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

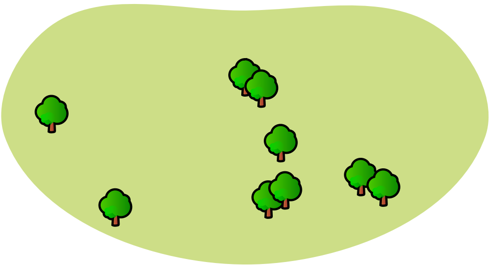
https://bit.ly/3g3S6X4

45 minutes !!!!

Good afternoon, thank you all for being here.I'm Rafael Accácio and I'm going to present my work on the security of distributed model predictive control under false data injection.

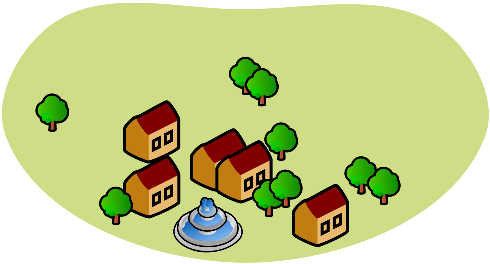
Context

Requirements evolve with time



These cyberphysical systems are the majority of the systems in our everyday lives.We can give example the traffic management,water distribution,electricity distribution, heat and cold and many more.But how to control those kinds of systems. Each has its own Dynamics and constraints, such comfort (Quality of service) or technical. Solution, mpc since we use models and it is easy to integrate the constraints.

Requirements evolve with time



These cyberphysical systems are the majority of the systems in our everyday lives.We can give example the traffic management,water distribution,electricity distribution, heat and cold and many more.But how to control those kinds of systems. Each has its own Dynamics and constraints, such comfort (Quality of service) or technical. Solution, mpc since we use models and it is easy to integrate the constraints.

Context

Requirements evolve with time

These cyberphysical systems are the majority of the systems in our everyday lives.We can give example the traffic management,water distribution,electricity distribution, heat and cold and many more.But how to control those kinds of systems. Each has its own Dynamics and constraints, such comfort (Quality of service) or technical. Solution, mpc since we use models and it is easy to integrate the constraints.

CentraleSupélec

Context

Requirements evolve with time

These cyberphysical systems are the majority of the systems in our everyday lives.We can give example the traffic management,water distribution,electricity distribution, heat and cold and many more.But how to control those kinds of systems. Each has its own Dynamics and constraints, such comfort (Quality of service) or technical. Solution, mpc since we use models and it is easy to integrate the constraints.

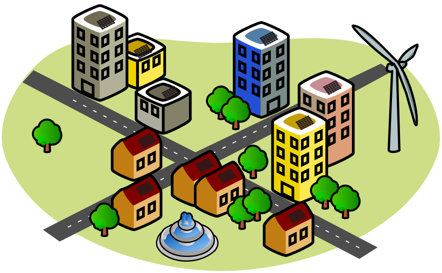
Context

Requirements evolve with time

CentraleSupélec

The systems are usually Geographically distributed These cyberphysical systems are the majority of the systems in our everyday lives.We can give example the traffic management,water distribution,electricity distribution, heat and cold and many more.But how to control those kinds of systems. Each has its own Dynamics and constraints, such comfort (Quality of service) or technical. Solution, mpc since we use models and it is easy to integrate the constraints.

Requirements evolve with time



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

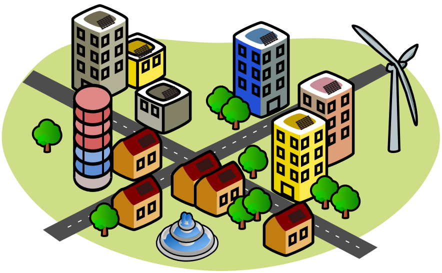
  (include your problem here)

CentraleSupélec

The systems are usually Geographically distributed Coupled by constraints as maximum input power or energy These cyberphysical systems are the majority of the systems in our everyday lives.We can give example the traffic management,water distribution,electricity distribution, heat and cold and many more.But how to control those kinds of systems. Each has its own Dynamics and constraints, such comfort (Quality of service) or technical. Solution, mpc since we use models and it is easy to integrate the constraints.

Context

Requirements evolve with time



- Electricity Distribution System
- Heat distribution
- Water distribution
- Traffic management
  (include your problem here)

The systems are usually Geographically distributed Coupled by constraints as maximum input power or energy and when they are implemented we try to optimize objectives such as cost, energy, user satisfaction and others.
These cyberphysical systems are the majority of the systems in our everyday lives.We can give example the traffic management,water distribution,electricity distribution, heat and cold and many more.But how to control those kinds of systems. Each has its own Dynamics and constraints, such comfort (Quality of service) or technical. Solution, mpc since we use models and it is easy to integrate the constraints.

Context

Requirements evolve with time

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

(include your problem here)

The systems are usually Geographically distributed Coupled by constraints as maximum input power or energy and when they are implemented we try to optimize objectives such as cost, energy, user satisfaction and others.
A solution is to use MPCThese cyberphysical systems are the majority of the systems in our everyday lives.We can give example the traffic management,water distribution,electricity distribution, heat and cold and many more.But how to control those kinds of systems. Each has its own Dynamics and constraints, such comfort (Quality of service) or technical. Solution, mpc since we use models and it is easy to integrate the constraints.

Context

Requirements evolve with time

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

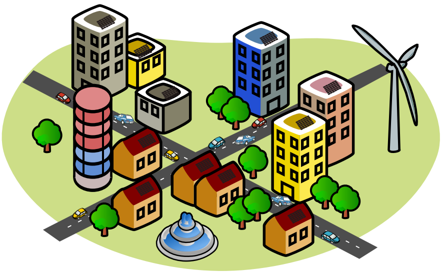
  (include your problem here)

CentraleSupélec

The systems are usually Geographically distributed Coupled by constraints as maximum input power or energy and when they are implemented we try to optimize objectives such as cost, energy, user satisfaction and others.
These cyberphysical systems are the majority of the systems in our everyday lives.We can give example the traffic management,water distribution,electricity distribution, heat and cold and many more.But how to control those kinds of systems. Each has its own Dynamics and constraints, such comfort (Quality of service) or technical. Solution, mpc since we use models and it is easy to integrate the constraints.

Context

Requirements evolve with time



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

The systems are usually Geographically distributed Coupled by constraints as maximum input power or energy and when they are implemented we try to optimize objectives such as cost, energy, user satisfaction and others.
These cyberphysical systems are the majority of the systems in our everyday lives.We can give example the traffic management,water distribution,electricity distribution, heat and cold and many more.But how to control those kinds of systems. Each has its own Dynamics and constraints, such comfort (Quality of service) or technical. Solution, mpc since we use models and it is easy to integrate the constraints.

