T\boldsymbol{U}_M[k] \\
&\text{subject to} && \bar{\Gamma}_M\boldsymbol{U}_M[k] \preceq \boldsymbol{\theta}_M[k] : \boldsymbol{\lambda}_M[k]
\end{aligned}$$

$$\boldsymbol{\theta}[k]^{(p+1)} = \mathrm{Proj}^{\mathcal{S}}(\boldsymbol{\theta}[k]^{(p)} + \rho^{(p)}\boldsymbol{\lambda}[k]^{(p)})$$

CentraleSupélec

Distributed Model Predictive Control

Decompose original problem using primal problem

$$\mathcal{S} = \{\boldsymbol{\theta}[k] \mid I_c^M \boldsymbol{\theta}[k] \preceq \boldsymbol{U}_{\max}\}$$

$$\begin{aligned} \underset{\boldsymbol{U}_1[k]}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{U}_1[k]\|_{H_1}^2 + \boldsymbol{f}_1[k]^T\boldsymbol{U}_1[k] \\ \text{subject to} \quad & \bar{\Gamma}_1\boldsymbol{U}_1[k] \preceq \boldsymbol{\theta}_1[k] : \boldsymbol{\lambda}_1[k] \\ & \vdots \\ \underset{\boldsymbol{U}_M[k]}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{U}_M[k]\|_{H_M}^2 + \boldsymbol{f}_M[k]^T\boldsymbol{U}_M[k] \\ \text{subject to} \quad & \bar{\Gamma}_M\boldsymbol{U}_M[k] \preceq \boldsymbol{\theta}_M[k] : \boldsymbol{\lambda}_M[k] \end{aligned}$$

$$\boldsymbol{\theta}[k]^{(p+1)} = \mathrm{Proj}^{\mathcal{S}}(\boldsymbol{\theta}[k]^{(p)} + \rho^{(p)}\boldsymbol{\lambda}[k]^{(p)})$$

Distributed Model Predictive Control

Decompose original problem using primal problem

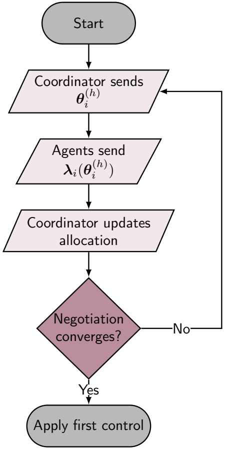
$$\mathcal{S} = \{\boldsymbol{\theta}[k] \mid I_c^M \boldsymbol{\theta}[k] \preceq \boldsymbol{U}_{\max}\}$$

$$\begin{array}{ll} \underset{\boldsymbol{U}_1[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_1[k]\right\|_{H_1}^2 + \boldsymbol{f}_1[k]^T \boldsymbol{U}_1[k] \\ \text{subject to} & \bar{\Gamma}_1 \boldsymbol{U}_1[k] \preceq \boldsymbol{\theta}_1[k] : \boldsymbol{\lambda}_1[k] \\ & \vdots \\ \underset{\boldsymbol{U}_M[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_M[k]\right\|_{H_M}^2 + \boldsymbol{f}_M[k]^T \boldsymbol{U}_M[k] \\ \text{subject to} & \bar{\Gamma}_M \boldsymbol{U}_M[k] \preceq \boldsymbol{\theta}_M[k] : \boldsymbol{\lambda}_M[k] \end{array}$$

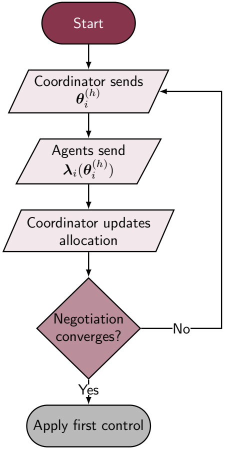
$$\boldsymbol{\theta}[k]^{(p+1)} = \mathrm{Proj}^{\mathcal{S}}(\boldsymbol{\theta}[k]^{(p)} + \rho^{(p)} \boldsymbol{\lambda}[k]^{(p)})$$

CentraleSupélec

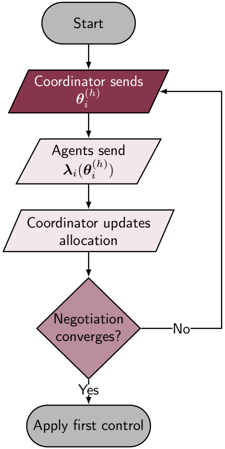
Quantity Decomposition | Resource Allocation



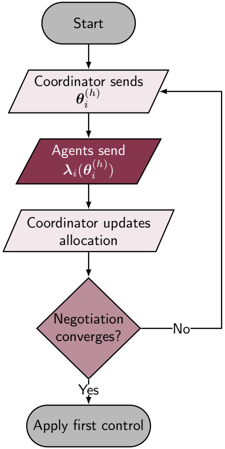
Quantity Decomposition | Resource Allocation



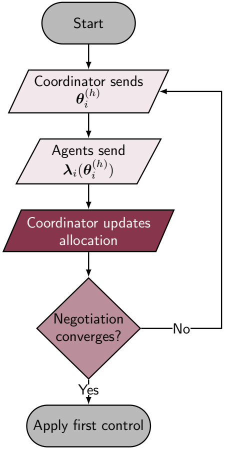
Quantity Decomposition | Resource Allocation



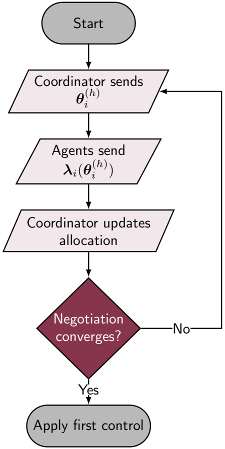
Quantity Decomposition | Resource Allocation



Quantity Decomposition | Resource Allocation

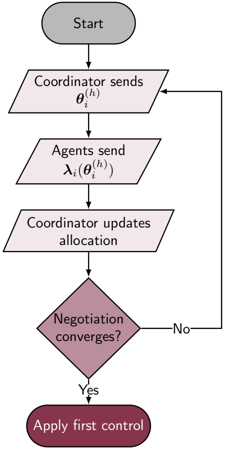


Quantity Decomposition | Resource Allocation



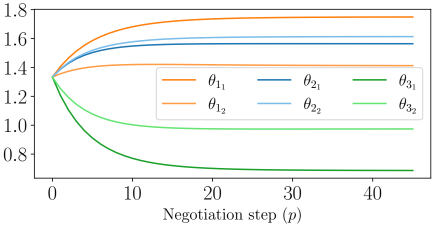
CentraleSupélec

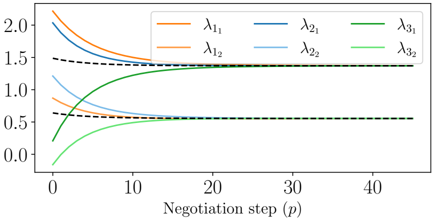
Quantity Decomposition | Resource Allocation



CentraleSupélec

Quantity Decomposition | Resource Allocation





How can a non-cooperative agent attack?

Literature

- [Vel+17a; CMI18] present some kinds of attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent (use different control)

Deception Attacks (False Data Injection)

How can a non-cooperative agent attack?

Literature

- [Vel+17a; CMI18] present some kinds of attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent (use different control)

Deception Attacks (False Data Injection)

CentraleSupélec

How can a non-cooperative agent attack?

Literature

- [Vel+17a; CMI18] present some kinds of attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent (use different control)

Deception Attacks (False Data Injection)

How can a non-cooperative agent attack?

Literature

- [Vel+17a; CMI18] present some kinds of attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent (use different control)

Deception Attacks (False Data Injection)

How can a non-cooperative agent attack?

Literature

- [Vel+17a; CMI18] present some kinds of attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent (use different control)

Deception Attacks (False Data Injection)

How can a non-cooperative agent attack?

Literature

- [Vel+17a; CMI18] present some kinds of attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent (use different control)

Deception Attacks
(False Data Injection)

CentraleSupélec

How can a non-cooperative agent attack?

Literature

- [Vel+17a; CMI18] present some kinds of attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent (use different control)

Deception Attacks
(False Data Injection)

How can a non-cooperative agent attack?

Literature

- [Vel+17a; CMI18] present some kinds of attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent (use different control)

Deception Attacks
(False Data Injection)

CentraleSupélec

How can a non-cooperative agent attack?

Our approach

- $\boldsymbol{\lambda}_i$ is the only interface with coordination
- $\boldsymbol{\lambda}_i$ depends on the parameters of the system
- Malicious agent sends a different $\boldsymbol{\lambda}_i$

CentraleSupélec

How can a non-cooperative agent attack?

Our approach

- $\boldsymbol{\lambda}_i$ is the only interface with coordination
- $\boldsymbol{\lambda}_i$ depends on the parameters of the system
- Malicious agent sends a different $\boldsymbol{\lambda}_i$

How can a non-cooperative agent attack?

Our approach

- $\boldsymbol{\lambda}_i$ is the only interface with coordination
- $\boldsymbol{\lambda}_i$ depends on the parameters of the system
- Malicious agent sends a different $\boldsymbol{\lambda}_i$

How can a non-cooperative agent attack?

Our approach

- $\boldsymbol{\lambda}_i$ is the only interface with coordination
- $\boldsymbol{\lambda}_i$ depends on the parameters of the system
- Malicious agent sends a different $\boldsymbol{\lambda}_i$

$$\tilde{\boldsymbol{\lambda}}_i = \gamma_i(\boldsymbol{\lambda}_i)$$

Example

Let's suppose $\gamma_i(\boldsymbol{\lambda}_i) = T_i\boldsymbol{\lambda}_i$

Example

Let's suppose $\gamma_i(\boldsymbol{\lambda}_i) = T_i\boldsymbol{\lambda}_i$

4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$

Example

Let's suppose $\gamma_i(\boldsymbol{\lambda}_i) = T_i\boldsymbol{\lambda}_i$

4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$

CentraleSupélec

Example

Let's suppose $\gamma_i(\boldsymbol{\lambda}_i) = T_i\boldsymbol{\lambda}_i$

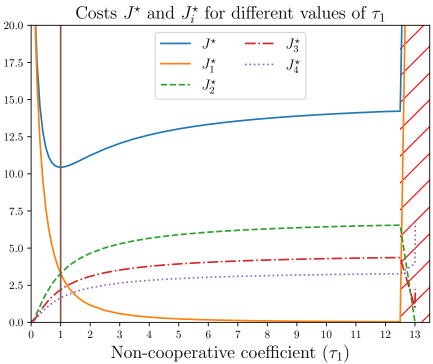
4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I\boldsymbol{\lambda}_1$

CentraleSupélec

Example

Let's suppose $\gamma_i(\boldsymbol{\lambda}_i) = T_i\boldsymbol{\lambda}_i$

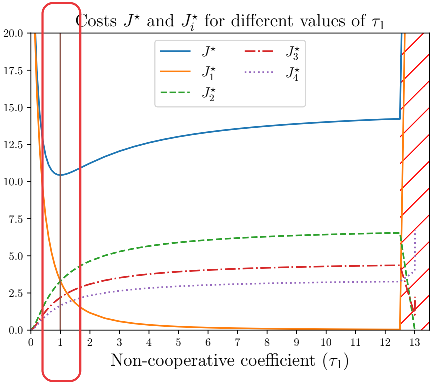


4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$

Example

Let's suppose $\gamma_i(\boldsymbol{\lambda}_i) = T_i\boldsymbol{\lambda}_i$



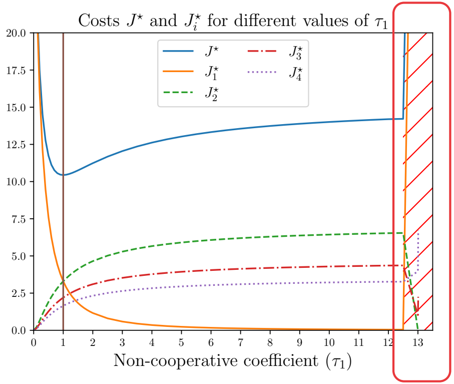
4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$

CentraleSupélec

Example

Let's suppose $\gamma_i(\boldsymbol{\lambda}_i) = T_i\boldsymbol{\lambda}_i$



4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$

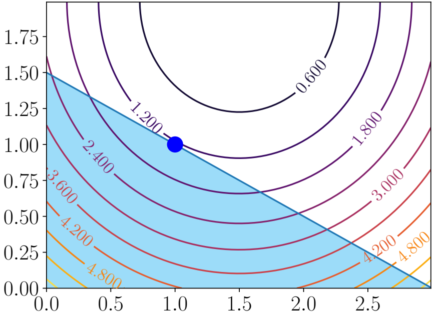
CentraleSupélec

Mitigating

There are vulnerabilities

Mitigating

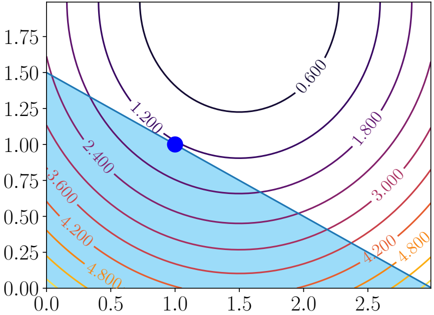
There are vulnerabilities



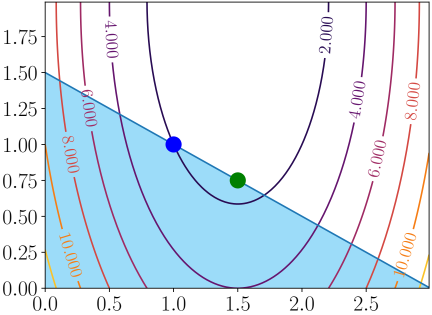
Original minimum.

Mitigating

There are vulnerabilities



Original minimum.



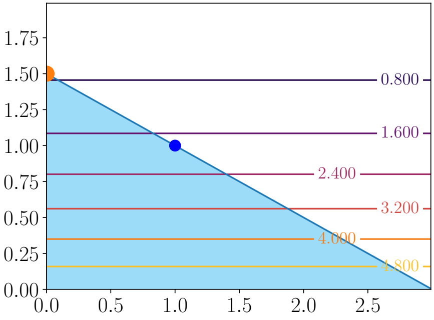
Minimum after attack.

