Security of distributed Model Predictive Control under False Data Injection

Rafael Accácio NOGUEIRA
rafael.accacio.nogueira@gmail.com

Seminar
École Centrale de Lyon / Laboratoire Ampère
26/05/2023 @ Écully

https://bit.ly/3g3S6X4

About me

Rafael Accácio Nogueira

Postdoctoral researcher at LAAS/CNRS

Garanteed relative localisation and anticollision scenario for autonomous vehicles

Project AutOCampus (GIS neOCampus)

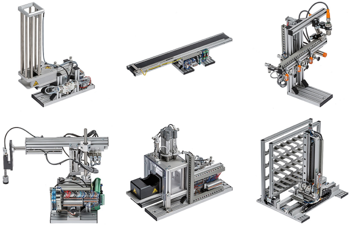
Advised by Soheib Fergani

About me

Bachelor Thesis at Escola Politécnica/UFRJ

Identification of DES for fault-diagnosis

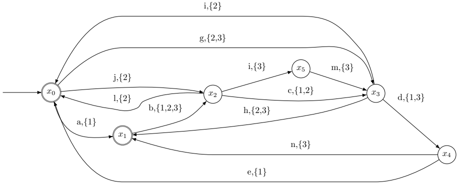
Advised by Marcos Vicente de Brito Moreira



I/O

+

DAOCT



About me

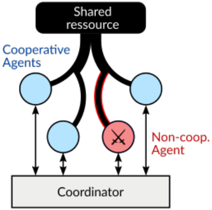
Doctoral Thesis at CentraleSupélec/IETR

Security of dMPC under False Data Injection

Advised by Hervé Guéguen and Romain Bourdais

CentraleSupélec

IETR



Context

Smart(er) Cities

Multiple systems interacting

Context

Smart(er) Cities

Multiple systems interacting

Context

Smart(er) Cities

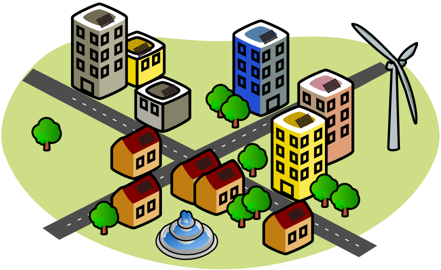
Multiple systems interacting

- Distribution:

Context

Smart(er) Cities

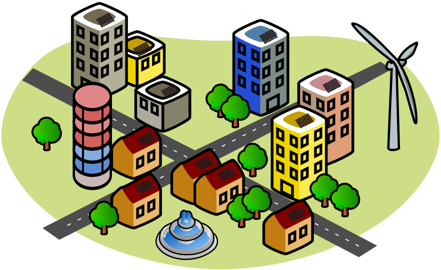
Multiple systems interacting



- Distribution:
    - Electricity

Smart(er) Cities

Multiple systems interacting



- Distribution:
  - Electricity
  - Heat
  - Water

Context

Smart(er) Cities

Multiple systems interacting

- Distribution:
  - Electricity
  - Heat
  - Water
- Traffic

Multiple systems interacting

- Distribution:
  - Electricity
  - Heat
  - Water
- Traffic

...

Context

Smart(er) Cities

Multiple systems interacting under

- Technical/Comfort Constraints

Context

Smart(er) Cities

Multiple systems interacting under

- Technical/Comfort Constraints
- We also want

Context

Smart(er) Cities

Multiple systems interacting under

- Technical/Comfort Constraints
- We also want
  - Minimize consumption

Smart(er) Cities

Multiple systems interacting under

- Technical/Comfort Constraints
- We also want
    - Minimize consumption
    - Maximizer satisfaction

Context

Smart(er) Cities

Multiple systems interacting under

- Technical/Comfort Constraints
- We also want
  - Minimize consumption
  - Maximizer satisfaction
  - Follow a trajectory

Smart(er) Cities

Multiple systems interacting under

- Technical/Comfort Constraints
- We also want
  - Minimize consumption
  - Maximizer satisfaction
  - Follow a trajectory
- Solution → MPC

Model-based Predictive Control

Brief recap

Model-based Predictive Control

Brief recap

Find best control sequence using predictions based on a model.