Context

Requirements evolve with time

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

The systems are usually Geographically distributed Coupled by constraints as maximum input power or energy and when they are implemented we try to optimize objectives such as cost, energy, user satisfaction and others.
These cyberphysical systems are the majority of the systems in our everyday lives.We can give example the traffic management,water distribution,electricity distribution, heat and cold and many more.But how to control those kinds of systems. Each has its own Dynamics and constraints, such comfort (Quality of service) or technical. Solution, mpc since we use models and it is easy to integrate the constraints.

Requirements evolve with time

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

CentraleSupélec

The systems are usually Geographically distributed Coupled by constraints as maximum input power or energy and when they are implemented we try to optimize objectives such as cost, energy, user satisfaction and others.
These cyberphysical systems are the majority of the systems in our everyday lives.We can give example the traffic management,water distribution,electricity distribution, heat and cold and many more.But how to control those kinds of systems. Each has its own Dynamics and constraints, such comfort (Quality of service) or technical. Solution, mpc since we use models and it is easy to integrate the constraints.

Context

Requirements evolve with time

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

The systems are usually Geographically distributed Coupled by constraints as maximum input power or energy and when they are implemented we try to optimize objectives such as cost, energy, user satisfaction and others.
These cyberphysical systems are the majority of the systems in our everyday lives.We can give example the traffic management,water distribution,electricity distribution, heat and cold and many more.But how to control those kinds of systems. Each has its own Dynamics and constraints, such comfort (Quality of service) or technical. Solution, mpc since we use models and it is easy to integrate the constraints.

Context

Requirements evolve with time

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

The systems are usually Geographically distributed Coupled by constraints as maximum input power or energy and when they are implemented we try to optimize objectives such as cost, energy, user satisfaction and others.
These cyberphysical systems are the majority of the systems in our everyday lives.We can give example the traffic management,water distribution,electricity distribution, heat and cold and many more.But how to control those kinds of systems. Each has its own Dynamics and constraints, such comfort (Quality of service) or technical. Solution, mpc since we use models and it is easy to integrate the constraints.

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

The systems are usually Geographically distributed Coupled by constraints as maximum input power or energy and when they are implemented we try to optimize objectives such as cost, energy, user satisfaction and others.
These cyberphysical systems are the majority of the systems in our everyday lives.We can give example the traffic management,water distribution,electricity distribution, heat and cold and many more.But how to control those kinds of systems. Each has its own Dynamics and constraints, such comfort (Quality of service) or technical. Solution, mpc since we use models and it is easy to integrate the constraints.

Context

Requirements evolve with time

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

The systems are usually Geographically distributed Coupled by constraints as maximum input power or energy and when they are implemented we try to optimize objectives such as cost, energy, user satisfaction and others.
These cyberphysical systems are the majority of the systems in our everyday lives.We can give example the traffic management,water distribution,electricity distribution, heat and cold and many more.But how to control those kinds of systems. Each has its own Dynamics and constraints, such comfort (Quality of service) or technical. Solution, mpc since we use models and it is easy to integrate the constraints.

Context

Requirements evolve with time

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

The systems are usually Geographically distributed Coupled by constraints as maximum input power or energy and when they are implemented we try to optimize objectives such as cost, energy, user satisfaction and others.
These cyberphysical systems are the majority of the systems in our everyday lives.We can give example the traffic management,water distribution,electricity distribution, heat and cold and many more.But how to control those kinds of systems. Each has its own Dynamics and constraints, such comfort (Quality of service) or technical. Solution, mpc since we use models and it is easy to integrate the constraints.

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

For those who are not familiar with mpc. Mpc is the model based predictive controller.

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

The objective is to find the best control sequence using predictions based on a model.

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

When we say best,

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, ...)

we mean optimal.

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, ...)

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

CentraleSupélec

So we need to solve an optimization problem.

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, ...)

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

CentraleSupélec

And we have the control sequence of u as the decision variable.

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, ...)

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

CentraleSupélec

which is calculated for a horizon N

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

So, we need an objective function. For example follow a trajectory while minimizing the energy.

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

A model of the system

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

with its states

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, ...)

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

CentraleSupélec

and inputs

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, ...)

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

CentraleSupélec

But we can also integrate some constraints, such QoS or technical restrictions

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, ...)