Mitigating

There are vulnerabilities
"Great". But what can we do?

Mitigating

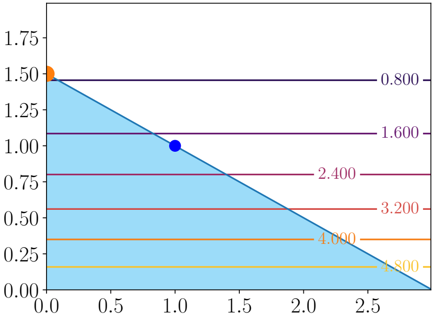
There are vulnerabilities
"Great". But what can we do?



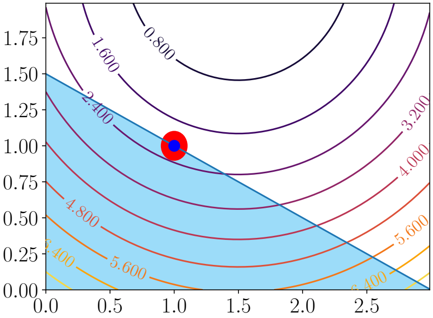
Ignore attacker.

Mitigating

There are vulnerabilities
"Great". But what can we do?



Ignore attacker.

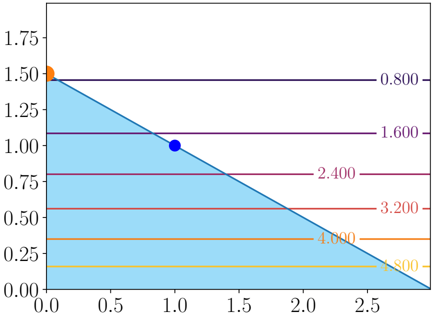


Recover original behavior.

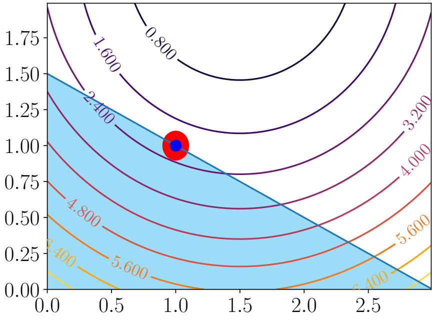
CentraleSupélec

Mitigating

There are vulnerabilities
"Great". But what can we do?



Ignore attacker.



Recover original behavior.

Strategy

- Recover original behavior
  - Invert effects of function $\gamma_i(\boldsymbol{\lambda}_i)$
- Is $\gamma_i(\boldsymbol{\lambda}_i)$ invertible?
  - Not necessarily, but let's reason

Strategy

- Recover original behavior
  - Invert effects of function $\gamma_i(\boldsymbol{\lambda}_i)$
- Is $\gamma_i(\boldsymbol{\lambda}_i)$ invertible?
  - Not necessarily, but let's reason

Strategy

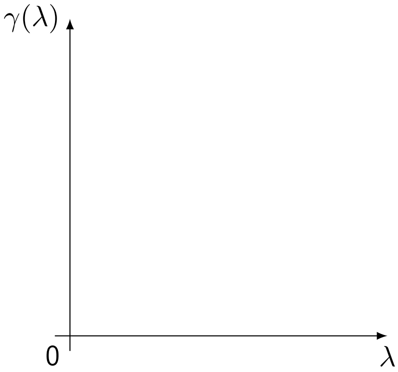
- Recover original behavior
  - Invert effects of function $\gamma_i(\boldsymbol{\lambda}_i)$
- Is $\gamma_i(\boldsymbol{\lambda}_i)$ invertible?
  - Not necessarily, but let's reason

Strategy

- Recover original behavior
  - Invert effects of function $\gamma_i(\boldsymbol{\lambda}_i)$
- Is $\gamma_i(\boldsymbol{\lambda}_i)$ invertible?
  - Not necessarily, but let's reason

Is $\gamma_i(\boldsymbol{\lambda}_i)$ invertible?

Unidimensional Case



- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction
  - $\gamma(\lambda) = 0 \Leftrightarrow \lambda = 0$

Assumptions

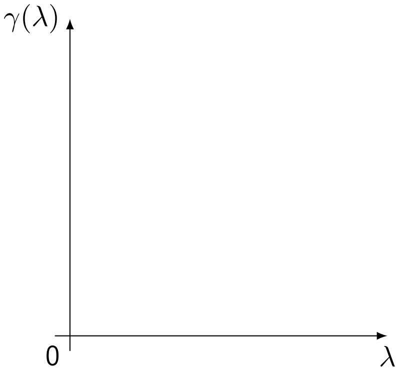
- Attacker is greedy $\gamma(\lambda) > \lambda$
- But not smart

  $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

CentraleSupélec

Is $\gamma_i(\boldsymbol{\lambda}_i)$ invertible?

Unidimensional Case



- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction
  - $\gamma(\lambda) = 0 \Leftrightarrow \lambda = 0$

Assumptions

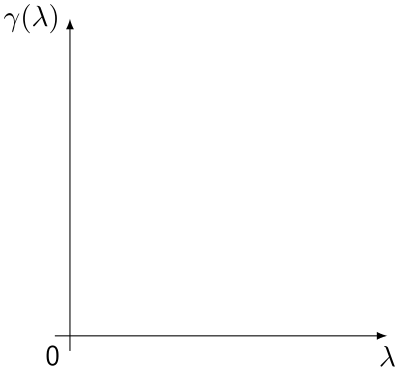
- Attacker is greedy $\gamma(\lambda) > \lambda$
- But not smart

$\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

CentraleSupélec

Is $\gamma_i(\boldsymbol{\lambda}_i)$ invertible?

Unidimensional Case



- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction
  - $\gamma(\lambda) = 0 \Leftrightarrow \lambda = 0$

Assumptions

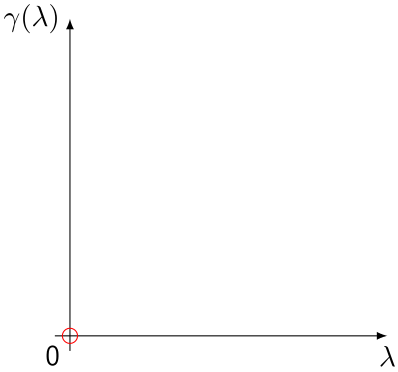
- Attacker is greedy $\gamma(\lambda) > \lambda$
- But not smart

$\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

CentraleSupélec

Is $\gamma_i(\boldsymbol{\lambda}_i)$ invertible?

Unidimensional Case



- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction
  - $\gamma(\lambda) = 0 \Leftrightarrow \lambda = 0$

Assumptions

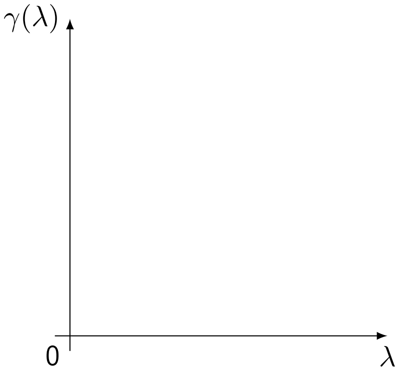
- Attacker is greedy $\gamma(\lambda) > \lambda$
- But not smart

  $\lambda_b > \lambda_a \to \gamma(\lambda_b) > \gamma(\lambda_a)$

CentraleSupélec

Is $\gamma_i(\boldsymbol{\lambda}_i)$ invertible?

Unidimensional Case



- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction
  - $\gamma(\lambda) = 0 \Leftrightarrow \lambda = 0$

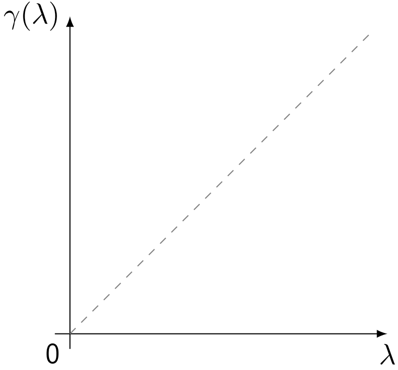
Assumptions

- Attacker is greedy $\gamma(\lambda) > \lambda$
- But not smart

$\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

Is $\gamma_i(\boldsymbol{\lambda}_i)$ invertible?

Unidimensional Case



- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction
  - $\gamma(\lambda) = 0 \Leftrightarrow \lambda = 0$

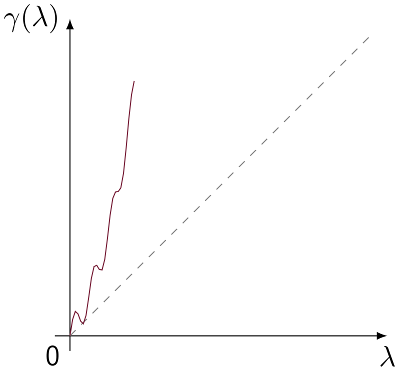
Assumptions

- Attacker is greedy $\gamma(\lambda) > \lambda$
- But not smart

  $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

Is $\gamma_i(\boldsymbol{\lambda}_i)$ invertible?

Unidimensional Case



- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction
  - $\gamma(\lambda) = 0 \Leftrightarrow \lambda = 0$

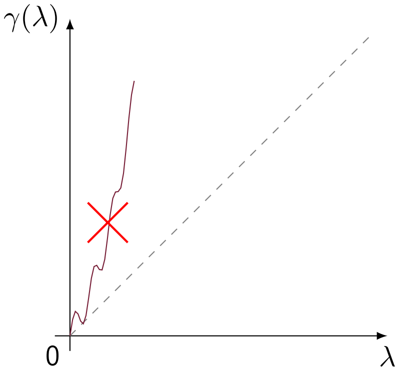
Assumptions

- Attacker is greedy $\gamma(\lambda) > \lambda$
- But not smart

  $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

Is $\gamma_i(\boldsymbol{\lambda}_i)$ invertible?

Unidimensional Case



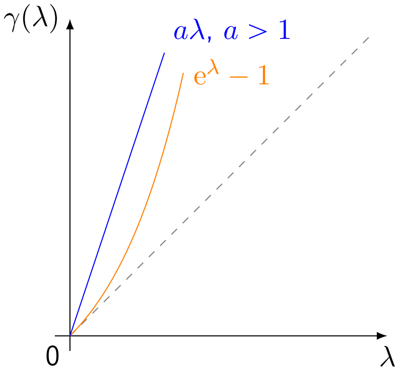
- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction
  - $\gamma(\lambda) = 0 \Leftrightarrow \lambda = 0$

Assumptions

- Attacker is greedy $\gamma(\lambda) > \lambda$
- But not smart
  $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

Is $\gamma_i(\boldsymbol{\lambda}_i)$ invertible?

Unidimensional Case



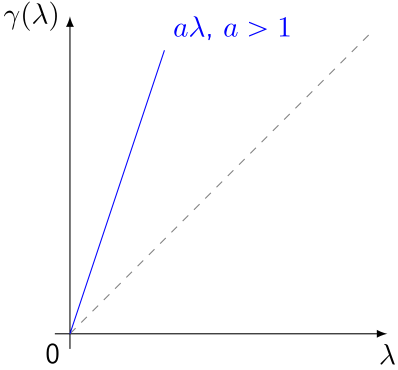
- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction
  - $\gamma(\lambda) = 0 \Leftrightarrow \lambda = 0$

Assumptions

- Attacker is greedy $\gamma(\lambda) > \lambda$
- But not smart
  $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

Is $\gamma_i(\boldsymbol{\lambda}_i)$ invertible?

Unidimensional Case



- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction
  - $\gamma(\lambda) = 0 \Leftrightarrow \lambda = 0$

Assumptions

- Attacker is greedy $\gamma(\lambda) > \lambda$
- But not smart
  $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

Is $\gamma_i(\boldsymbol{\lambda}_i)$ invertible?

Linear Multidimensional Case

- $\gamma(\lambda) = T\lambda$
- $T\lambda = 0 \Leftrightarrow \lambda = 0 \rightarrow \exists T^{-1}$
- Each agent could choose different T_i each time

Is $\gamma_i(\boldsymbol{\lambda}_i)$ invertible?

Linear Multidimensional Case

- $\gamma(\lambda) = T\lambda$
- $T\lambda = 0 \Leftrightarrow \lambda = 0 \rightarrow \exists T^{-1}$
- Each agent could choose different T_i each time

Is $\gamma_i(\boldsymbol{\lambda}_i)$ invertible?

Linear Multidimensional Case

- $\gamma(\lambda) = T\lambda$
- $T\lambda = 0 \Leftrightarrow \lambda = 0 \rightarrow \exists T^{-1}$
- Each agent could choose different T_i each time

CentraleSupélec

Is $\gamma_i(\boldsymbol{\lambda}_i)$ invertible?

Linear Multidimensional Case

- $\gamma(\lambda) = T\lambda$
- $T\lambda = 0 \Leftrightarrow \lambda = 0 \rightarrow \exists T^{-1}$
- Each agent could choose different T_i each time

CentraleSupélec

Is $\gamma_i(\boldsymbol{\lambda}_i)$ invertible?