Model-based Predictive Control

Brief recap

Find best control sequence using predictions based on a model.

Model-based Predictive Control

Brief recap

Find optimal control sequence using predictions based on a model.

Model-based Predictive Control

Brief recap

Find optimal control sequence using predictions based on a model.

- We need an optimization problem

$$\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k])$$

Model-based Predictive Control

Brief recap

Find optimal control sequence using predictions based on a model.

- We need an optimization problem
  - Decision variable is the control sequence

$$\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k])$$

Model-based Predictive Control

Brief recap

Find optimal control sequence using predictions based on a model.

- We need an optimization problem
  - Decision variable is the control sequence calculated over horizon N

$$\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k])$$

Model-based Predictive Control

Brief recap

Find optimal control sequence using predictions based on a model.

- We need an optimization problem
  - Decision variable is the control sequence calculated over horizon N
  - Objective function to optimize

$$\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k])$$

Model-based Predictive Control

Brief recap

Find optimal control sequence using predictions based on a model.

- We need an optimization problem
  - Decision variable is the control sequence calculated over horizon N
  - Objective function to optimize
  - System's Model

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

Find optimal control sequence using predictions based on a model.

- We need an optimization problem
  - Decision variable is the control sequence calculated over horizon N
  - Objective function to optimize
  - System's Model
  - Other constraints to respect

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

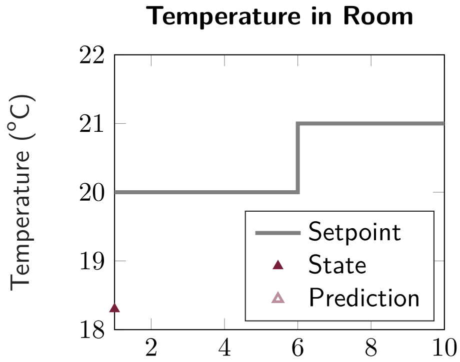
Find optimal control sequence using predictions based on a model.

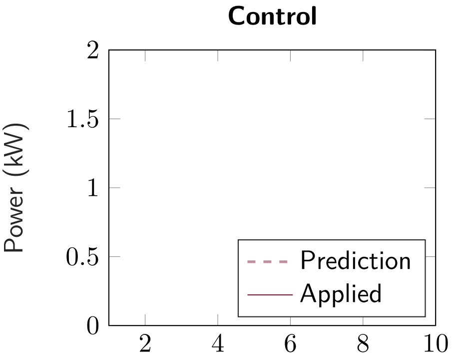
- We need an optimization problem
  - Decision variable is the control sequence calculated over horizon N
  - Objective function to optimize
  - System's Model
  - Other constraints to respect (QoS, technical restrictions, ...)