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

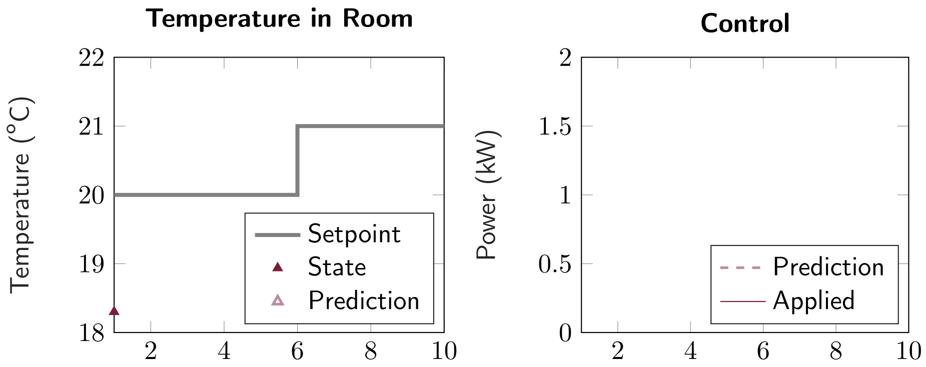
CentraleSupélec

But we can also integrate some constraints, such QoS or technical restrictions

Model Predictive Control

In a nutshell

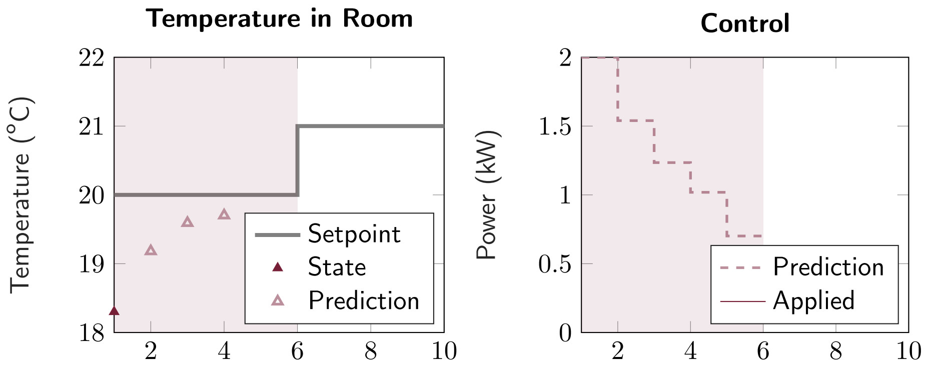
Find optimal control sequence



CentraleSupélec

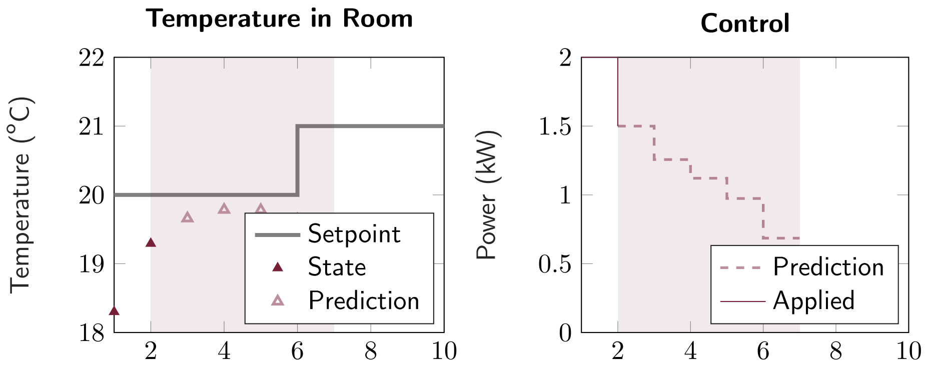
So, for example, if we may have a setpoint to follow

Find optimal control sequence



We find an optimal control sequence

Find optimal control sequence, apply first element

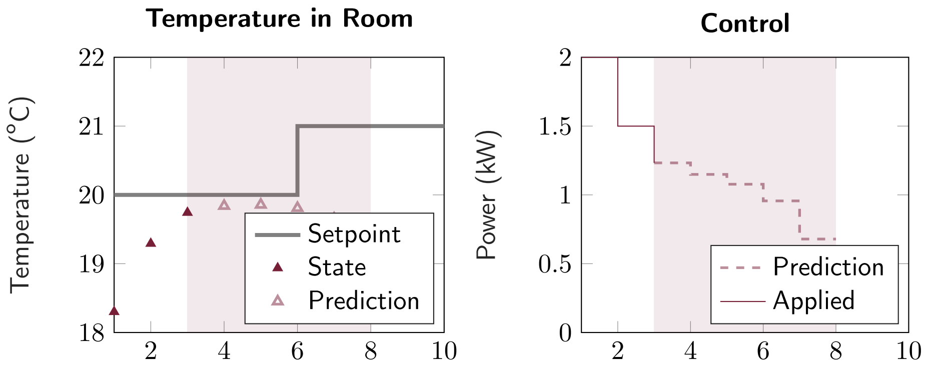


We apply only the first element

Model Predictive Control

In a nutshell

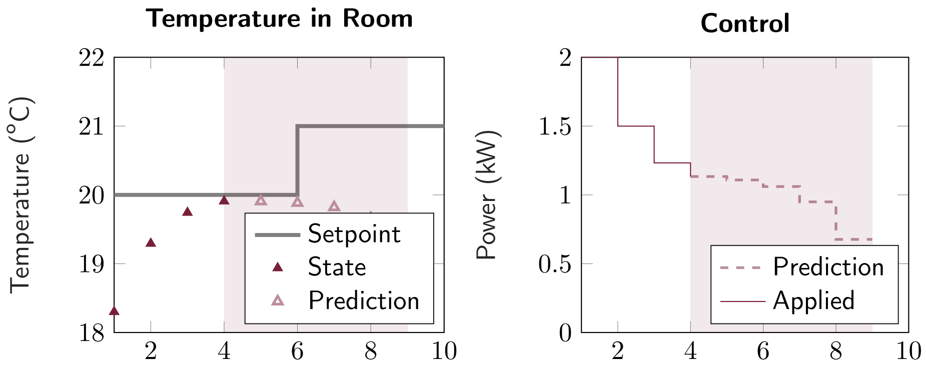
Find optimal control sequence, apply first element, rinse repeat



CentraleSupélec

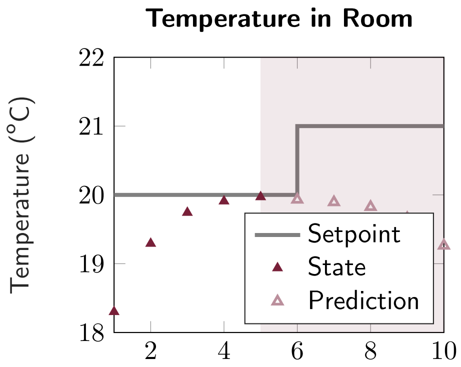
and then we repeat

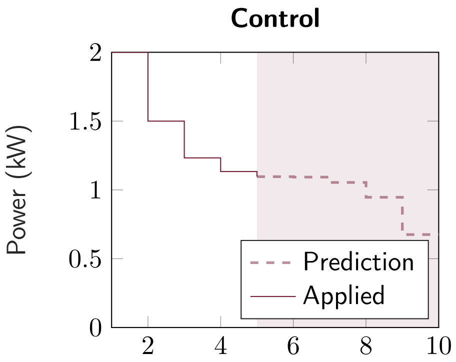
Find optimal control sequence, apply first element, rinse repeat → Receding Horizon



following what we call the receding horizon strategy. However this problem is not always straightforward to solve, for some cases it can be easier.

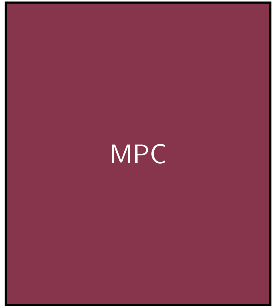
Find optimal control sequence, apply first element, rinse repeat → Receding Horizon





following what we call the receding horizon strategy. However this problem is not always straightforward to solve, for some cases it can be easier.

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

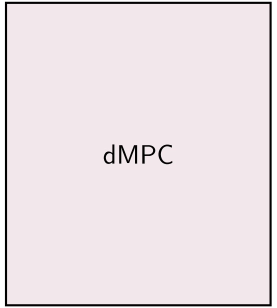


1

CentraleSupélec

However, the solution will depend on the horizon, the number of constraints, and sizes of input and states, increasing the complexity of the calculation

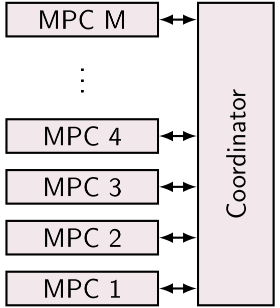
- Problem: Complexity depends on N, m, p and sizes of $\boldsymbol{x}$ and $\boldsymbol{u}$
- Solution: Divide and Conquer[1]



[1] *Distributed Model Predictive Control made easy*

A strategy to alleviate is to distribute the calculation whenever possible. And there are many ways to divide it as the book shows.