Linear Multidimensional Case

- $\gamma(\lambda) = T\lambda$
- $T\lambda = 0 \Leftrightarrow \lambda = 0 \rightarrow \exists T^{-1}$
- Each agent could choose different T_i each time $\rightarrow T_i[k]$

Strategy

- Recover original behavior
  - Invert effects of function $\gamma_i(\boldsymbol{\lambda}_i)$ $\rightarrow$ Estimate $T_i[k]^{-1}$
- But how? Analyzing the system

Strategy

- Recover original behavior
  - Invert effects of function $\gamma_i(\boldsymbol{\lambda}_i) \rightarrow$ Estimate $T_i[k]^{-1}$
- But how? Analyzing the system

CentraleSupélec

Strategy

- Recover original behavior
  - Invert effects of function $\gamma_i(\boldsymbol{\lambda}_i) \rightarrow$ Estimate $T_i[k]^{-1}$
- But how? Analyzing the system

Strategy

- Recover original behavior
  - Invert effects of function $\gamma_i(\boldsymbol{\lambda}_i) \rightarrow$ Estimate $T_i[k]^{-1}$
- But how? Analyzing the system

CentraleSupélec

Outline

What are deprived systems?

$$\begin{aligned} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} \quad & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T \boldsymbol{U}_i[k] \\ \text{subject to} \quad & \bar{\Gamma}_i \boldsymbol{U}_i[k] \preceq \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{aligned}$$

- Unconstrained solution $\mathring{\boldsymbol{U}}_i^\star[k] = -H_i^{-1}\boldsymbol{f}_i[k]$
- Deprived if constraints are active $\rightarrow \bar{\Gamma}_i \mathring{\boldsymbol{U}}_i^\star[k] \succ \boldsymbol{\theta}_i[k], \forall k$
  - Solution projected onto boundaries (equality constraints)

What are deprived systems?

$$\begin{array}{ll} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} & \bar{\Gamma}_i\boldsymbol{U}_i[k] \preceq \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{array}$$

- Unconstrained solution $\mathring{\boldsymbol{U}}_i^\star[k] = -H_i^{-1}\boldsymbol{f}_i[k]$
- Deprived if constraints are active $\rightarrow \bar{\Gamma}_i\mathring{\boldsymbol{U}}_i^\star[k] \succ \boldsymbol{\theta}_i[k], \forall k$
  - Solution projected onto boundaries (equality constraints)

CentraleSupélec

What are deprived systems?

$$\begin{array}{ll} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T \boldsymbol{U}_i[k] \\ \text{subject to} & \bar{\Gamma}_i \boldsymbol{U}_i[k] \preceq \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{array}$$

- Unconstrained solution $\mathring{\boldsymbol{U}}_i^\star[k] = -H_i^{-1}\boldsymbol{f}_i[k]$
- Deprived if constraints are active $\rightarrow \bar{\Gamma}_i \mathring{\boldsymbol{U}}_i^\star[k] \succ \boldsymbol{\theta}_i[k], \forall k$
  - Solution projected onto boundaries (equality constraints)

What are deprived systems?

$$\begin{aligned} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} \quad & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} \quad & \bar{\Gamma}_i\boldsymbol{U}_i[k] \preceq \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{aligned}$$

- Unconstrained solution $\mathring{\boldsymbol{U}}_i^\star[k] = -H_i^{-1}\boldsymbol{f}_i[k]$
- Deprived if constraints are active $\rightarrow \bar{\Gamma}_i\mathring{\boldsymbol{U}}_i^\star[k] \succ \boldsymbol{\theta}_i[k], \forall k$
  - Solution projected onto boundaries (equality constraints)

CentraleSupélec

What are deprived systems?

$$\begin{array}{ll} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T \boldsymbol{U}_i[k] \\ \text{subject to} & \bar{\Gamma}_i \boldsymbol{U}_i[k] \preceq \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{array}$$

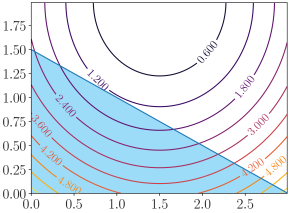
- Unconstrained solution $\mathring{\boldsymbol{U}}_i^\star[k] = -H_i^{-1}\boldsymbol{f}_i[k]$
- Deprived if constraints are active $\rightarrow \bar{\Gamma}_i \mathring{\boldsymbol{U}}_i^\star[k] \succ \boldsymbol{\theta}_i[k], \forall k$
    - Solution projected onto boundaries (equality constraints)

CentraleSupélec

What are deprived systems?

$$\begin{array}{ll} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} & \bar{\Gamma}_i\boldsymbol{U}_i[k] \preceq \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{array}$$

- Unconstrained solution $\mathring{\boldsymbol{U}}_i^\star[k] = -H_i^{-1}\boldsymbol{f}_i[k]$
- Deprived if constraints are active $\rightarrow \bar{\Gamma}_i\mathring{\boldsymbol{U}}_i^\star[k] \succ \boldsymbol{\theta}_i[k], \forall k$
  - Solution projected onto boundaries (equality constraints)

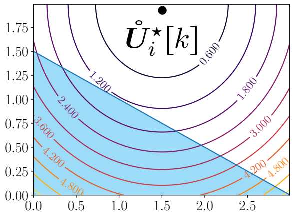


CentraleSupélec

What are deprived systems?

$$\begin{array}{ll} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\|\boldsymbol{U}_i[k]\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} & \bar{\Gamma}_i\boldsymbol{U}_i[k] \preceq \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{array}$$

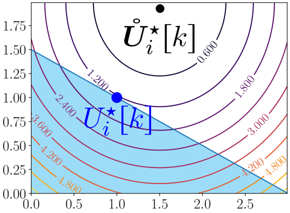
- Unconstrained solution $\mathring{\boldsymbol{U}}_i^\star[k] = -H_i^{-1}\boldsymbol{f}_i[k]$
- Deprived if constraints are active $\rightarrow \bar{\Gamma}_i\mathring{\boldsymbol{U}}_i^\star[k] \succ \boldsymbol{\theta}_i[k], \forall k$
  - Solution projected onto boundaries (equality constraints)



What are deprived systems?

$$\begin{array}{ll} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} & \bar{\Gamma}_i\boldsymbol{U}_i[k] \preceq \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{array}$$

- Unconstrained solution $\mathring{\boldsymbol{U}}_i^\star[k] = -H_i^{-1}\boldsymbol{f}_i[k]$
- Deprived if constraints are active $\rightarrow \bar{\Gamma}_i\mathring{\boldsymbol{U}}_i^\star[k] \succ \boldsymbol{\theta}_i[k], \forall k$
  - Solution projected onto boundaries (equality constraints)



CentraleSupélec

Deprived Systems

But why?

- No Scarcity → All constraints satisfied → $\lambda_i = 0$ → No coordination needed
- Scarcity → Competition → Consensus/Compromise (or cheating)

Deprived Systems

But why?

- No Scarcity → All constraints satisfied → $\lambda_i = 0$ → No coordination needed
- Scarcity → Competition → Consensus/Compromise (or cheating)

CentraleSupélec

Deprived Systems

But why?

- No Scarcity → All constraints satisfied $\rightarrow \lambda_i = 0 \rightarrow$ No coordination needed
- Scarcity → Competition → Consensus/Compromise (or cheating)

Deprived Systems

But why?

- No Scarcity → All constraints satisfied → $\lambda_i = 0$ → No coordination needed
- Scarcity → Competition → Consensus/Compromise (or cheating)

Deprived Systems

But why?

- No Scarcity → All constraints satisfied $\rightarrow \lambda_i = 0 \rightarrow$ No coordination needed
- Scarcity → Competition → Consensus/Compromise (or cheating)

Deprived Systems

But why?

- No Scarcity → All constraints satisfied $\rightarrow \lambda_i = 0 \rightarrow$ No coordination needed
- Scarcity → Competition → Consensus/Compromise (or cheating)

Deprived Systems

But why?

- No Scarcity $\rightarrow$ All constraints satisfied $\rightarrow \lambda_i = 0 \rightarrow$ No coordination needed
- Scarcity $\rightarrow$ Competition $\rightarrow$ Consensus/Compromise (or cheating)

Deprived Systems

But why?

- No Scarcity → All constraints satisfied $\rightarrow \lambda_i = 0 \rightarrow$ No coordination needed
- Scarcity → Competition → Consensus/Compromise (or cheating)

Deprived Systems

Analysis

- We can transform inequality constraints into equality ones[2]
- Solution is analytical and trivial.
  - If we solve for λ_i

$$
\begin{aligned}
\underset{\boldsymbol{U}_i[k]}{\text{minimize}} \quad & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\
\text{subject to} \quad & \bar{\Gamma}_i\boldsymbol{U}_i[k] \geqslant \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k]
\end{aligned}
$$

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k],$$

where $P_i = (\bar{\Gamma}_i H_i^{-1}\bar{\Gamma}_i^T)^{-1}$ and $\boldsymbol{s}_i[k] = P_i\bar{\Gamma}_i H_i^{-1}\boldsymbol{f}_i[k]$.

[2] Under some conditions

CentraleSupélec

Deprived Systems

Analysis

- We can transform inequality constraints into equality ones[2]
- Solution is analytical and trivial.
  - If we solve for λ_i

$$\begin{aligned} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} \quad & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} \quad & \bar{\Gamma}_i\boldsymbol{U}_i[k] = \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{aligned}$$

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k],$$

where $P_i = (\bar{\Gamma}_i H_i^{-1}\bar{\Gamma}_i^T)^{-1}$ and $\boldsymbol{s}_i[k] = P_i\bar{\Gamma}_i H_i^{-1}\boldsymbol{f}_i[k]$.

[2] Under some conditions see here

Deprived Systems

Analysis

- We can transform inequality constraints into equality ones[2]
- Solution is analytical and trivial.
  - If we solve for λ_i

$$\begin{aligned} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} \quad & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} \quad & \bar{\Gamma}_i\boldsymbol{U}_i[k] = \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{aligned}$$

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k],$$

where $P_i = \left(\bar{\Gamma}_i H_i^{-1}\bar{\Gamma}_i^T\right)^{-1}$ and $\boldsymbol{s}_i[k] = P_i\bar{\Gamma}_i H_i^{-1}\boldsymbol{f}_i[k]$.

[2] Under some conditions see here

Deprived Systems

Analysis

- We can transform inequality constraints into equality ones[2]
- Solution is analytical and trivial.
  - If we solve for λ_i

$$\begin{aligned} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} \quad & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T \boldsymbol{U}_i[k] \\ \text{subject to} \quad & \bar{\Gamma}_i \boldsymbol{U}_i[k] = \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{aligned}$$

$$\boldsymbol{\lambda}_i[k] = -P_i \boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k],$$

where $P_i = (\bar{\Gamma}_i H_i^{-1} \bar{\Gamma}_i^T)^{-1}$ and $\boldsymbol{s}_i[k] = P_i \bar{\Gamma}_i H_i^{-1} \boldsymbol{f}_i[k]$.

[2] Under some conditions see here

Deprived Systems

Analysis

- We can transform inequality constraints into equality ones[2]
- Solution is analytical and trivial.
  - If we solve for λ_i

$$\begin{aligned} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} \quad & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} \quad & \bar{\Gamma}_i\boldsymbol{U}_i[k] = \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{aligned}$$

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k],$$

where $P_i = (\bar{\Gamma}_i H_i^{-1} \bar{\Gamma}_i^T)^{-1}$ and $\boldsymbol{s}_i[k] = P_i\bar{\Gamma}_i H_i^{-1}\boldsymbol{f}_i[k]$.

[2]Under some conditions see here

CentraleSupélec

Deprived Systems

Analysis

- We can transform inequality constraints into equality ones[2]
- Solution is analytical and trivial.
  - If we solve for λ_i

$$\begin{array}{ll} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} & \bar{\Gamma}_i\boldsymbol{U}_i[k] = \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{array}$$

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k],$$

where $P_i = \left(\bar{\Gamma}_i H_i^{-1}\bar{\Gamma}_i{}^T\right)^{-1}$ and $\boldsymbol{s}_i[k] = P_i\bar{\Gamma}_i H_i^{-1}\boldsymbol{f}_i[k]$.

[2] Under some conditions see here

Deprived Systems

Analysis

- We can transform inequality constraints into equality ones[2]
- Solution is analytical and trivial.
  - If we solve for λ_i

$$\begin{aligned} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} \quad & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|^2_{H_i} + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} \quad & \bar{\Gamma}_i\boldsymbol{U}_i[k] = \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{aligned}$$

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k],$$

where $P_i = \left(\bar{\Gamma}_i H_i^{-1}\bar{\Gamma}_i^T\right)^{-1}$ and $\boldsymbol{s}_i[k] = P_i\bar{\Gamma}_i H_i^{-1}\boldsymbol{f}_i[k]$.

[2] Under some conditions see here

CentraleSupélec

Deprived Systems

Analysis

- We can transform inequality constraints into equality ones[2]
- Solution is analytical and trivial.
  - If we solve for λ_i

$$\begin{array}{ll}\underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2+\boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k]\\ \text{subject to} & \bar{\Gamma}_i\boldsymbol{U}_i[k]=\boldsymbol{\theta}_i[k]:\boldsymbol{\lambda}_i[k]\end{array}$$

$$\boldsymbol{\lambda}_i[k]=-P_i\boldsymbol{\theta}_i[k]-\boldsymbol{s}_i[k],$$

where $P_i=(\bar{\Gamma}_iH_i^{-1}\bar{\Gamma}_i^T)^{-1}$ and $\boldsymbol{s}_i[k]=P_i\bar{\Gamma}_iH_i^{-1}\boldsymbol{f}_i[k]$.

[2] Under some conditions see here

CentraleSupélec

Deprived Systems

Analysis (Continued)

$$\mathcal{S} = \{\boldsymbol{\theta}[k] \mid I_c^M \boldsymbol{\theta}[k] \leqslant \boldsymbol{U}_{\max}\}$$

$$\boldsymbol{\theta}[k]^{(p+1)} = \mathrm{Proj}^{\mathcal{S}}(\boldsymbol{\theta}[k]^{(p)} + \rho^{(p)}\boldsymbol{\lambda}[k]^{(p)})$$

CentraleSupélec

Deprived Systems

Analysis (Continued)

$$\mathcal{S} = \{\boldsymbol{\theta}[k] \mid I_c^M \boldsymbol{\theta}[k] = \boldsymbol{U}_{\max}\}$$

$$\boldsymbol{\theta}_i^{(p+1)} = \boldsymbol{\theta}_i^{(p)} + \rho^{(p)} \left(\boldsymbol{\lambda}_i^{(p)} - \frac{1}{M} \sum_{j=1}^{M} \boldsymbol{\lambda}_j^{(p)} \right), \forall i \in \mathcal{M}$$

Deprived Systems

Analysis (Continued)

$$\mathcal{S} = \{\boldsymbol{\theta}[k] \mid I_c^M \boldsymbol{\theta}[k] = \boldsymbol{U}_{\max}\}$$

$$\boldsymbol{\theta}_i^{(p+1)} = \boldsymbol{\theta}_i^{(p)} + \rho^{(p)} \left(\boldsymbol{\lambda}_i^{(p)} - \frac{1}{M} \sum_{j=1}^{M} \boldsymbol{\lambda}_j^{(p)} \right), \forall i \in \mathcal{M}$$

CentraleSupélec

Deprived Systems

Analysis (Continued)

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$$

$$\mathcal{S} = \{\boldsymbol{\theta}[k] \mid I_c^M\boldsymbol{\theta}[k] = \boldsymbol{U}_{\max}\}$$

$$\boldsymbol{\theta}_i^{(p+1)} = \boldsymbol{\theta}_i^{(p)} + \rho^{(p)}\left(\boldsymbol{\lambda}_i^{(p)} - \frac{1}{M}\sum_{j=1}^{M}\boldsymbol{\lambda}_j^{(p)}\right), \forall i \in \mathcal{M}$$

CentraleSupélec

Deprived Systems

Analysis (Continued)

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$$

$$\mathcal{S} = \{\boldsymbol{\theta}[k] \mid I_c^M\boldsymbol{\theta}[k] = \boldsymbol{U}_{\max}\}$$

$$\boldsymbol{\theta}_i^{(p+1)} = \boldsymbol{\theta}_i^{(p)} + \rho^{(p)}\left(\boldsymbol{\lambda}_i^{(p)} - \frac{1}{M}\sum_{j=1}^{M}\boldsymbol{\lambda}_j^{(p)}\right), \forall i \in \mathcal{M}$$

$$\boldsymbol{\theta}^{(p+1)} = \mathcal{A}_\theta\boldsymbol{\theta}^{(p)} + \mathcal{B}_\theta[k]$$ see here

CentraleSupélec

Deprived Systems

Under attack!