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

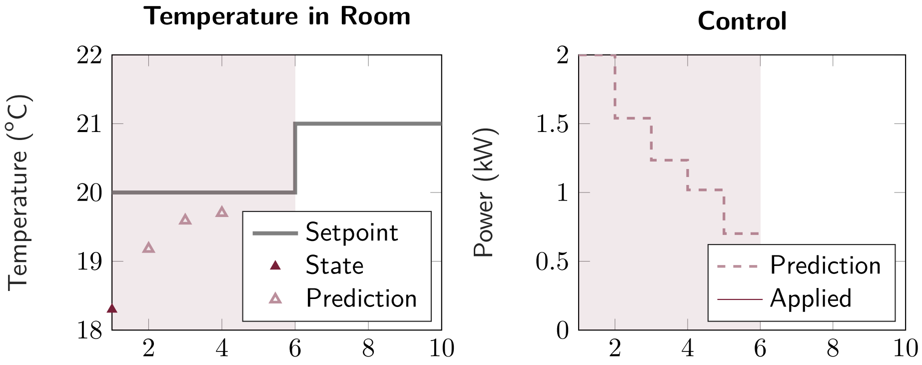
Model Predictive Control

In a nutshell





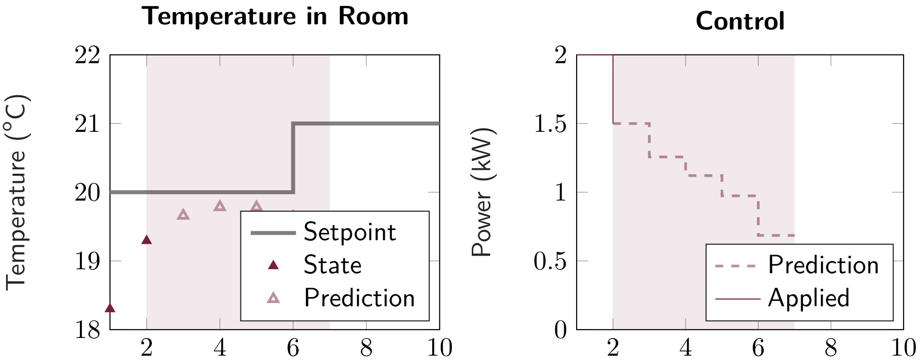
Find optimal control sequence



Model Predictive Control

In a nutshell

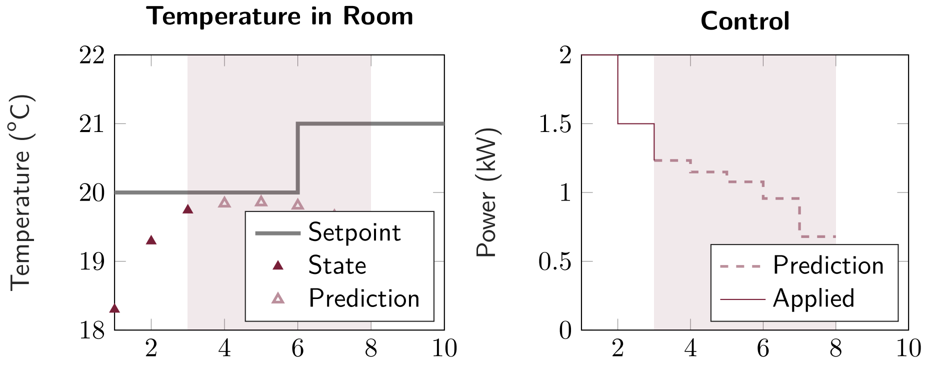
Find optimal control sequence, apply first element



Model Predictive Control

In a nutshell

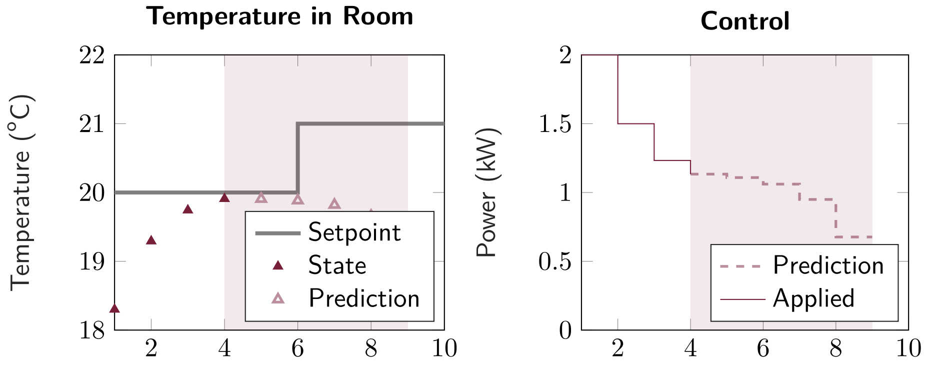
Find optimal control sequence, apply first element, rinse repeat



Model Predictive Control

In a nutshell

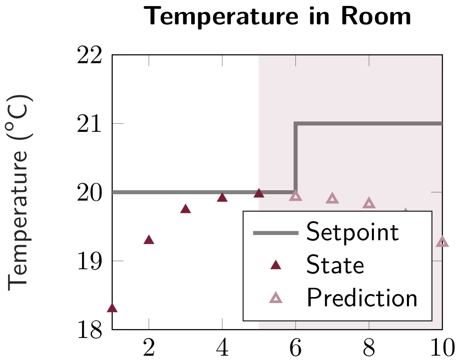
Find optimal control sequence, apply first element, rinse repeat → Receding Horizon