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

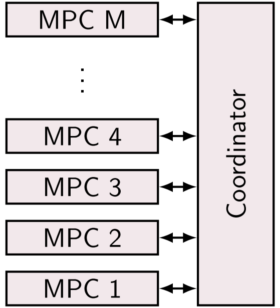


[1] *Distributed Model Predictive Control made easy*

CentraleSupélec

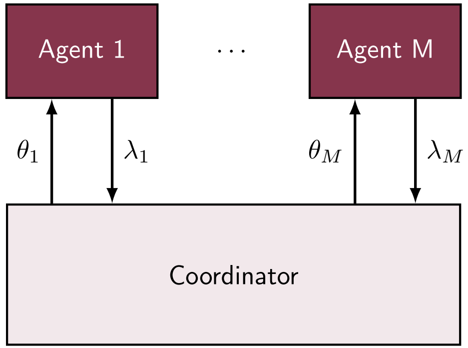
Here we opt for a hierarchical strategy where we use multiple MPCs and an agent to coordinate and manage the coupling aspects of the problem.

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



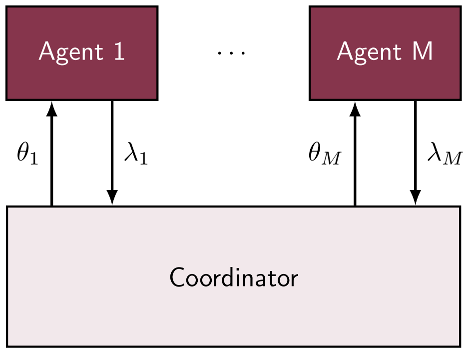
[1] *Distributed Model Predictive Control made easy*

Here we opt for a hierarchical strategy where we use multiple MPCs and an agent to coordinate and manage the coupling aspects of the problem.



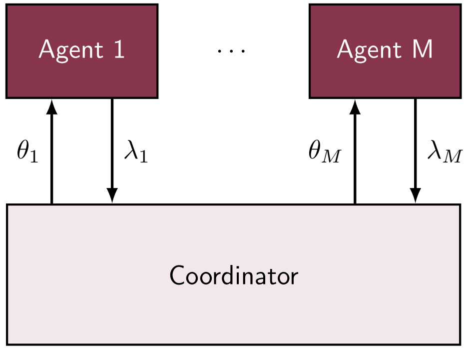
- Agents solve local problems
- Variables are updated

$\Big\}$ Until Convergence



- Agents solve local problems
- Variables are updated

Until Convergence



- Agents solve local problems
- Variables are updated

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?

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?

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

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

To respond this this presentation is divided into 3 parts.First we present the decomposition and its vulnerabilites,

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

To respond this this presentation is divided into 3 parts.First we present the decomposition and its vulnerabilites, We propose a resilient method for two kind of systems with increasing complexities.

Outline

To respond this this presentation is divided into 3 parts.First we present the decomposition and its vulnerabilites, We propose a resilient method for two kind of systems with increasing complexities.

Outline

CentraleSupélec

Distributed Model Predictive Control

Primal Decomposition | Quantity Decomposition | Resource Allocation

Decompose original problem using primal problem

An example of decomposition method is the Quantity decomposition where a semi-decomposable problem with a global coupling constraints can be decomposed into

Distributed Model Predictive Control

Decompose original problem using primal problem

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

CentraleSupélec

An example of decomposition method is the Quantity decomposition where a semi-decomposable problem with a global coupling constraints can be decomposed into multiple sub-problems, which can be solved in parallel, and a master problem which corresponds to the initial problem. Those coupling constraints are replaced by local constraints with an allocation theta i.

Distributed Model Predictive Control

Primal Decomposition | Quantity Decomposition | Resource Allocation

Decompose original problem using primal problem

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

CentraleSupélec

An example of decomposition method is the Quantity decomposition where a semi-decomposable problem with a global coupling constraints can be decomposed into multiple sub-problems, which can be solved in parallel, and a master problem which corresponds to the initial problem. Those coupling constraints are replaced by local constraints with an allocation theta i. The allocation for each sub-problem is updated by a projected subgradient method solving the master problem, thus the original problem. The subgradient used in this method is the dual variable associated to the coupling constraints

Distributed Model Predictive Control

Decompose original problem using primal problem E.g. $v_i = w_i - x_i$

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

CentraleSupélec

An example of decomposition method is the Quantity decomposition where a semi-decomposable problem with a global coupling constraints can be decomposed into multiple sub-problems, which can be solved in parallel, and a master problem which corresponds to the initial problem.
Those coupling constraints are replaced by local constraints with an allocation theta i. The allocation for each sub-problem is updated by a projected subgradient method solving the master problem, thus the original problem. The subgradient used in this method is the dual variable associated to the coupling constraintsWe can represent the sub-problems as agents and the master problem as a coordinator. The coordinator sends the allocations for each agent, which responds with a dual variable.

Distributed Model Predictive Control

Primal Decomposition | Quantity Decomposition | Resource Allocation

Decompose original problem using primal problem

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

CentraleSupélec

An example of decomposition method is the Quantity decomposition where a semi-decomposable problem with a global coupling constraints can be decomposed into multiple sub-problems, which can be solved in parallel, and a master problem which corresponds to the initial problem.
Those coupling constraints are replaced by local constraints with an allocation theta i. The allocation for each sub-problem is updated by a projected subgradient method solving the master problem, thus the original problem. The subgradient used in this method is the dual variable associated to the coupling constraintsWe can represent the sub-problems as agents and the master problem as a coordinator. The coordinator sends the allocations for each agent, which responds with a dual variable.

Distributed Model Predictive Control

Primal Decomposition | Quantity Decomposition | Resource Allocation

Decompose original problem using primal problem

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

CentraleSupélec

An example of decomposition method is the Quantity decomposition where a semi-decomposable problem with a global coupling constraints can be decomposed into multiple sub-problems, which can be solved in parallel, and a master problem which corresponds to the initial problem. Those coupling constraints are replaced by local constraints with an allocation theta i. The allocation for each sub-problem is updated by a projected subgradient method solving the master problem, thus the original problem. The subgradient used in this method is the dual variable associated to the coupling constraintsWe can represent the sub-problems as agents and the master problem as a coordinator. The coordinator sends the allocations for each agent, which responds with a dual variable.

Distributed Model Predictive Control

Primal Decomposition | Quantity Decomposition | Resource Allocation

Decompose original problem using primal problem

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

CentraleSupélec

An example of decomposition method is the Quantity decomposition where a semi-decomposable problem with a global coupling constraints can be decomposed into multiple sub-problems, which can be solved in parallel, and a master problem which corresponds to the initial problem.
Those coupling constraints are replaced by local constraints with an allocation theta i. The allocation for each sub-problem is updated by a projected subgradient method solving the master problem, thus the original problem. The subgradient used in this method is the dual variable associated to the coupling constraintsWe can represent the sub-problems as agents and the master problem as a coordinator. The coordinator sends the allocations for each agent, which responds with a dual variable.