- Normal behavior

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k],$$

- Under attack

$$\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i = -T_iP_i\boldsymbol{\theta}_i[k] - T_i\boldsymbol{s}_i[k]$$

$$\tilde{\boldsymbol{\lambda}}_i = -\tilde{P}_i[k]\boldsymbol{\theta}_i[k] - \tilde{\boldsymbol{s}}_i[k]$$

Deprived Systems

Under attack!

- Normal behavior

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k],$$

- Under attack

$$\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i = -T_iP_i\boldsymbol{\theta}_i[k] - T_i\boldsymbol{s}_i[k]$$

$$\tilde{\lambda}_i = -\tilde{P}_i[k]\boldsymbol{\theta}_i[k] - \tilde{s}_i[k]$$

Deprived Systems

Under attack!

- Normal behavior

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k],$$

- Under attack

$$\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i = -T_iP_i\boldsymbol{\theta}_i[k] - T_i\boldsymbol{s}_i[k]$$

$$\tilde{\boldsymbol{\lambda}}_i = -\tilde{P}_i[k]\boldsymbol{\theta}_i[k] - \tilde{\boldsymbol{s}}_i[k]$$

Deprived Systems

Under attack!

- Normal behavior

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k],$$

- Under attack

$$\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i = -T_iP_i\boldsymbol{\theta}_i[k] - T_i\boldsymbol{s}_i[k]$$

$$\tilde{\boldsymbol{\lambda}}_i = -\tilde{P}_i[k]\boldsymbol{\theta}_i[k] - \tilde{\boldsymbol{s}}_i[k]$$

CentraleSupélec

Deprived Systems

Under attack!

- Normal behavior

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k],$$

- Under attack

$$\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i = -T_iP_i\boldsymbol{\theta}_i[k] - T_i\boldsymbol{s}_i[k]$$

$$\tilde{\boldsymbol{\lambda}}_i = -\tilde{P}_i[k]\boldsymbol{\theta}_i[k] - \tilde{\boldsymbol{s}}_i[k]$$

$$\boldsymbol{\theta}^{(p+1)} = \tilde{\mathcal{A}}_\theta[k]\boldsymbol{\theta}^{(p)} + \tilde{\mathcal{B}}_\theta[k]$$

CentraleSupélec

Deprived Systems

Under attack!

- Normal behavior

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k],$$

- Under attack

$$\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i = -T_iP_i\boldsymbol{\theta}_i[k] - T_i\boldsymbol{s}_i[k]$$

$$\tilde{\boldsymbol{\lambda}}_i = -\tilde{P}_i[k]\boldsymbol{\theta}_i[k] - \tilde{\boldsymbol{s}}_i[k]$$

$$\boldsymbol{\theta}^{(p+1)} = \tilde{\mathcal{A}}_\theta[k]\boldsymbol{\theta}^{(p)} + \tilde{\mathcal{B}}_\theta[k]$$

CentraleSupélec

Detection Mechanism

Assumption

We know nominal $\bar{P}_i$

Assumption

Attacker chooses $\tilde{\boldsymbol{\lambda}}_i = \gamma_i(\boldsymbol{\lambda}_i) = T_i(k)\boldsymbol{\lambda}_i$
$-T_i(k)P_i\boldsymbol{\theta}_i - T_i(k)\boldsymbol{s}_i(k) \rightarrow -\tilde{P}_i\boldsymbol{\theta}_i - \tilde{\boldsymbol{s}}_i(k)$

- We can estimate[1] $\hat{P}_i$ and $\widehat{\tilde{\boldsymbol{s}}}_i(k)$ such as:

$$\tilde{\boldsymbol{\lambda}}_i = \gamma_i(\boldsymbol{\lambda}_i(\boldsymbol{\theta}_i)) = -\widehat{\tilde{P}_i}(k)\boldsymbol{\theta}_i - \widehat{\tilde{\boldsymbol{s}}}_i(k)$$

- If $\widehat{\tilde{P}}_i(k) \neq \bar{P}_i \rightarrow$ Attack

[1] Using Recursive Least Squares

CentraleSupélec

Detection Mechanism

Assumption

We know nominal $\bar{P}_i$

Assumption

Attacker chooses $\tilde{\boldsymbol{\lambda}}_i = \gamma_i(\boldsymbol{\lambda}_i) = T_i(k)\boldsymbol{\lambda}_i$
$-T_i(k)P_i\boldsymbol{\theta}_i - T_i(k)\boldsymbol{s}_i(k) \rightarrow -\tilde{P}_i\boldsymbol{\theta}_i - \tilde{\boldsymbol{s}}_i(k)$

- We can estimate[1] $\hat{P}_i$ and $\widehat{\tilde{\boldsymbol{s}}}_i(k)$ such as:

$$\tilde{\boldsymbol{\lambda}}_i = \gamma_i(\boldsymbol{\lambda}_i(\boldsymbol{\theta}_i)) = -\widehat{\tilde{P}_i}(k)\boldsymbol{\theta}_i - \widehat{\tilde{\boldsymbol{s}}}_i(k)$$

- If $\widehat{\tilde{P}}_i(k) \neq \bar{P}_i \rightarrow$ Attack

[1] Using Recursive Least Squares

CentraleSupélec

Detection Mechanism

Assumption

We know nominal $\bar{P}_i$

Assumption

Attacker chooses $\tilde{\boldsymbol{\lambda}}_i = \gamma_i(\boldsymbol{\lambda}_i) = T_i(k)\boldsymbol{\lambda}_i$
$-T_i(k)P_i\boldsymbol{\theta}_i - T_i(k)\boldsymbol{s}_i(k) \rightarrow -\tilde{P}_i\boldsymbol{\theta}_i - \tilde{\boldsymbol{s}}_i(k)$

- We can estimate[1] $\hat{P}_i$ and $\widehat{\tilde{\boldsymbol{s}}}_i(k)$ such as:

$$\tilde{\boldsymbol{\lambda}}_i = \gamma_i(\boldsymbol{\lambda}_i(\boldsymbol{\theta}_i)) = -\widehat{\tilde{P}_i}(k)\boldsymbol{\theta}_i - \widehat{\tilde{\boldsymbol{s}}}_i(k)$$

- If $\widehat{\tilde{P}}_i(k) \neq \bar{P}_i \rightarrow$ Attack

[1] Using Recursive Least Squares

CentraleSupélec

Detection Mechanism

Assumption

We know nominal $\bar{P}_i$

Assumption

Attacker chooses $\tilde{\boldsymbol{\lambda}}_i = \gamma_i(\boldsymbol{\lambda}_i) = T_i(k)\boldsymbol{\lambda}_i$
$-T_i(k)P_i\boldsymbol{\theta}_i - T_i(k)\boldsymbol{s}_i(k) \rightarrow -\tilde{P}_i\boldsymbol{\theta}_i - \tilde{\boldsymbol{s}}_i(k)$

- We can estimate[1] $\hat{P}_i$ and $\hat{\tilde{\boldsymbol{s}}}_i(k)$ such as:

$$\tilde{\boldsymbol{\lambda}}_i = \gamma_i(\boldsymbol{\lambda}_i(\boldsymbol{\theta}_i)) = -\widehat{\tilde{P}_i}(k)\boldsymbol{\theta}_i - \hat{\tilde{\boldsymbol{s}}}_i(k)$$

- If $\hat{\tilde{P}}_i(k) \neq \bar{P}_i \rightarrow$ Attack

[1] Using Recursive Least Squares

Detection Mechanism

Assumption

We know nominal $\bar{P}_i$

Assumption

Attacker chooses $\tilde{\boldsymbol{\lambda}}_i = \gamma_i(\boldsymbol{\lambda}_i) = T_i(k)\boldsymbol{\lambda}_i$
$-T_i(k)P_i\boldsymbol{\theta}_i - T_i(k)\boldsymbol{s}_i(k) \rightarrow -\tilde{P}_i\boldsymbol{\theta}_i - \tilde{\boldsymbol{s}}_i(k)$

- We can estimate[1] $\hat{P}_i$ and $\hat{\tilde{\boldsymbol{s}}}_i(k)$ such as:

$$\tilde{\boldsymbol{\lambda}}_i = \gamma_i(\boldsymbol{\lambda}_i(\boldsymbol{\theta}_i)) = -\widehat{\tilde{P}_i}(k)\boldsymbol{\theta}_i - \hat{\tilde{\boldsymbol{s}}}_i(k)$$

- If $\hat{\tilde{P}}_i(k) \neq \bar{P}_i \rightarrow$ Attack

[1] Using Recursive Least Squares

CentraleSupélec

Detection

In detail

- Error $E_i(k) = \|\widehat{\tilde{P}}_i(k) - \bar{P}_i\|_F$
- Create threshold ϵ_P
- Indicator $d_i \in \{0, 1\}$ detects the attack in agent i.
- $d_i = 1$ if $E_i(k) > \epsilon_P$, 0 otherwise

Detection

In detail

- Error $E_i(k) = \|\widehat{\tilde{P}}_i(k) - \bar{P}_i\|_F$
- Create threshold ϵ_P
- Indicator $d_i \in \{0, 1\}$ detects the attack in agent i.
- $d_i = 1$ if $E_i(k) > \epsilon_P$, 0 otherwise

Detection

In detail

- Error $E_i(k) = \|\widehat{\tilde{P}}_i(k) - \bar{P}_i\|_F$
- Create threshold ϵ_P
- Indicator $d_i \in \{0, 1\}$ detects the attack in agent i.
- $d_i = 1$ if $E_i(k) > \epsilon_P$, 0 otherwise

Detection

In detail

- Error $E_i(k) = \|\widehat{\tilde{P}}_i(k) - \bar{P}_i\|_F$
- Create threshold ϵ_P
- Indicator $d_i \in \{0, 1\}$ detects the attack in agent i.
- $d_i = 1$ if $E_i(k) > \epsilon_P$, 0 otherwise

CentraleSupélec

Detection

In detail

- Error $E_i(k) = \|\widehat{\tilde{P}}_i(k) - \bar{P}_i\|_F$
- Create threshold ϵ_P
- Indicator $d_i \in \{0, 1\}$ detects the attack in agent i.
- $d_i = 1$ if $E_i(k) > \epsilon_P$, 0 otherwise

CentraleSupélec

Estimating $\widehat{\widetilde{P}}_i[k]$

- We estimate $\hat{P}_i$ and $\widehat{\widetilde{\boldsymbol{s}}}_i(k)$ simultaneously using Recursive Least Squares
- Problem: Estimation during negotiation fails
  - Consecutive $\boldsymbol{\lambda}_i^p$ and $\boldsymbol{\theta}_i^p$ are linearly dependent $\rightarrow$ low input excitation
- Solution: Send sequence of random values of $\boldsymbol{\theta}_i$ until estimates converge

CentraleSupélec

Estimating $\widehat{\widetilde{P}}_i[k]$

- We estimate $\hat{P}_i$ and $\widehat{\tilde{\boldsymbol{s}}}_i(k)$ simultaneously using Recursive Least Squares
- Problem: Estimation during negotiation fails
  - Consecutive $\boldsymbol{\lambda}_i^p$ and $\boldsymbol{\theta}_i^p$ are linearly dependent $\rightarrow$ low input excitation
- Solution: Send sequence of random values of $\boldsymbol{\theta}_i$ until estimates converge

Estimating $\widehat{\widetilde{P}}_i[k]$

- We estimate $\hat{P}_i$ and $\widehat{\tilde{\boldsymbol{s}}}_i(k)$ simultaneously using Recursive Least Squares
- Problem: Estimation during negotiation fails
  - Consecutive $\boldsymbol{\lambda}_i^p$ and $\boldsymbol{\theta}_i^p$ are linearly dependent $\rightarrow$ low input excitation
- Solution: Send sequence of random values of $\boldsymbol{\theta}_i$ until estimates converge

CentraleSupélec

Estimating $\widehat{\widetilde{P}}_i[k]$

- We estimate $\hat{P}_i$ and $\widehat{\widetilde{\boldsymbol{s}}}_i(k)$ simultaneously using Recursive Least Squares
- Problem: Estimation during negotiation fails
  - Consecutive $\boldsymbol{\lambda}_i^p$ and $\boldsymbol{\theta}_i^p$ are linearly dependent $\rightarrow$ low input excitation
- Solution: Send sequence of random values of $\boldsymbol{\theta}_i$ until estimates converge

CentraleSupélec

Mitigation mechanism

- Main idea: Reconstruct $\boldsymbol{\lambda}_i$ and use in negotiation

Assumption

We suppose $\tilde{\boldsymbol{\lambda}}_i = \mathbf{0}$ only if $\boldsymbol{\lambda}_i = \mathbf{0}$, which implies $T_i(k)$ invertible.