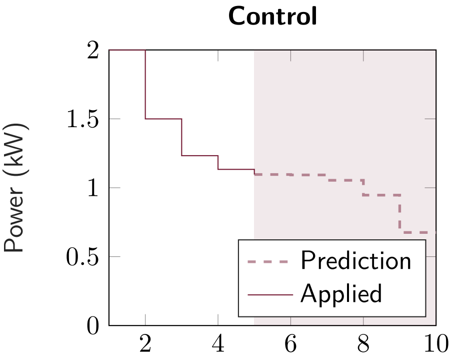


Model Predictive Control

In a nutshell

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





Model Predictive Control

Nothing is perfect

Nothing is perfect

- Problems

Model Predictive Control

Nothing is perfect

- Problems
    - Topology (Geographical distribution)

Model Predictive Control

Nothing is perfect

- Problems
  - Topology (Geographical distribution)
  - Complexity of calculation

Nothing is perfect

- Problems
    - Topology (Geographical distribution)
    - Complexity of calculation
    - Flexibility (Add/remove parts)

Nothing is perfect

- Problems
  - Topology (Geographical distribution)
  - Complexity of calculation
  - Flexibility (Add/remove parts)
  - Privacy (RGPD)

Nothing is perfect

- Problems
  - Topology (Geographical distribution)
  - Complexity of calculation
  - Flexibility (Add/remove parts)
  - Privacy (RGPD)
- Solution: Divide and Conquer (distributed MPC)

Outline

1. Decomposing the MPC

Outline

Outline

Outline

Distributed Model Predictive Control

Distributed Model Predictive Control

- We break the MPC optimization problem

Distributed Model Predictive Control

- We break the MPC optimization problem
- Make agents communicate

Distributed Model Predictive Control

- We break the MPC optimization problem
- Make agents communicate

In other words

Distributed Model Predictive Control

- We break the MPC optimization problem
- Make agents communicate

In other words

- Agents solve local problems

Distributed Model Predictive Control

- We break the MPC optimization problem
- Make agents communicate

In other words

- Agents solve local problems
- Exchange some variables

Distributed Model Predictive Control

- We break the MPC optimization problem
- Make agents communicate

In other words

- Agents solve local problems
- Exchange some variables
- Variables are updated

Distributed Model Predictive Control

- We break the MPC optimization problem
- Make agents communicate

In other words

- Agents solve local problems
- Exchange some variables
- Variables are updated

Until Convergence

Distributed Model Predictive Control

- We break the MPC optimization problem
- Make agents communicate

In other words

- Agents solve local problems
- Exchange some variables
- Variables are updated

Until Convergence

Remark

If agents exchange same variable → consensus problem

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods[1]:

[1] Boyd et al., "Notes on Decomposition Methods"

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods[1]:

- Local problems with auxiliary variables

[1] Boyd et al., "Notes on Decomposition Methods"

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods[1]:

- Local problems with auxiliary variables
- Update auxiliary variables

[1] Boyd et al., "Notes on Decomposition Methods"

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods[1]:

- Local problems with auxiliary variables
- Update auxiliary variables

Basically 2 choices[2]:

[1] Boyd et al., "Notes on Decomposition Methods"
[2] Other approaches, but similar concepts

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods[1]:

- Local problems with auxiliary variables
- Update auxiliary variables

Basically 2 choices[2]:

- Modify based on dual problem[3] (Solve with dual and send primal)

[1] Boyd et al., "Notes on Decomposition Methods"
[2] Other approaches, but similar concepts
[3] Lagrangian, ADMM, prices, etc +1000 articles in scopus

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods[1]:

- Local problems with auxiliary variables
- Update auxiliary variables

Basically 2 choices[2]:

- Modify based on dual problem[3] (Solve with dual and send primal)
- Modify based on primal problem (Solve with primal and send dual)