Distributed Model Predictive Control

Primal Decomposition | Quantity Decomposition | Resource Allocation

Decompose original problem using primal problem

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

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

CentraleSupélec

An example of decomposition method is the Quantity decomposition where a semi-decomposable problem with a global coupling constraints can be decomposed into multiple sub-problems, which can be solved in parallel, and a master problem which corresponds to the initial problem.
Those coupling constraints are replaced by local constraints with an allocation theta i. The allocation for each sub-problem is updated by a projected subgradient method solving the master problem, thus the original problem. The subgradient used in this method is the dual variable associated to the coupling constraintsWe can represent the sub-problems as agents and the master problem as a coordinator. The coordinator sends the allocations for each agent, which responds with a dual variable.

Distributed Model Predictive Control

Primal Decomposition | Quantity Decomposition | Resource Allocation

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

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

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

CentraleSupélec

An example of decomposition method is the Quantity decomposition where a semi-decomposable problem with a global coupling constraints can be decomposed into multiple sub-problems, which can be solved in parallel, and a master problem which corresponds to the initial problem.
Those coupling constraints are replaced by local constraints with an allocation theta i. The allocation for each sub-problem is updated by a projected subgradient method solving the master problem, thus the original problem. The subgradient used in this method is the dual variable associated to the coupling constraintsWe can represent the sub-problems as agents and the master problem as a coordinator. The coordinator sends the allocations for each agent, which responds with a dual variable.

Distributed Model Predictive Control

Primal Decomposition | Quantity Decomposition | Resource Allocation

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

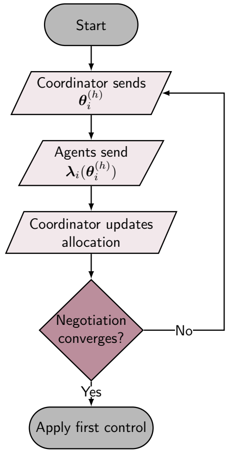
$$\begin{aligned}
\underset{\boldsymbol{U}_1[k]}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{U}_1[k]\|_{H_1}^2 + \boldsymbol{f}_1[k]^T \boldsymbol{U}_1[k] \\
\text{subject to} \quad & \bar{\Gamma}_1 \boldsymbol{U}_1[k] \preceq \boldsymbol{\theta}_1[k] : \boldsymbol{\lambda}_1[k] \\
& \vdots \\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

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

CentraleSupélec

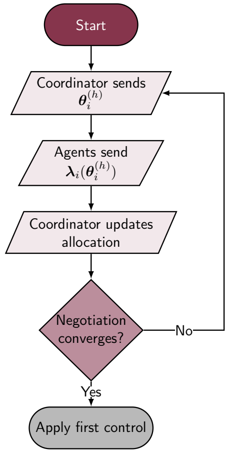
An example of decomposition method is the Quantity decomposition where a semi-decomposable problem with a global coupling constraints can be decomposed into multiple sub-problems, which can be solved in parallel, and a master problem which corresponds to the initial problem. Those coupling constraints are replaced by local constraints with an allocation theta i. The allocation for each sub-problem is updated by a projected subgradient method solving the master problem, thus the original problem. The subgradient used in this method is the dual variable associated to the coupling constraintsWe can represent the sub-problems as agents and the master problem as a coordinator. The coordinator sends the allocations for each agent, which responds with a dual variable.

Quantity Decomposition | Resource Allocation



In a flowchart for a quantity decomposition based DMPC,

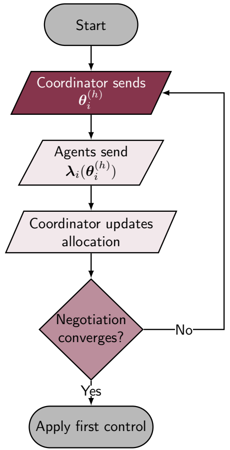
Quantity Decomposition | Resource Allocation



CentraleSupélec

In a flowchart for a quantity decomposition based DMPC,

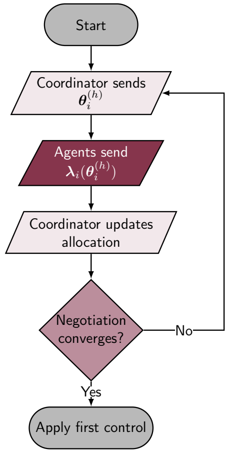
Quantity Decomposition | Resource Allocation



CentraleSupélec

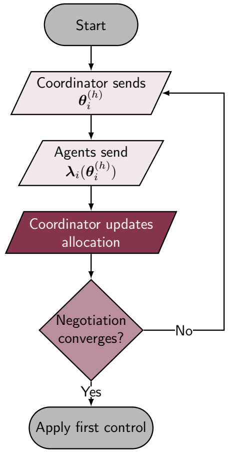
In a flowchart for a quantity decomposition based DMPC,the coordinator sends the allocation theta,

Quantity Decomposition | Resource Allocation



In a flowchart for a quantity decomposition based DMPC,the coordinator sends the allocation theta, the agents send the dual variable lambda,

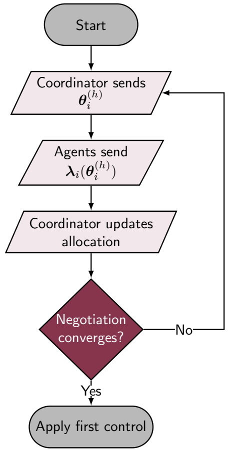
Quantity Decomposition | Resource Allocation



CentraleSupélec

In a flowchart for a quantity decomposition based DMPC,the coordinator sends the allocation theta, the agents send the dual variable lambda, the coordinator updates the allocation.

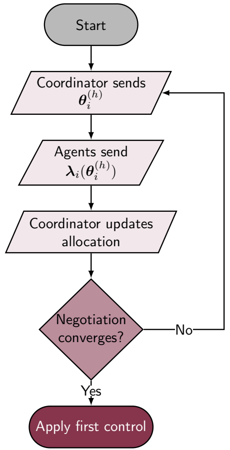
Quantity Decomposition | Resource Allocation



CentraleSupélec

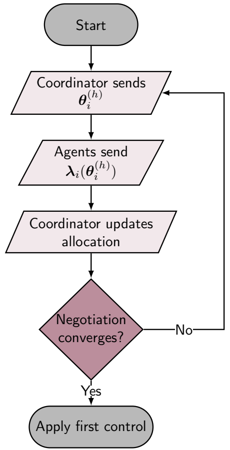
In a flowchart for a quantity decomposition based DMPC,the coordinator sends the allocation theta, the agents send the dual variable lambda, the coordinator updates the allocation. If negotiation converges,