- Estimate the inverse of $T_i(k)$

$$\widehat{T_i(k)^{-1}} = \bar{P}_i\, \widehat{\tilde{P}_i}(k)^{-1}$$

- Reconstruct $\boldsymbol{\lambda}_i$

$$\boldsymbol{\lambda}_{i\mathrm{rec}} = \widehat{T_i(k)^{-1}}\tilde{\boldsymbol{\lambda}}_i = -\bar{P}_i\boldsymbol{\theta}_i - \widehat{T_i(k)^{-1}}\hat{\tilde{\boldsymbol{s}}}_i(k)$$

Mitigation mechanism

- Main idea: Reconstruct $\boldsymbol{\lambda}_i$ and use in negotiation

Assumption

We suppose $\tilde{\boldsymbol{\lambda}}_i = \mathbf{0}$ only if $\boldsymbol{\lambda}_i = \mathbf{0}$, which implies $T_i(k)$ invertible.

- Estimate the inverse of $T_i(k)$

$$\widehat{T_i(k)^{-1}} = \bar{P}_i \, \widehat{\tilde{P}_i(k)^{-1}}$$

- Reconstruct $\boldsymbol{\lambda}_i$

$$\boldsymbol{\lambda}_{i\mathrm{rec}} = \widehat{T_i(k)^{-1}} \tilde{\boldsymbol{\lambda}}_i = -\bar{P}_i \boldsymbol{\theta}_i - \widehat{T_i(k)^{-1}} \hat{\boldsymbol{s}}_i(k)$$

CentraleSupélec

Mitigation mechanism

- Main idea: Reconstruct $\boldsymbol{\lambda}_i$ and use in negotiation

Assumption

We suppose $\tilde{\boldsymbol{\lambda}}_i = \mathbf{0}$ only if $\boldsymbol{\lambda}_i = \mathbf{0}$, which implies $T_i(k)$ invertible.

- Estimate the inverse of $T_i(k)$

$$\widehat{T_i(k)^{-1}} = \bar{P}_i \, \widehat{\tilde{P}_i}(k)^{-1}$$

- Reconstruct $\boldsymbol{\lambda}_i$

$$\boldsymbol{\lambda}_{i\mathrm{rec}} = \widehat{T_i(k)^{-1}} \tilde{\boldsymbol{\lambda}}_i = -\bar{P}_i \boldsymbol{\theta}_i - \widehat{T_i(k)^{-1}} \hat{\boldsymbol{s}}_i(k)$$

CentraleSupélec

Mitigation mechanism

- Main idea: Reconstruct $\boldsymbol{\lambda}_i$ and use in negotiation

Assumption

We suppose $\tilde{\boldsymbol{\lambda}}_i = \mathbf{0}$ only if $\boldsymbol{\lambda}_i = \mathbf{0}$, which implies $T_i(k)$ invertible.

- Estimate the inverse of $T_i(k)$

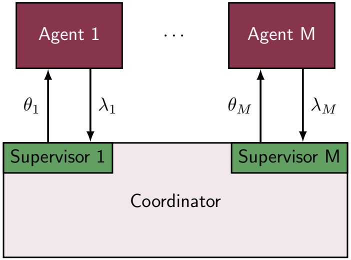
$$\widehat{T_i(k)^{-1}} = \bar{P}_i \, \widehat{\tilde{P}_i}(k)^{-1}$$

- Reconstruct $\boldsymbol{\lambda}_i$

$$\boldsymbol{\lambda}_{i\text{rec}} = \widehat{T_i(k)^{-1}} \tilde{\boldsymbol{\lambda}}_i = -\bar{P}_i \boldsymbol{\theta}_i - \widehat{T_i(k)^{-1}} \hat{\tilde{\boldsymbol{s}}}_i(k)$$

CentraleSupélec

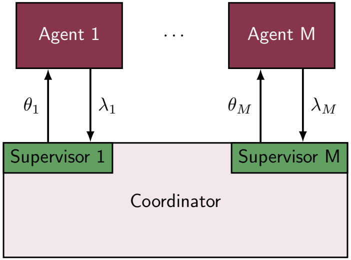
Complete Mechanism



Two phases:

1. Detect which agents are non-cooperative
2. Reconstruct λ_i and use in negotiation

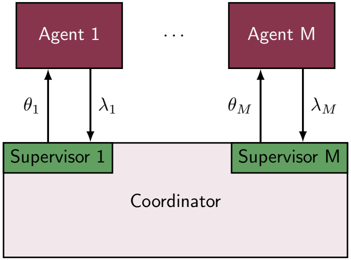
Complete Mechanism



Two phases:

1. Detect which agents are non-cooperative
2. Reconstruct λ_i and use in negotiation

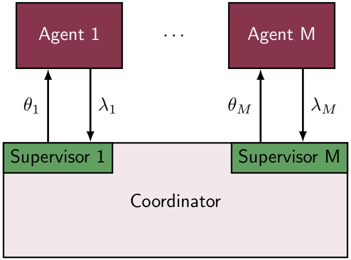
Complete Mechanism



Two phases:

1. Detect which agents are non-cooperative
2. Reconstruct λ_i and use in negotiation

Complete Mechanism

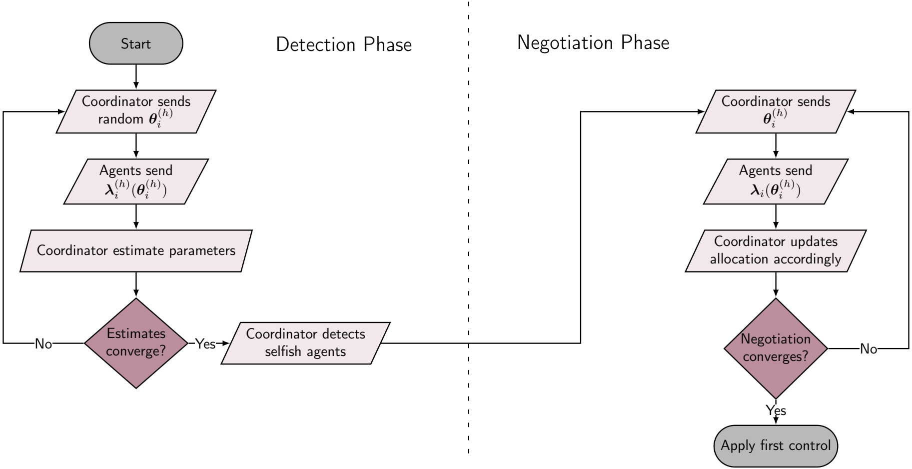


Two phases:

1. Detect which agents are non-cooperative
2. Reconstruct $\boldsymbol{\lambda}_i$ and use in negotiation

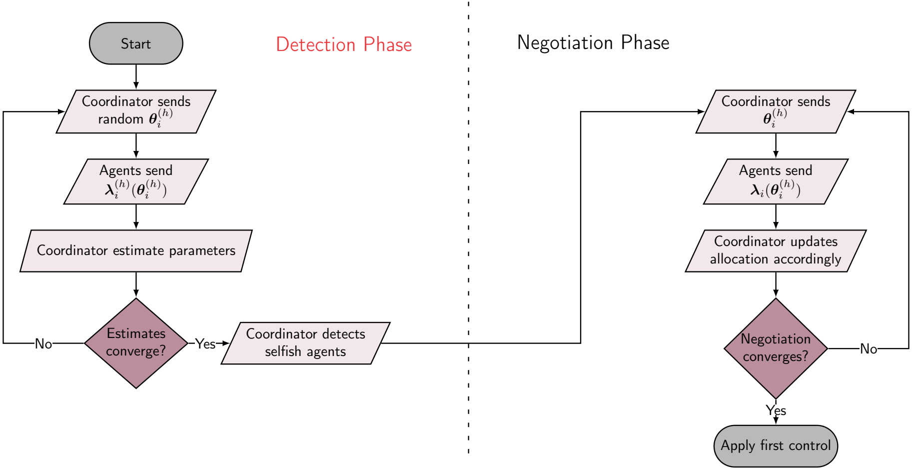
Complete algorithm

RPdMPC-DS



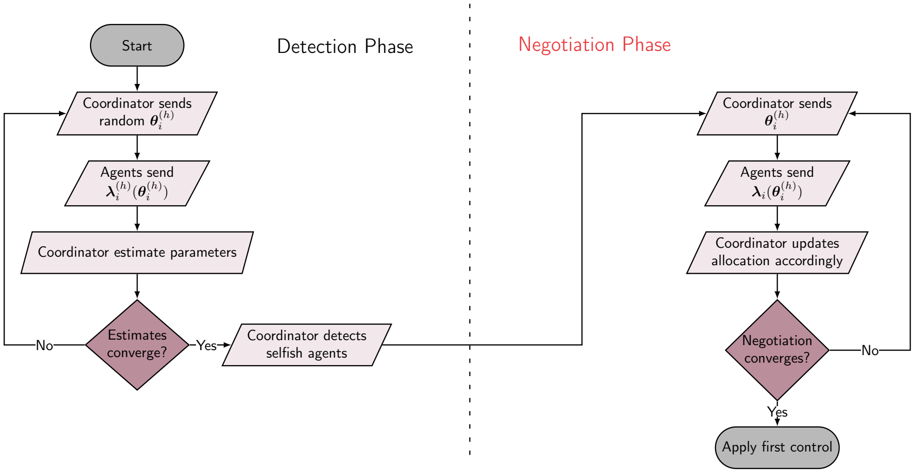
Complete algorithm

RPdMPC-DS



Complete algorithm

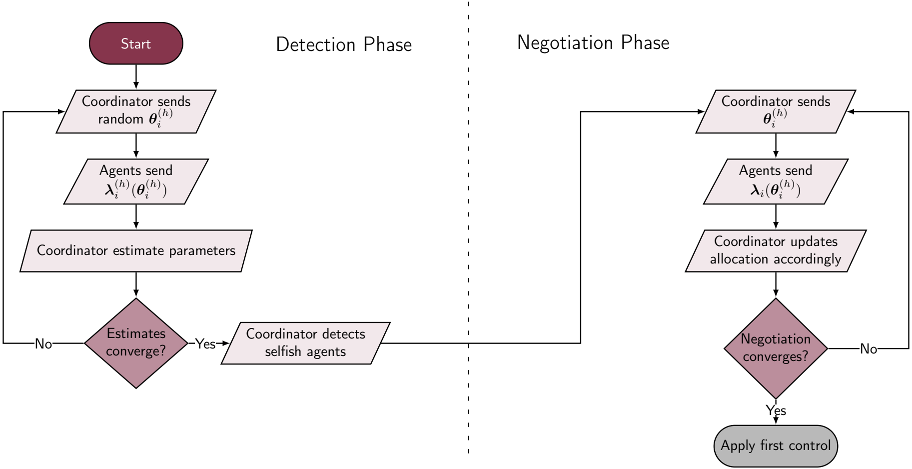
RPdMPC-DS



CentraleSupélec

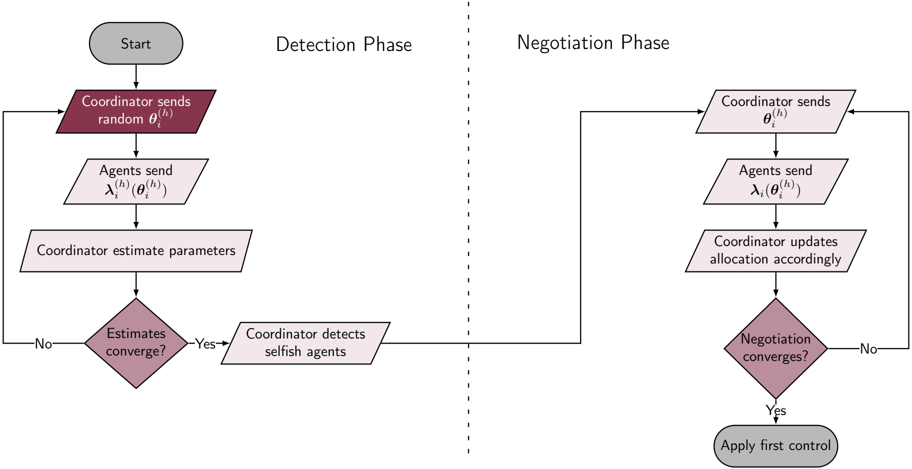
Complete algorithm

RPdMPC-DS



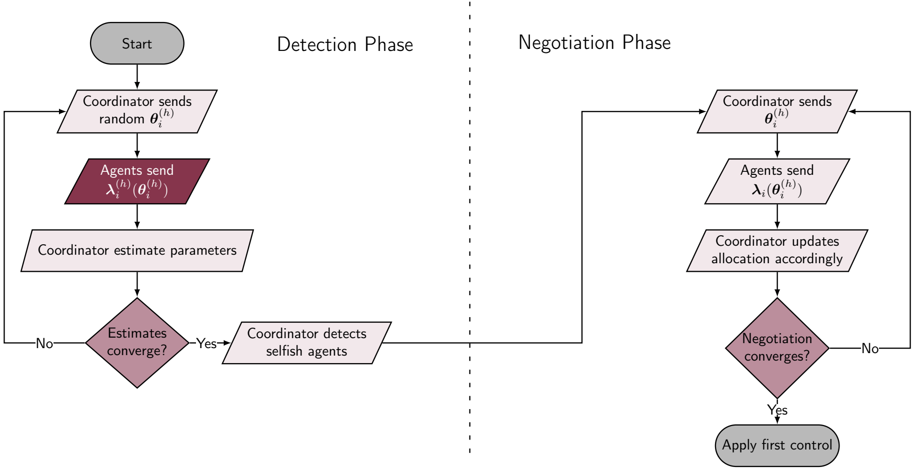
Complete algorithm

RPdMPC-DS



Complete algorithm

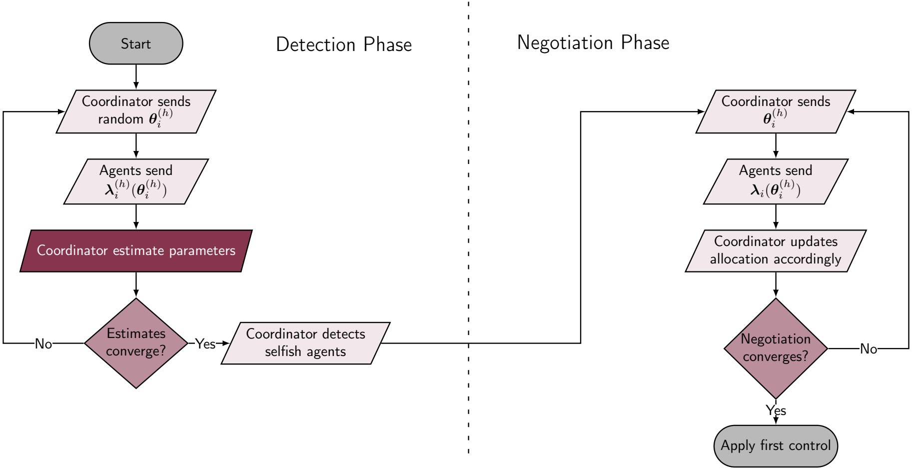
RPdMPC-DS



CentraleSupélec

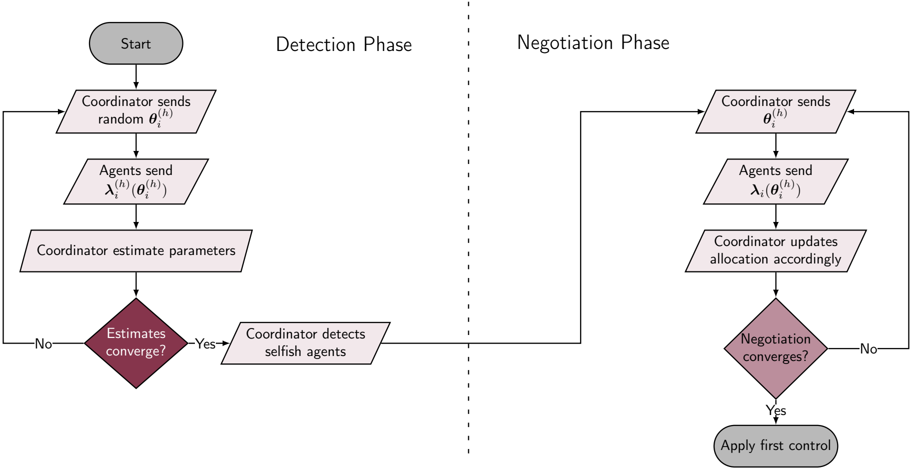
Complete algorithm

RPdMPC-DS



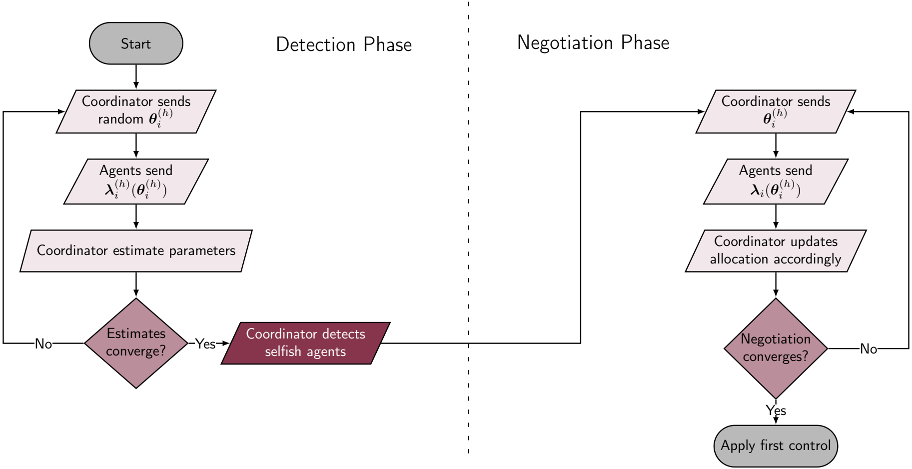
Complete algorithm

RPdMPC-DS



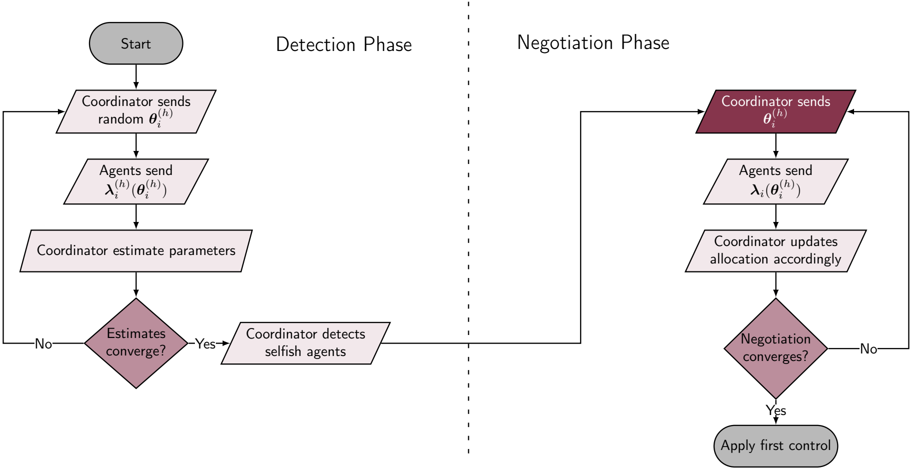
Complete algorithm

RPdMPC-DS



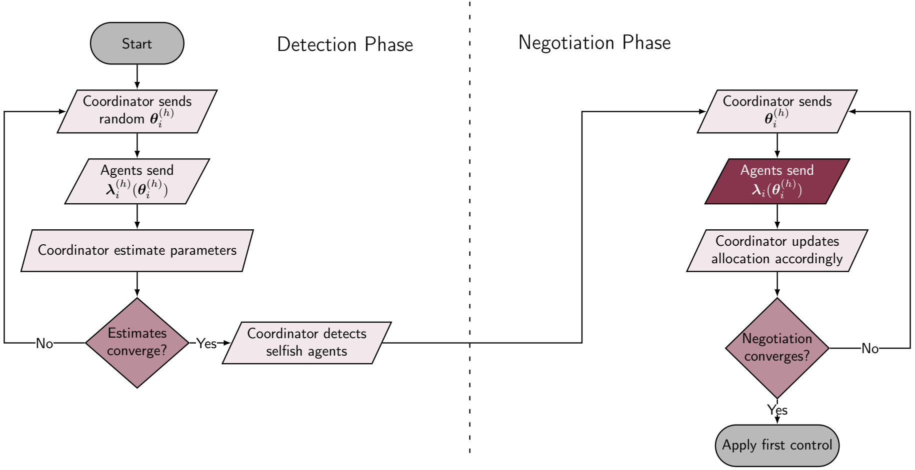
Complete algorithm

RPdMPC-DS



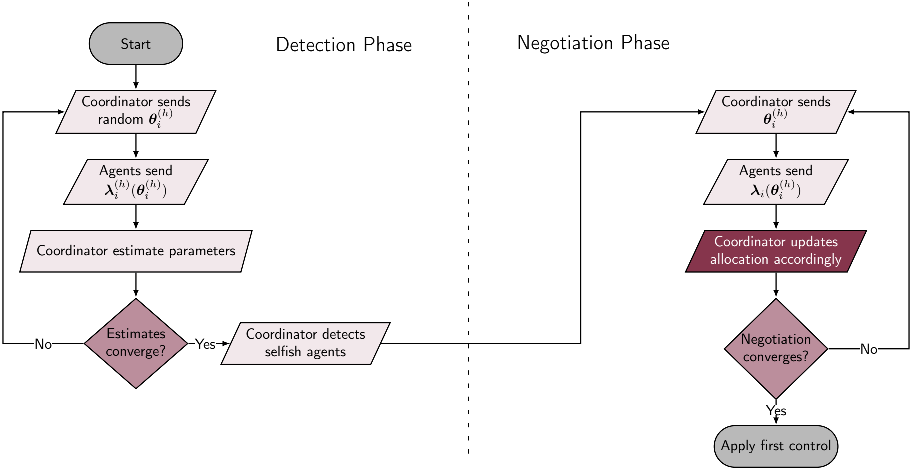
Complete algorithm

RPdMPC-DS



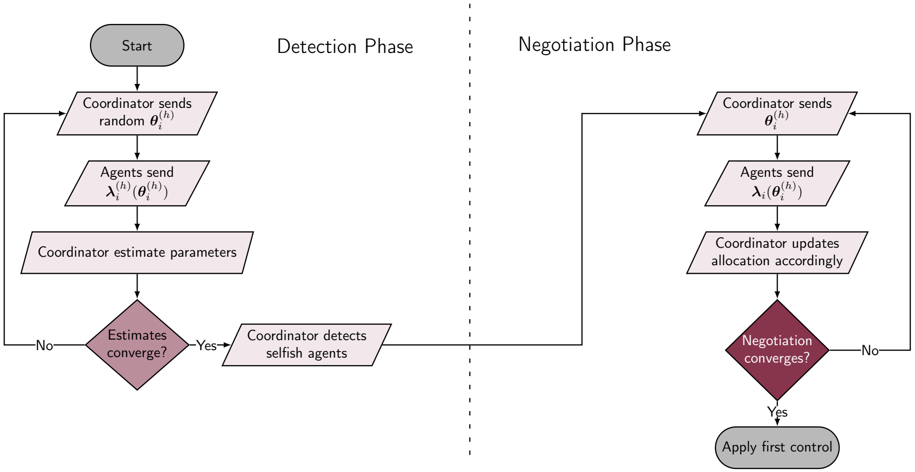
Complete algorithm

RPdMPC-DS



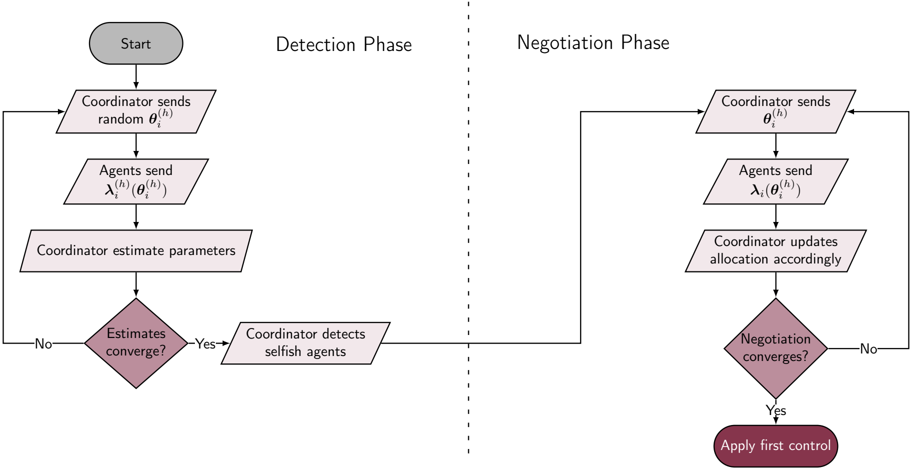
Complete algorithm

RPdMPC-DS



Complete algorithm

RPdMPC-DS



Example

District Heating of 4 Distinct Houses Under Power Scarcity

- 4 distinct rooms modeled using 3R-2C
- Initial temperature under 20^oC
- Not enough power to achieve setpoint $\left(\sum_{i=1}^{4} \boldsymbol{u}_i(k) \leqslant 4\text{kW}\right)$
- Simulated for a period of 5h
- ZOH $T_s = 0.25\text{h}$
- 3 scenarios
  1. Nominal
  2. Agent I non cooperative from k>6 with T=4*I
  3. Similar but with secure algorithm

CentraleSupélec

Example

District Heating of 4 Distinct Houses Under Power Scarcity

- 4 distinct rooms modeled using 3R-2C
- Initial temperature under 20^oC
- Not enough power to achieve setpoint $\left(\sum_{i=1}^{4} \boldsymbol{u}_i(k) \leqslant 4\text{kW}\right)$
- Simulated for a period of 5h
- ZOH $T_s = 0.25\text{h}$
- 3 scenarios
  1. Nominal
  2. Agent I non cooperative from k>6 with T=4*I
  3. Similar but with secure algorithm

CentraleSupélec

Example

District Heating of 4 Distinct Houses Under Power Scarcity

- 4 distinct rooms modeled using 3R-2C
- Initial temperature under $20^o\mathrm{C}$
- Not enough power to achieve setpoint $\left(\sum_{i=1}^{4} \boldsymbol{u}_i(k) \leqslant 4\mathrm{kW}\right)$
- Simulated for a period of $5\mathrm{h}$
- ZOH $T_s = 0.25\mathrm{h}$
- 3 scenarios
  1. Nominal
  2. Agent I non cooperative from k>6 with T=4*I
  3. Similar but with secure algorithm

CentraleSupélec

Example

District Heating of 4 Distinct Houses Under Power Scarcity

- 4 distinct rooms modeled using 3R-2C
- Initial temperature under 20^oC
- Not enough power to achieve setpoint $\left(\sum_{i=1}^{4} \boldsymbol{u}_i(k) \leqslant 4\text{kW}\right)$
- Simulated for a period of 5h
- ZOH $T_s = 0.25\text{h}$
- 3 scenarios
  1. Nominal
  2. Agent I non cooperative from k>6 with T=4*I
  3. Similar but with secure algorithm

CentraleSupélec

Example

District Heating of 4 Distinct Houses Under Power Scarcity

- 4 distinct rooms modeled using 3R-2C
- Initial temperature under $20^o\mathrm{C}$
- Not enough power to achieve setpoint $\left(\sum_{i=1}^{4} \boldsymbol{u}_i(k) \leqslant 4\mathrm{kW}\right)$
- Simulated for a period of $5\mathrm{h}$
- ZOH $T_s = 0.25\mathrm{h}$
- 3 scenarios
  1. Nominal
  2. Agent I non cooperative from k>6 with T=4*I
  3. Similar but with secure algorithm

CentraleSupélec

Example

District Heating of 4 Distinct Houses Under Power Scarcity

- 4 distinct rooms modeled using 3R-2C
- Initial temperature under $20^{o}\mathrm{C}$
- Not enough power to achieve setpoint $\left(\sum_{i=1}^{4} \boldsymbol{u}_i(k) \leqslant 4\mathrm{kW}\right)$
- Simulated for a period of $5\mathrm{h}$
- ZOH $T_s = 0.25\mathrm{h}$
- 3 scenarios
  1. Nominal
  2. Agent I non cooperative from k>6 with T=4*I
  3. Similar but with secure algorithm

CentraleSupélec

Example

District Heating of 4 Distinct Houses Under Power Scarcity

- 4 distinct rooms modeled using 3R-2C
- Initial temperature under $20^o\mathrm{C}$
- Not enough power to achieve setpoint $\left(\sum_{i=1}^{4} \boldsymbol{u}_i(k) \leqslant 4\mathrm{kW}\right)$
- Simulated for a period of $5\mathrm{h}$
- ZOH $T_s = 0.25\mathrm{h}$
- 3 scenarios
  1. Nominal
  2. Agent I non cooperative from k>6 with T=4*I
  3. Similar but with secure algorithm

CentraleSupélec

Results

Temporal

- N Nominal
- S Selflish behavior
- C selfish behavior with Correction

Results

Temporal

- N Nominal
- S Selflish behavior
- C selfish behavior with Correction

CentraleSupélec

Results

Table 1: Comparison of costs J_i^N and J_G^N

Agent	Nominal	Selfish	Selfish + correction
I	103	64	104
II	73	91	73
III	100	123	101
IV	132	154	131
Global	408	442	409

Results

Table 1: Comparison of costs J_i^N and J_G^N

Agent	Nominal	Selfish	Selfish + correction
I	103	64	104
II	73	91	73
III	100	123	101
IV	132	154	131
Global	408	442	409

Outline

3. Resilient Primal Decomposition-based dMPC using Artificial Scarcity
    - Relaxing some assumptions
    - Adapting the algorithm
    - Results

Analysing

$$\boldsymbol{\lambda}_i[k] = \begin{cases} -P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}_{\boldsymbol{\lambda}_i}^{n} \\ \vdots & \vdots \\ -P_i^{\left(2^{n_{\text{ineq}}}-1\right)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{\left(2^{n_{\text{ineq}}}-1\right)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}_{\boldsymbol{\lambda}_i}^{2^{n_{\text{ineq}}}-1} \end{cases}. \tag{1}$$

Artificial Scarcity

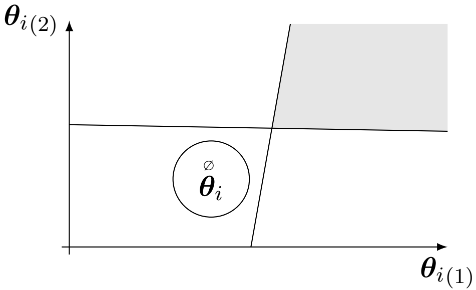


Figure 1: Ball $\mathcal{B}(\overset{\varnothing}{\boldsymbol{\theta}}_i, r)$.