[1] Boyd et al., "Notes on Decomposition Methods"
[2] Other approaches, but similar concepts
[3] Lagrangian, ADMM, prices, etc +1000 articles in scopus

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods[1]:

- Local problems with auxiliary variables
- Update auxiliary variables

Basically 2 choices[2]:

- Modify based on dual problem[3] (Solve with dual and send primal)
- Modify based on primal problem (Solve with primal and send dual)

Many methods:

[1] Boyd et al., "Notes on Decomposition Methods"
[2] Other approaches, but similar concepts
[3] Lagrangian, ADMM, prices, etc +1000 articles in scopus

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods[1]:

- Local problems with auxiliary variables
- Update auxiliary variables

Basically 2 choices[2]:

- Modify based on dual problem[3] (Solve with dual and send primal)
- Modify based on primal problem (Solve with primal and send dual)

Many methods:

- Cutting plane, sub-gradient methods, . . .

[1] Boyd et al., “Notes on Decomposition Methods”

[2] Other approaches, but similar concepts

[3] Lagrangian, ADMM, prices, etc +1000 articles in scopus

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods[1]:

- Local problems with auxiliary variables
- Update auxiliary variables

Basically 2 choices[2]:

- Modify based on dual problem[3] (Solve with dual and send primal)
- Modify based on primal problem (Solve with primal and send dual)

Many methods:

- Cutting plane, sub-gradient methods, . . .

[1] Boyd et al., "Notes on Decomposition Methods"

[2] Other approaches, but similar concepts

[3] Lagrangian, ADMM, prices, etc +1000 articles in scopus

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods[1]:

- Local problems with auxiliary variables
- Update auxiliary variables

Basically 2 choices[2]:

- Modify based on dual problem[3] (Solve with dual and send primal)
- Modify based on primal problem (Solve with primal and send dual)

Many methods:

Security/privacy properties

- Cutting plane, sub-gradient methods, . . .

[1] Boyd et al., "Notes on Decomposition Methods"
[2] Other approaches, but similar concepts
[3] Lagrangian, ADMM, prices, etc +1000 articles in scopus

Distributed Model Predictive Control

It is about communication

- We break the MPC optimization problem
- Make agents communicate.

Distributed Model Predictive Control

It is about communication

- We break the MPC optimization problem
- Make agents communicate. But how?

Distributed Model Predictive Control

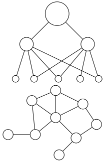
It is about communication

- We break the MPC optimization problem
- Make agents communicate. But how?
    - Many flavors to choose from

Distributed Model Predictive Control

It is about communication

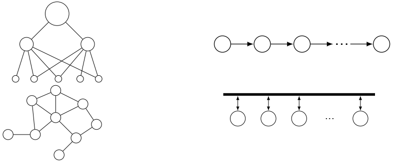
- We break the MPC optimization problem
- Make agents communicate. But how?
    - Many flavors to choose from
        - Hierarchical/Anarchical



Distributed Model Predictive Control

It is about communication

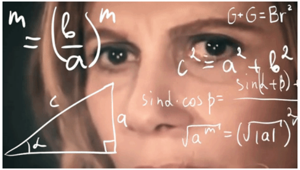
- We break the MPC optimization problem
- Make agents communicate. But how?
  - Many flavors to choose from
    - Hierarchical/Anarchical
    - Parallel/Sequential

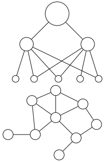


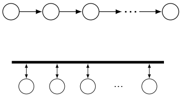
Distributed Model Predictive Control

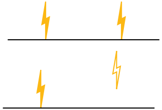
It is about communication

- We break the MPC optimization problem
- Make agents communicate. But how?
    - Many flavors to choose from
        - Hierarchical/Anarchical
        - Parallel/Sequential
        - Synchronous/Asynchronous