Quantity Decomposition | Resource Allocation



In a flowchart for a quantity decomposition based DMPC,the coordinator sends the allocation theta, the agents send the dual variable lambda, the coordinator updates the allocation. If negotiation converges, then the negotiation ends and each agent applies the first element of the control found

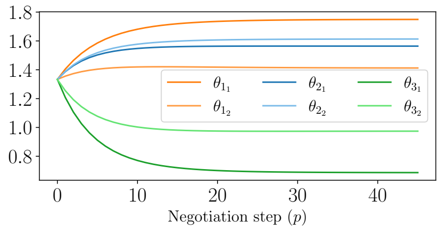
Quantity Decomposition | Resource Allocation

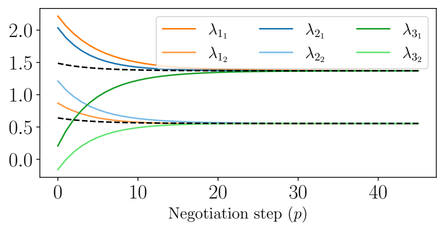


CentraleSupélec

In a flowchart for a quantity decomposition based DMPC,the coordinator sends the allocation theta, the agents send the dual variable lambda, the coordinator updates the allocation. If negotiation converges, then the negotiation ends and each agent applies the first element of the control found

Quantity Decomposition | Resource Allocation





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)

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)

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)

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)

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)

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)

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)

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 $\boldsymbol{\lambda}_i$

CentraleSupélec

How can an agent attack the system? Its only interface with the coordination is lambda

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 an agent attack the system? Its only interface with the coordination is lambdaso we suppose it attacks by sending a different $lambda_i$, modified by a function $gamma_i$ of $lambda_i$

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 an agent attack the system? Its only interface with the coordination is lambdaso we suppose it attacks by sending a different $lambda_i$, modified by a function $gamma_i$ of $lambda_i$

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 $\boldsymbol{\lambda}_i$

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

CentraleSupélec

How can an agent attack the system? Its only interface with the coordination is lambdaso we suppose it attacks by sending a different $lambda_i$, modified by a function $gamma_i$ of $lambda_i$

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 $\boldsymbol{\lambda}_i$

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

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

How can an agent attack the system? Its only interface with the coordination is lambdaso we suppose it attacks by sending a different $lambda_i$, modified by a function $gamma_i$ of $lambda_i$

Example

4 distinct agents

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

We give an example of 4 agents negotiating

Example

4 distinct agents

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

We give an example of 4 agents negotiating Agent 1 is non-cooperative

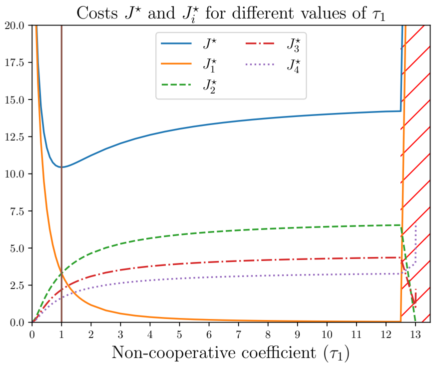
Example

4 distinct agents

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

We give an example of 4 agents negotiating Agent 1 is non-cooperative
It uses a linear cheating function gamma tau times identity times the original lambda i
In the figure we can see the cost functions for each agent, we see that agent 1 cost decreases if we increase tau, but the overall cost is increased, The minimum value of the global cost is when

Example



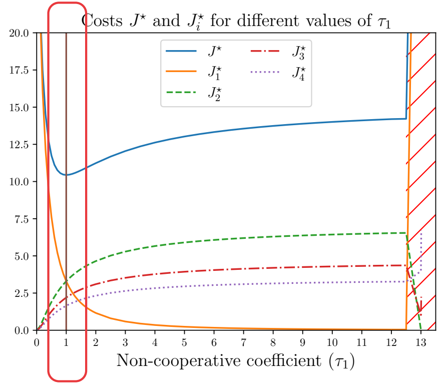
4 distinct agents

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

CentraleSupélec

We give an example of 4 agents negotiating Agent 1 is non-cooperative
It uses a linear cheating function gamma tau times identity times the original lambda i
In the figure we can see the cost functions for each agent, we see that agent 1 cost decreases if we increase tau, but the overall cost is increased, The minimum value of the global cost is when tau is equal to one. If tau is close to zero it increases its cost and if tau is too high

Example

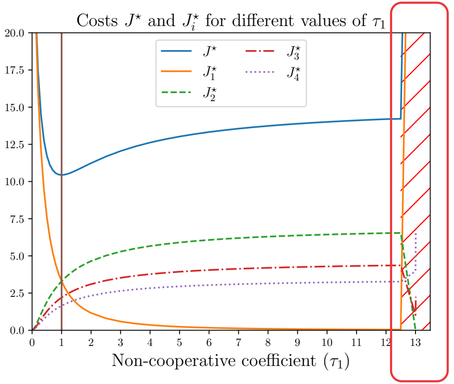


4 distinct agents

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

We give an example of 4 agents negotiating Agent 1 is non-cooperative
It uses a linear cheating function gamma tau times identity times the original lambda i
In the figure we can see the cost functions for each agent, we see that agent 1 cost decreases if we increase tau, but the overall cost is increased, The minimum value of the global cost is when tau is equal to one. If tau is close to zero it increases its cost and if tau is too high it can turn the negotiation unstable

Example



4 distinct agents

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

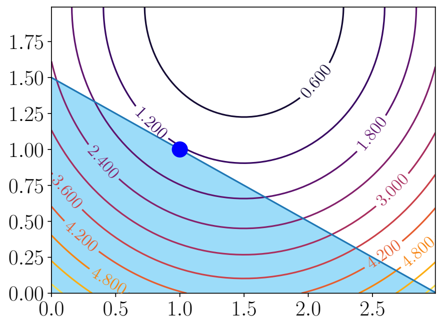
We give an example of 4 agents negotiating Agent 1 is non-cooperative
It uses a linear cheating function gamma tau times identity times the original lambda i
In the figure we can see the cost functions for each agent, we see that agent 1 cost decreases if we increase tau, but the overall cost is increased, The minimum value of the global cost is when tau is equal to one. If tau is close to zero it increases its cost and if tau is too high
it can turn the negotiation unstable

Mitigating

There are vulnerabilities

Mitigating

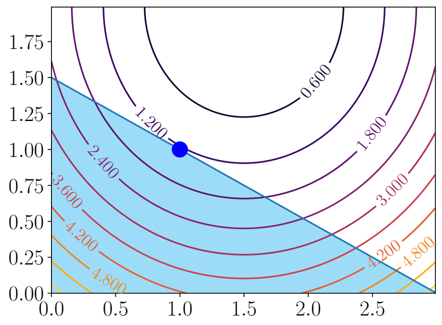
There are vulnerabilities



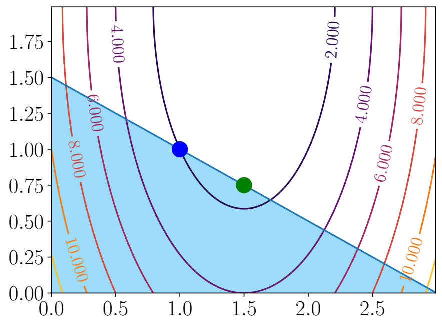
Original minimum.