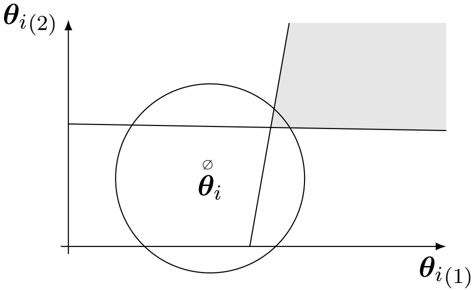


Figure 2: Ball $\mathcal{B}(\overset{\varnothing}{\boldsymbol{\theta}}_i, r)$ traversing zones.

Expectation Maximization

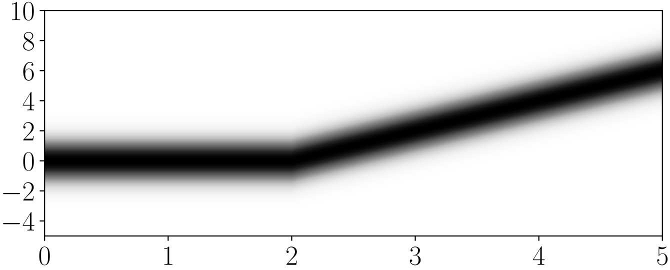


Figure 3: Gaussian Mixture for a 1D PWA function with 2 modes.

Expectation Maximization

Algorithm

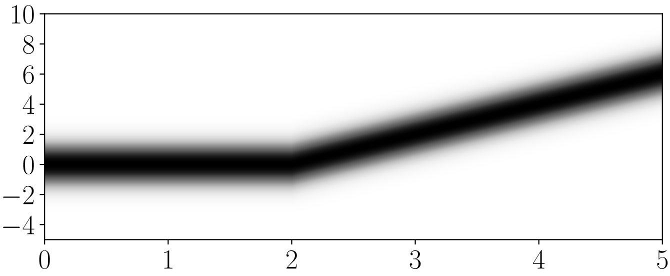
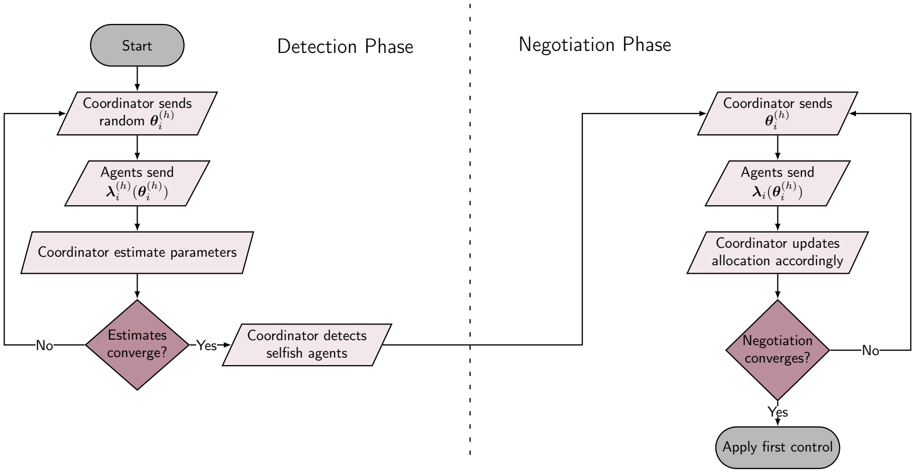


Figure 4: Gaussian Mixture for a 1D PWA function with 2 modes.