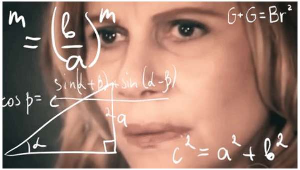


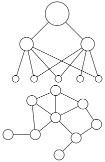


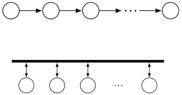
Distributed Model Predictive Control

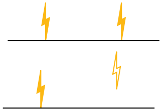
It is about communication

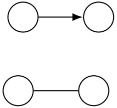
- We break the MPC optimization problem
- Make agents communicate. But how?
  - Many flavors to choose from
    - Hierarchical/Anarchical
    - Parallel/Sequential
    - Synchronous/Asynchronous
    - Bidirectional/Unidirectional







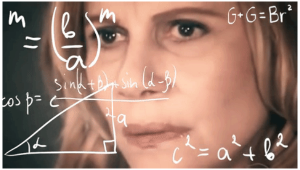


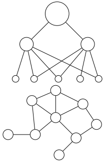


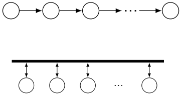
Distributed Model Predictive Control

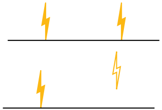
It is about communication

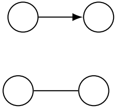
- We break the MPC optimization problem
- Make agents communicate. But how?
  - Many flavors to choose from[4]
    - Hierarchical/Anarchical
    - Parallel/Sequential
    - Synchronous/Asynchronous
    - Bidirectional/Unidirectional
    - ...









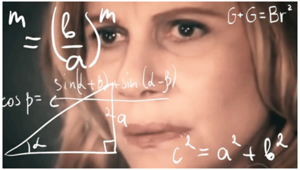


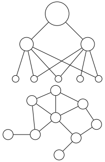
[4] José M Maestre, Negenborn, et al., Distributed Model Predictive Control made easy

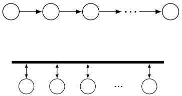
Distributed Model Predictive Control

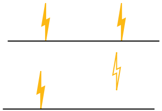
It is about communication

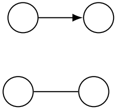
- We break the MPC optimization problem
- Make agents communicate. But how?
  - Many flavors to choose from[4]
    - Hierarchical/Anarchical
    - Parallel/Sequential
    - Synchronous/Asynchronous
    - Bidirectional/Unidirectional
    - ...











[4] José M Maestre, Negenborn, et al., Distributed Model Predictive Control made easy