Mitigating

There are vulnerabilities



Original minimum.



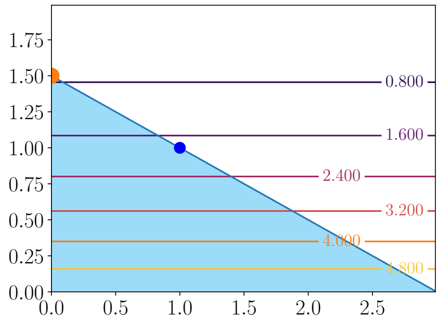
Minimum after attack.

Mitigating

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

Mitigating

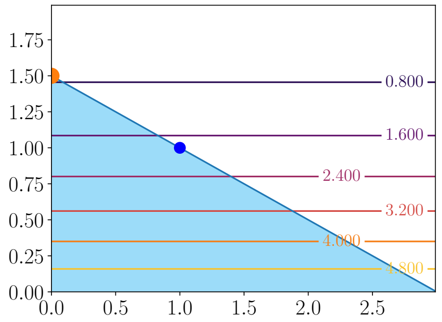
There are vulnerabilities
"Great". But what can we do?



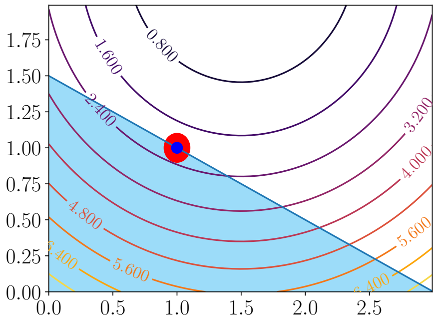
Ignore attacker.

Mitigating

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



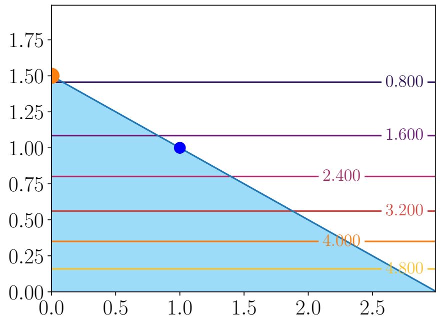
Ignore attacker.



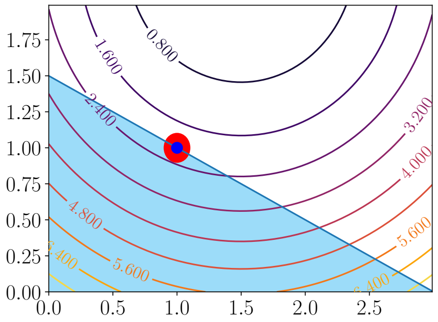
Recover original behavior.

Mitigating

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



Ignore attacker.



Recover original behavior.

Outline

2. Resilient Primal Decomposition-based dMPC for deprived systems

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

CentraleSupélec

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

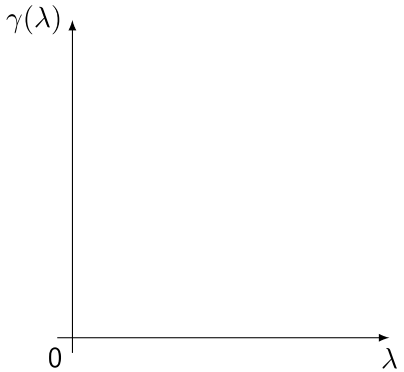
- Recover original behavior
  - Invert effects of function $\gamma_i(\boldsymbol{\lambda}_i)$
- Is $\gamma_i(\boldsymbol{\lambda}_i)$ invertible?
  - Not necessarily, but let's reason

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



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

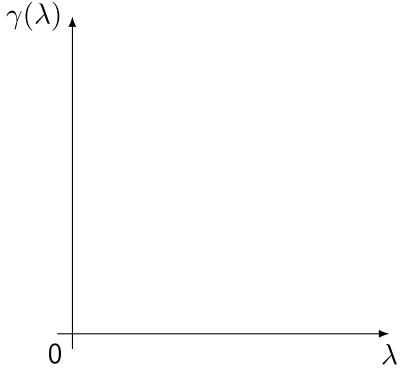
Assumptions

- 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



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

Assumptions

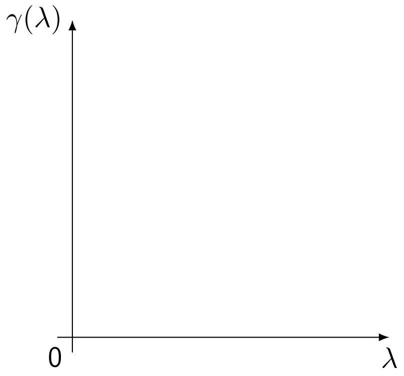
- Attacker is greedy $\gamma(\lambda) > \lambda$
- But not smart

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

CentraleSupélec

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

Unidimensional Case



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

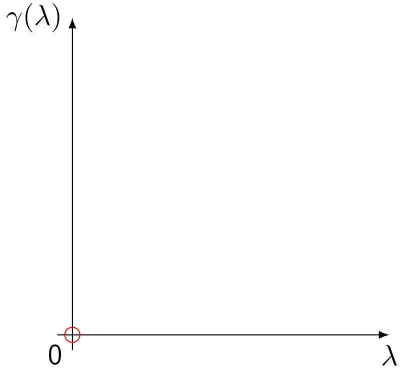
Assumptions

- 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



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

Assumptions

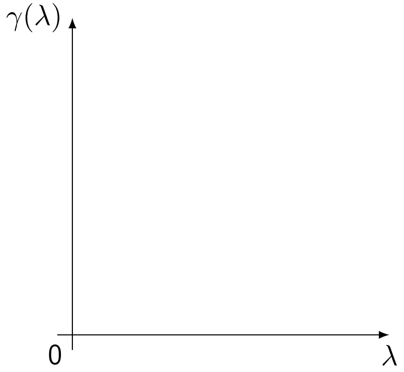
- Attacker is greedy $\gamma(\lambda) > \lambda$
- But not smart