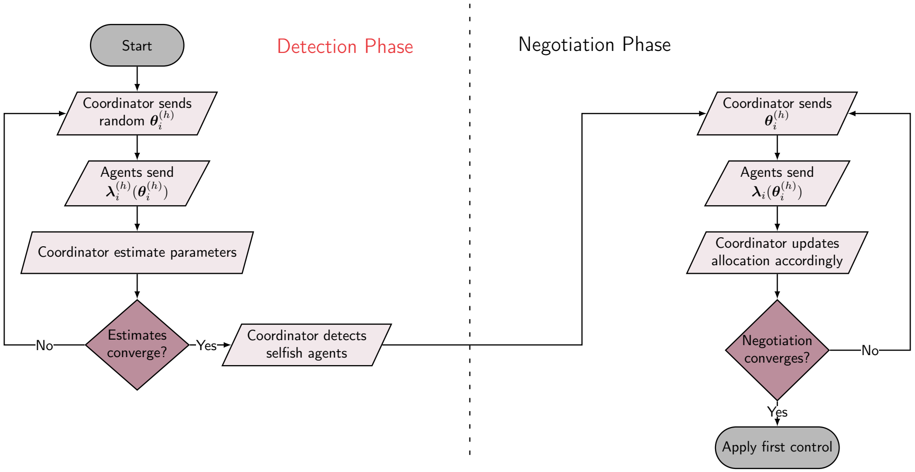
Complete algorithm

RPdMPC-AS



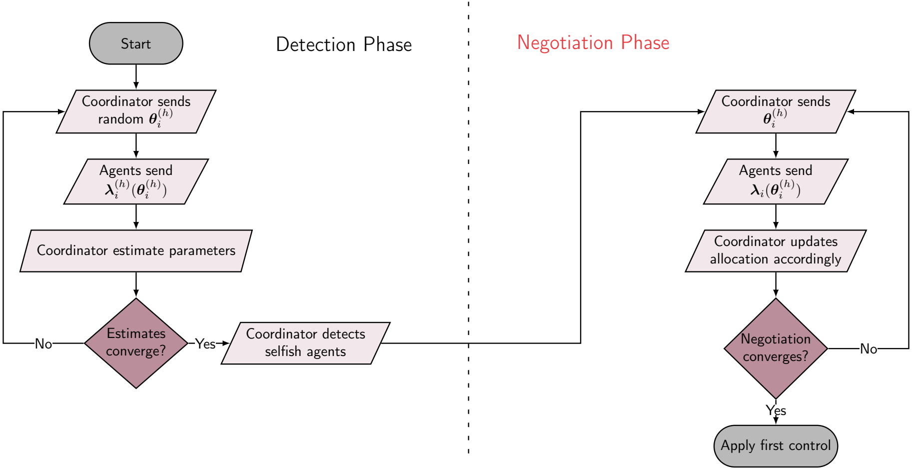
Complete algorithm

RPdMPC-AS



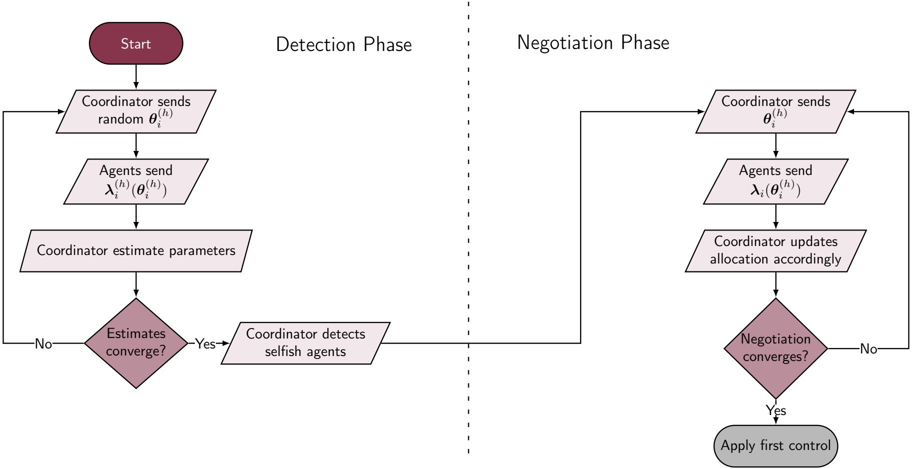
Complete algorithm

RPdMPC-AS



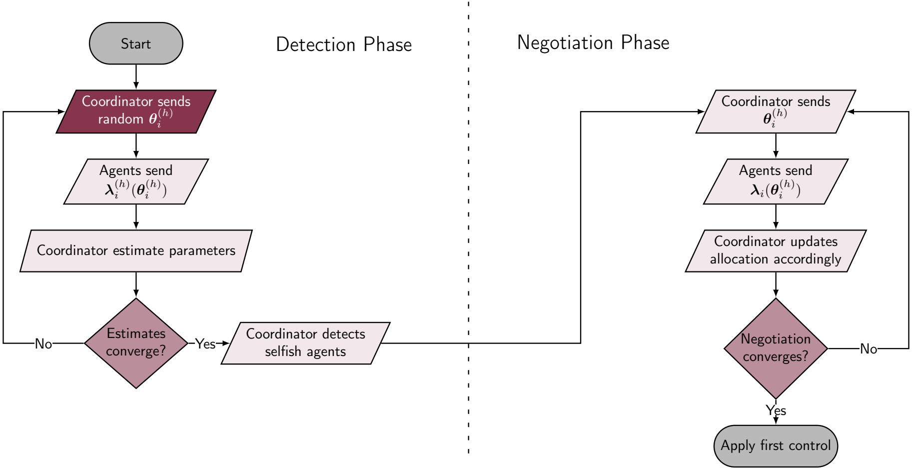
Complete algorithm

RPdMPC-AS



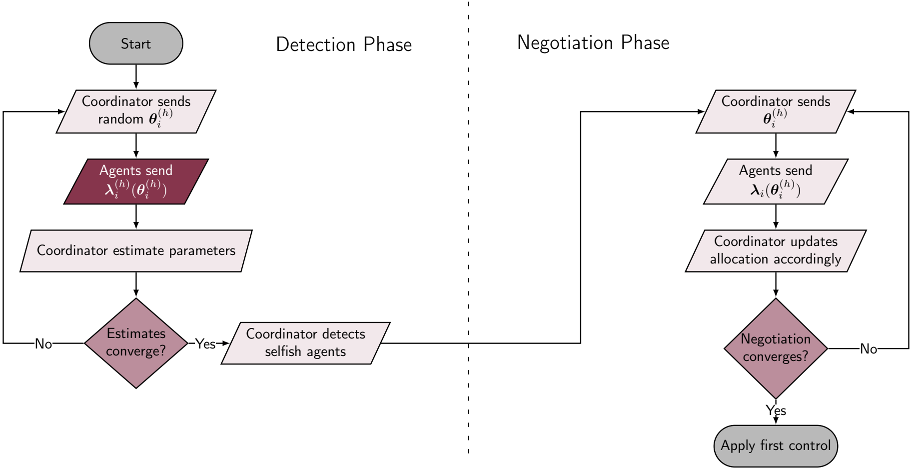
Complete algorithm

RPdMPC-AS



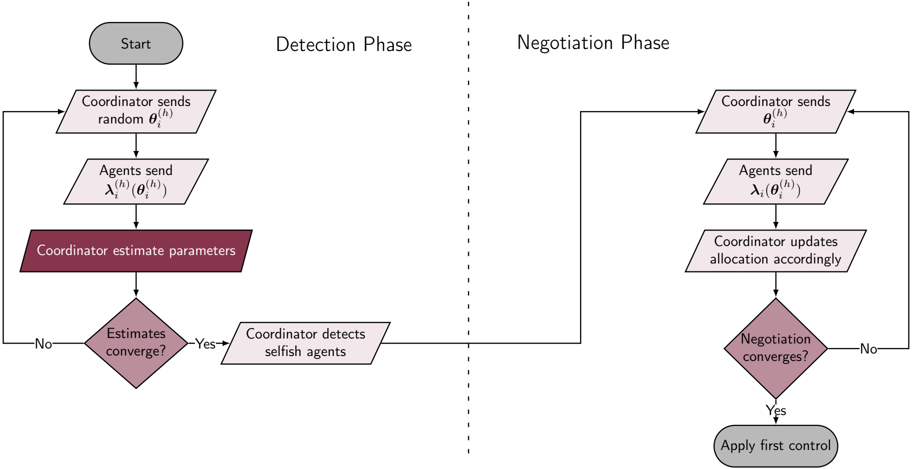
Complete algorithm

RPdMPC-AS



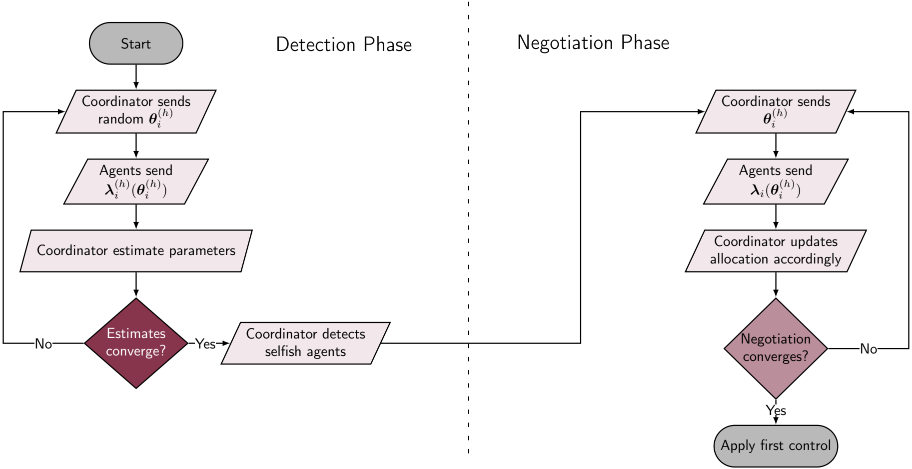
Complete algorithm

RPdMPC-AS



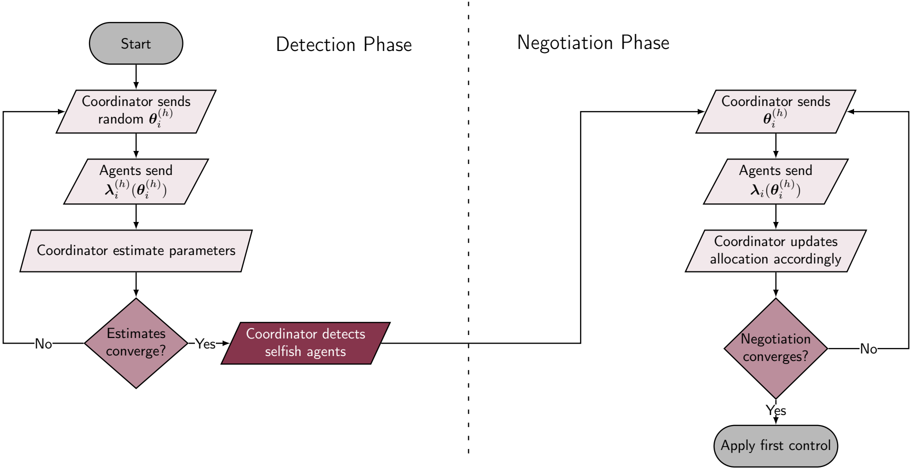
Complete algorithm

RPdMPC-AS



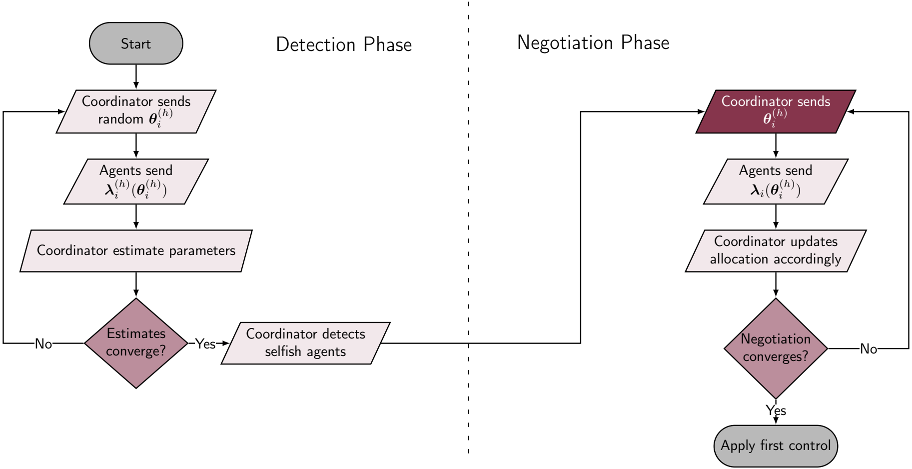
Complete algorithm

RPdMPC-AS



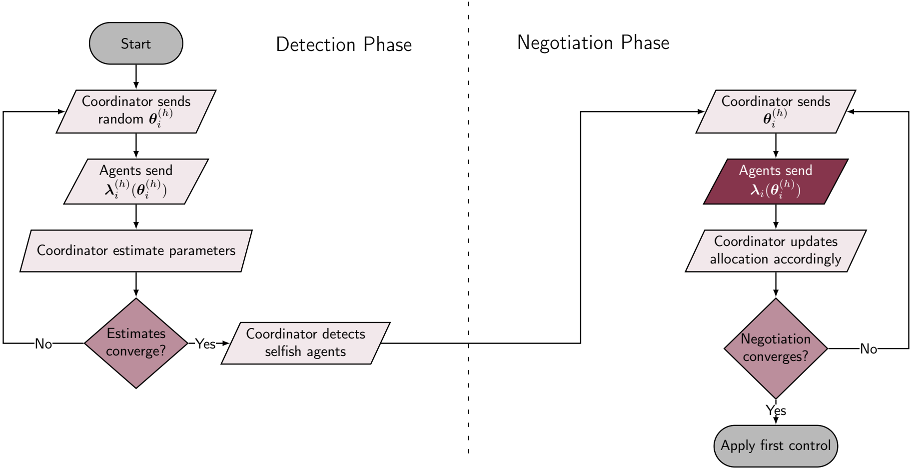
Complete algorithm

RPdMPC-AS



Complete algorithm

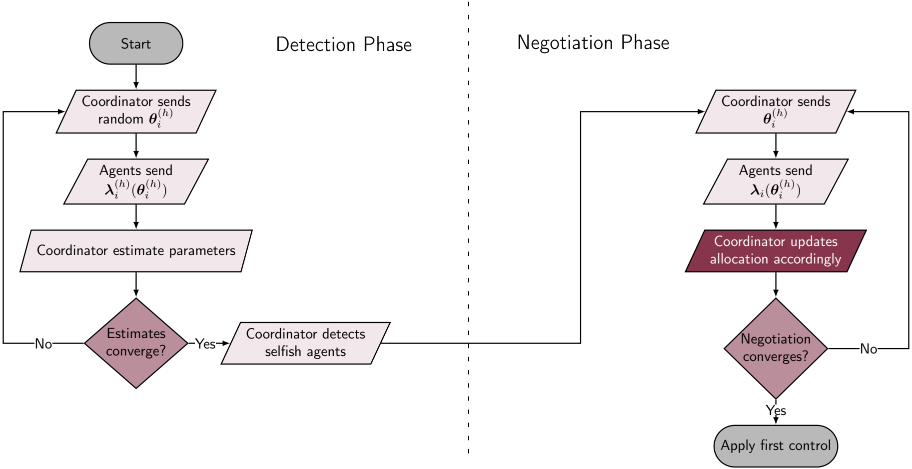
RPdMPC-AS



CentraleSupélec

Complete algorithm

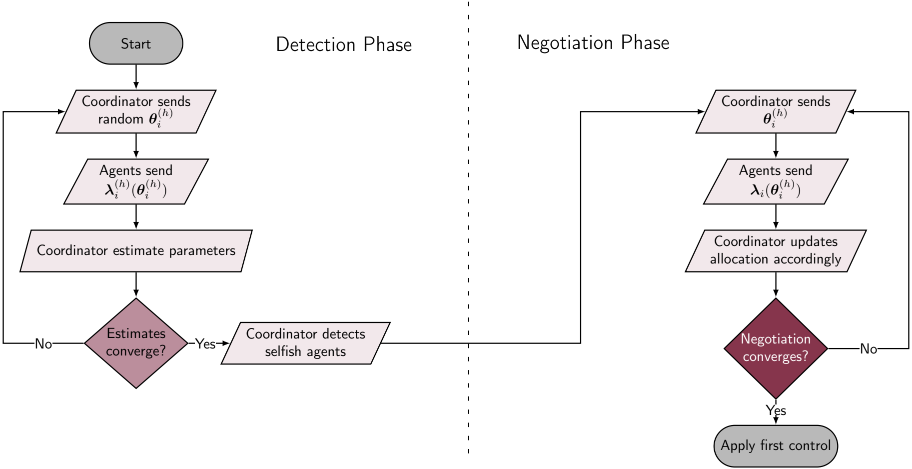
RPdMPC-AS



CentraleSupélec

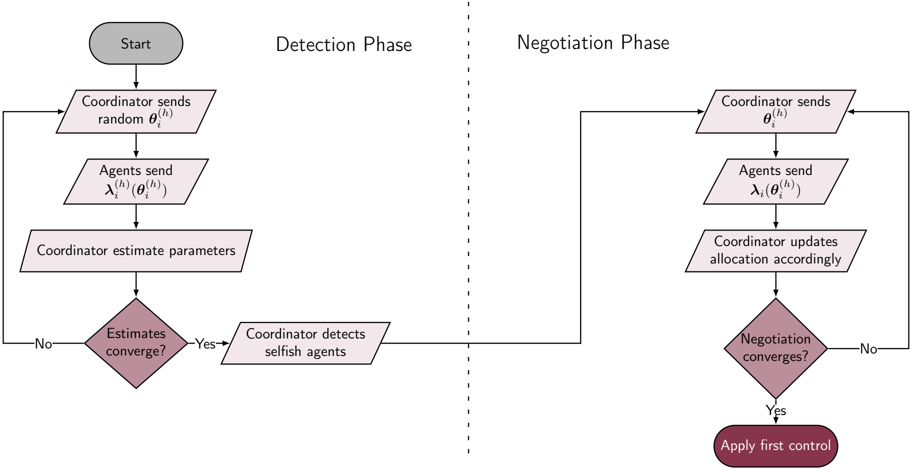
Complete algorithm

RPdMPC-AS



Complete algorithm

RPdMPC-AS



Thank you!

Repository
https://github.com/Accacio/thesis

Contact
rafael.accacio.nogueira@gmail.com

For Further Reading I

José M Maestre, Rudy R Negenborn, et al. *Distributed Model Predictive Control made easy*. Vol. 69. Springer, 2014. ISBN: 978-94-007-7005-8.

Wicak Ananduta et al. "Resilient Distributed Model Predictive Control for Energy Management of Interconnected Microgrids". In: *Optimal Control Applications and Methods* 41.1 (2020), pp. 146–169. DOI: 10.1002/oca.2534. URL: https://onlinelibrary.wiley.com/doi/pdf/10.1002/oca.2534.

José M. Maestre et al. "Scenario-Based Defense Mechanism Against Vulnerabilities in Lagrange-Based Dmpc". In: *Control Eng Pract* 114 (2021), p. 104879. ISSN: 0967-0661. DOI: 10.1016/j.conengprac.2021.104879.

CentraleSupélec

For Further Reading II

Pablo Velarde et al. "Vulnerabilities in Lagrange-Based Distributed Model Predictive Control". In: *Optimal Control Applications and Methods* 39.2 (Sept. 2018), pp. 601–621. DOI: 10.1002/oca.2368.

Wicak Ananduta et al. "Resilient Distributed Energy Management for Systems of Interconnected Microgrids". In: *2018 IEEE Conference on Decision and Control (CDC)*. 2018, pp. 3159–3164. DOI: 10.1109/CDC.2018.8619548.

Wicak Ananduta et al. "A Resilient Approach for Distributed MPC-Based Economic Dispatch in Interconnected Microgrids". In: *2019 18th European Control Conference (ECC)*. 2019, pp. 691–696. DOI: 10.23919/ECC.2019.8796208.

CentraleSupélec

For Further Reading III

- P. Chanfreut, J. M. Maestre, and H. Ishii. "Vulnerabilities in Distributed Model Predictive Control based on Jacobi-Gauss Decomposition". In: *2018 European Control Conference (ECC)*. June 2018, pp. 2587–2592. DOI: 10.23919/ECC.2018.8550239.
- Pablo Velarde et al. "Scenario-based defense mechanism for distributed model predictive control". In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE. Dec. 2017, pp. 6171–6176. DOI: 10.1109/CDC.2017.8264590.
- Pablo Velarde et al. "Vulnerabilities in Lagrange-Based DMPC in the Context of Cyber-Security". In: *2017 IEEE International Conference on Autonomic Computing (ICAC)*. July 2017, pp. 215–220. DOI: 10.1109/ICAC.2017.53.

Conditions

◂ back

One way to ensure this, is to make the original constraint (??) to have at most as many rows as columns, i.e., $\#\boldsymbol{u}_{\max} \leqslant n_u$, although it may be a little restrictive.

θ dynamics

back

$$\boldsymbol{\theta}^{(p+1)} = \mathcal{A}_\theta \boldsymbol{\theta}^{(p)} + \mathcal{B}_\theta[k]$$

where

$$\mathcal{A}_\theta = \begin{bmatrix} I - \frac{M-1}{M}\rho^{(p)}P_1 & \frac{1}{M}\rho^{(p)}P_2 & \dots & \frac{1}{M}\rho^{(p)}P_M \\ \frac{1}{M}\rho^{(p)}P_1 & I - \frac{M-1}{M}\rho^{(p)}P_2 & \dots & \frac{1}{M}\rho^{(p)}P_M \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{M}\rho^{(p)}P_1 & \frac{1}{M}\rho^{(p)}P_2 & \dots & I - \frac{M-1}{M}\rho^{(p)}P_M \end{bmatrix}$$

$$\mathcal{B}_\theta[k] = \begin{bmatrix} -\frac{M-1}{M}\rho^{(p)}\boldsymbol{s}_1[k] + \frac{1}{M}\rho^{(p)}\boldsymbol{s}_2[k] \cdots - \frac{1}{M}\rho^{(p)}\boldsymbol{s}_M[k] \\ \frac{1}{M}\rho^{(p)}\boldsymbol{s}_1[k] - \frac{M-1}{M}\rho^{(p)}\boldsymbol{s}_2[k] \cdots - \frac{1}{M}\rho^{(p)}\boldsymbol{s}_M[k] \\ \vdots \\ \frac{1}{M}\rho^{(p)}\boldsymbol{s}_1[k] + \frac{1}{M}\rho^{(p)}\boldsymbol{s}_2[k] \cdots - \frac{M-1}{M}\rho^{(p)}\boldsymbol{s}_M[k] \end{bmatrix}$$

CentraleSupélec