Distributed Model Predictive Control

Optimization Decomposition

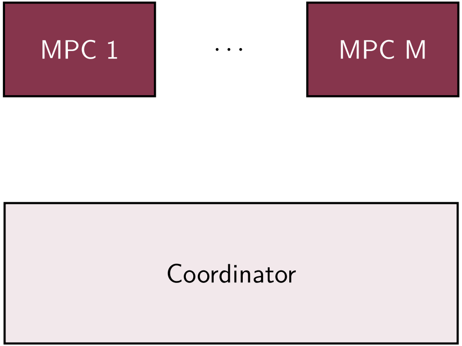
MPC

Distributed Model Predictive Control

Optimization Decomposition

Distributed Model Predictive Control

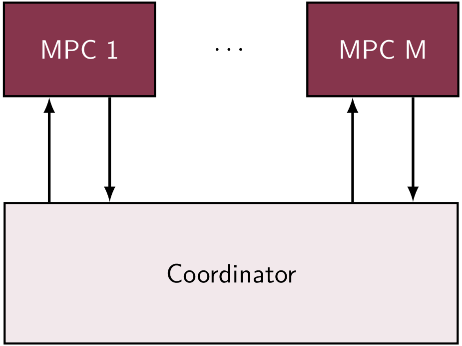
Optimization Decomposition



- Coordinator → Hierarchical

Distributed Model Predictive Control

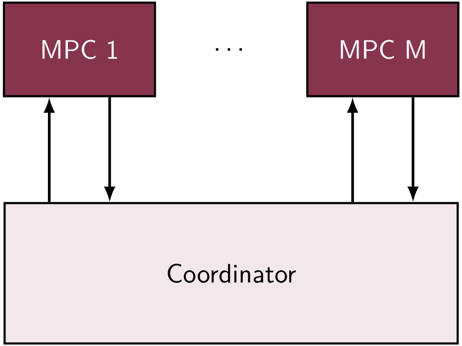
Optimization Decomposition



- Coordinator → Hierarchical
- Bidirectional

Distributed Model Predictive Control

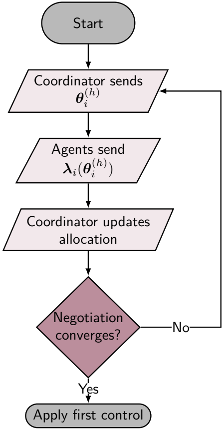
Optimization Decomposition



- Coordinator $\rightarrow$ Hierarchical
- Bidirectional
- No delay $\rightarrow$ Synchronous

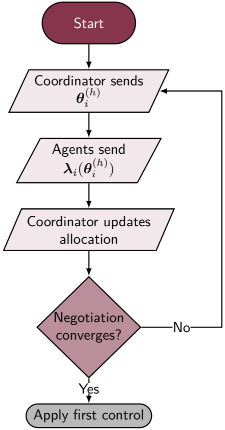
Primal Decomposition

or Quantity Decomposition | or Resource Allocation



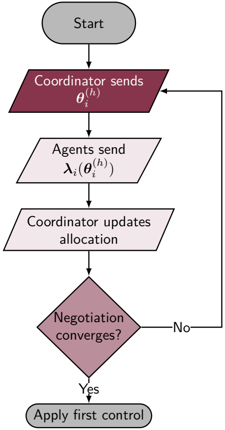
Primal Decomposition

or Quantity Decomposition | or Resource Allocation



Primal Decomposition

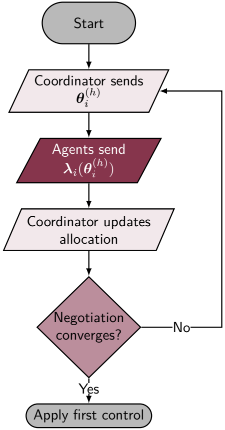
or Quantity Decomposition | or Resource Allocation



Allocation $\boldsymbol{\theta}_i$

Primal Decomposition

or Quantity Decomposition | or Resource Allocation

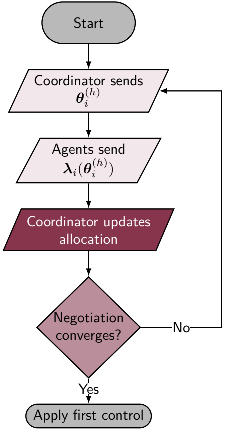


Allocation $\boldsymbol{\theta}_i$

Dissatisfaction $\boldsymbol{\lambda}_i$

Primal Decomposition

or Quantity Decomposition | or Resource Allocation



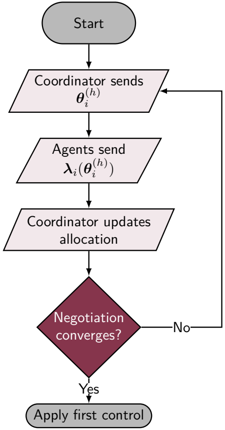
Allocation $\boldsymbol{\theta}_i$

Dissatisfaction $\boldsymbol{\lambda}_i$

Update $\boldsymbol{\theta}_i^+ = f_i(\boldsymbol{\theta}_i, \boldsymbol{\lambda}_i)$

Primal Decomposition

or Quantity Decomposition | or Resource Allocation

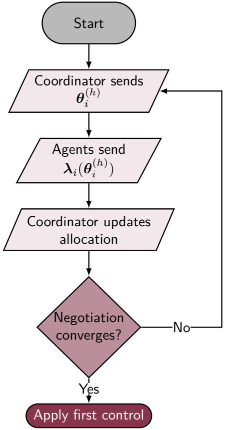


Allocation $\boldsymbol{\theta}_i$
Dissatisfaction $\boldsymbol{\lambda}_i$

Update $\boldsymbol{\theta}_i^+ = f_i(\boldsymbol{\theta}_i, \boldsymbol{\lambda}_i)$