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

CentraleSupélec

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

Unidimensional Case



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

Assumptions

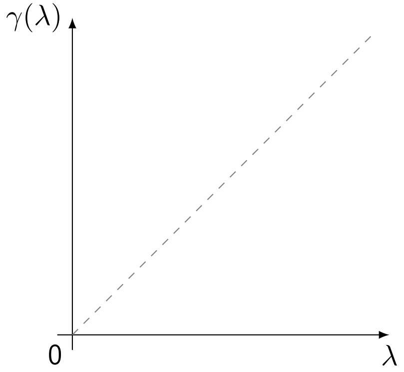
- Attacker is greedy $\gamma(\lambda) > \lambda$
- But not smart

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

CentraleSupélec

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

Unidimensional Case



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

Assumptions

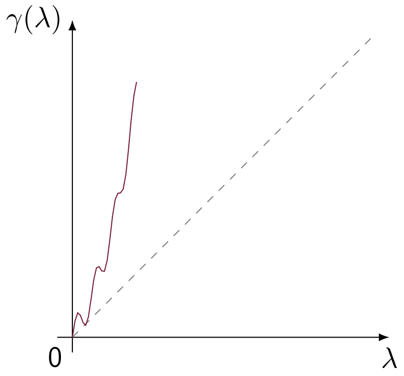
- Attacker is greedy $\gamma(\lambda) > \lambda$
- But not smart

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

CentraleSupélec

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

Unidimensional Case



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

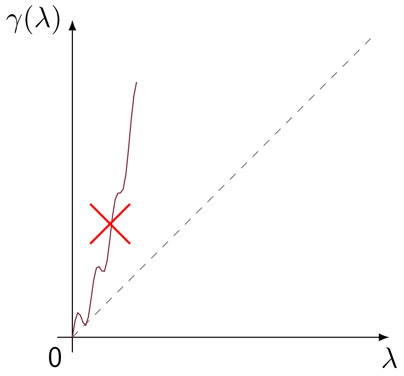
Assumptions

- 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



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

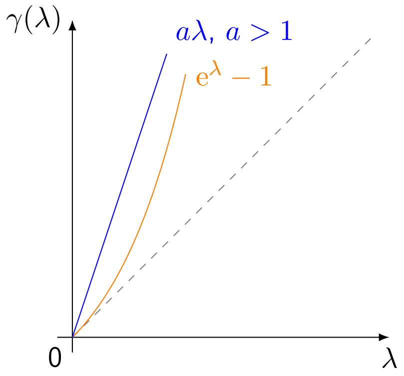
Assumptions

- 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



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

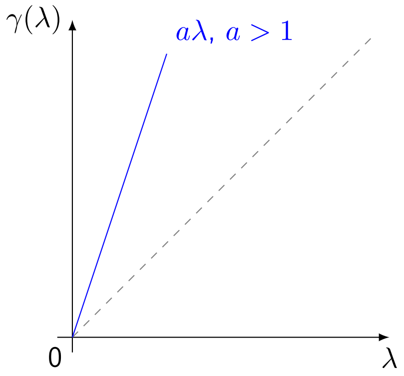
Assumptions

- 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



- $\lambda \geq 0$ 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?

Linear Multidimensional Case

- $\gamma(\lambda) = T\lambda$
- $T\lambda = 0 \Leftrightarrow \lambda = 0 \rightarrow \exists T^{-1}$

CentraleSupélec

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

Linear Multidimensional Case

- $\gamma(\lambda) = T\lambda$
- $T\lambda = 0 \Leftrightarrow \lambda = 0 \rightarrow \exists T^{-1}$

CentraleSupélec

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

Linear Multidimensional Case

- $\gamma(\lambda) = T\lambda$
- $T\lambda = 0 \Leftrightarrow \lambda = 0 \rightarrow \exists T^{-1}$

CentraleSupélec

Strategy

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

Strategy

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

Strategy

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

Strategy

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

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 $\bar{\Gamma}_i\mathring{\boldsymbol{U}}_i^\star[k] \succ \boldsymbol{\theta}_i[k], \forall k$
  - Solution projected onto boundaries (equality constraints)

CentraleSupélec

Deprived Systems

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

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

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 $\bar{\Gamma}_i\mathring{\boldsymbol{U}}_i^\star[k] \succ \boldsymbol{\theta}_i[k], \forall k$
  - Solution projected onto boundaries (equality constraints)

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 $\bar{\Gamma}_i\mathring{\boldsymbol{U}}_i^\star[k] \succ \boldsymbol{\theta}_i[k], \forall k$
  - Solution projected onto boundaries (equality constraints)

CentraleSupélec

Deprived Systems

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

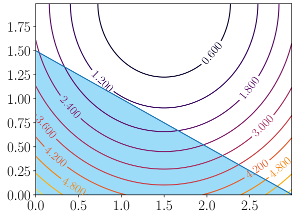
- Unconstrained solution $\mathring{\boldsymbol{U}}_i^\star[k] = -H_i^{-1}\boldsymbol{f}_i[k]$
- Deprived if $\bar{\Gamma}_i \mathring{\boldsymbol{U}}_i^\star[k] \succ \boldsymbol{\theta}_i[k], \forall k$
  - Solution projected onto boundaries (equality constraints)

CentraleSupélec

Deprived Systems

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

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

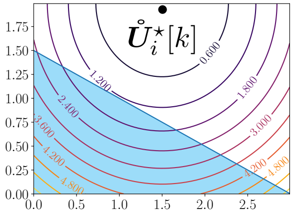


CentraleSupélec

Deprived Systems

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

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

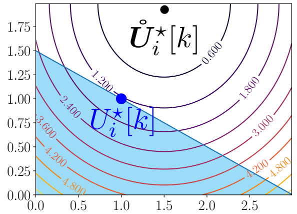


CentraleSupélec

Deprived Systems

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

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



CentraleSupélec

Deprived Systems

But why?

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

Deprived Systems

But why?

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

CentraleSupélec

Deprived Systems

But why?

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

CentraleSupélec

Deprived Systems

But why?

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

Deprived Systems

But why?

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

CentraleSupélec

Deprived Systems

But why?

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

Deprived Systems

But why?

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

Deprived Systems

But why?

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

CentraleSupélec

Deprived Systems

Analysis

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

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

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

where $P_i = (\bar{\Gamma}_i H_i^{-1}\bar{\Gamma}_i^T)^{-1}$ 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 $P_i = (\bar{\Gamma}_i H_i^{-1} \bar{\Gamma}_i^T)^{-1}$ 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

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

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

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

[2]Under some conditions, see here

CentraleSupélec

Outline

3. Resilient Primal Decomposition-based dMPC using Artificial Scarcity

CentraleSupélec

Thank you!

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

Contact
rafael.accacio.nogueira@gmail.com

If you want to see the simulations of this paper we have a github repository, and if you want to send me an email about this paper or this presentation you can flash the QR code in the right. Thank you!

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.

As a recommended reading I give a book about distributed MPC and an article with another secure dmpc algorithm based in another decomposition method. That's all

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.

CentraleSupélec