Primal Decomposition

or Quantity Decomposition | or Resource Allocation



Allocation $\boldsymbol{\theta}_i$

Dissatisfaction $\boldsymbol{\lambda}_i$

Update $\boldsymbol{\theta}_i^+ = f_i(\boldsymbol{\theta}_i, \boldsymbol{\lambda}_i)$

Primal Decomposition

In detail

$$\begin{aligned} \underset{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_M}{\text{minimize}} \quad & \sum_{i \in \mathcal{M}} J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s.t.} \quad & \sum_{i \in \mathcal{M}} \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{u}_{\mathsf{total}} \end{aligned}$$

Primal Decomposition

In detail

- Objective is sum of local ones

$$\begin{aligned} \underset{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_M}{\text{minimize}} \quad & \sum_{i\in\mathcal{M}} J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s.t.} \quad & \sum_{i\in\mathcal{M}} \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{u}_{\text{total}} \end{aligned}$$

Primal Decomposition

In detail

- Objective is sum of local ones
- Constraints couple variables

$$\begin{aligned} \underset{\boldsymbol{u}_1,\dots,\boldsymbol{u}_M}{\text{minimize}} \quad & \sum_{i\in\mathcal{M}} J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s.t.} \quad & \sum_{i\in\mathcal{M}} \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{u}_{\text{total}} \end{aligned}$$

Primal Decomposition

In detail

- Objective is sum of local ones
- Constraints couple variables

$$\begin{aligned} \underset{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_M}{\text{minimize}} \quad & \sum_{i\in\mathcal{M}} J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s.t.} \quad & \sum_{i\in\mathcal{M}} \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{u}_{\text{total}} \end{aligned}$$

$$\downarrow \text{ For each } i \in \mathcal{M}$$

$$\begin{aligned} \underset{\boldsymbol{u}_i}{\text{minimize}} \quad & J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s. t.} \quad & \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{\theta}_i \end{aligned}$$

Primal Decomposition

In detail

- Objective is sum of local ones
- Constraints couple variables

1. Allocate $\boldsymbol{\theta}_i$ for each agent

$$\begin{aligned} \underset{\boldsymbol{u}_i}{\text{minimize}} \quad & J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s. t.} \quad & \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{\theta}_i \end{aligned}$$

Primal Decomposition

In detail

- Objective is sum of local ones
- Constraints couple variables

1. Allocate $\boldsymbol{\theta}_i$ for each agent
2. They solve local problems and

$$\begin{aligned} \underset{\boldsymbol{u}_i}{\text{minimize}} \quad & J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s. t.} \quad & \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{\theta}_i \end{aligned}$$

Primal Decomposition

In detail

- Objective is sum of local ones
- Constraints couple variables

1. Allocate $\boldsymbol{\theta}_i$ for each agent
2. They solve local problems and
3. Send dual variable $\boldsymbol{\lambda}_i$

$$\begin{aligned} \underset{\boldsymbol{u}_i}{\text{minimize}} \quad & J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s. t.} \quad & \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{\theta}_i : \boldsymbol{\lambda}_i \end{aligned}$$

Primal Decomposition

In detail

- Objective is sum of local ones
- Constraints couple variables

1. Allocate $\boldsymbol{\theta}_i$ for each agent
2. They solve local problems and
3. Send dual variable $\boldsymbol{\lambda}_i$
4. Allocation is updated

$$\begin{aligned} \underset{\boldsymbol{u}_i}{\text{minimize}} \quad & J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s. t.} \quad & \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{\theta}_i : \boldsymbol{\lambda}_i \end{aligned}$$

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

Primal Decomposition

In detail

- Objective is sum of local ones
- Constraints couple variables

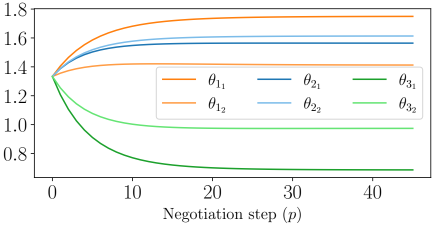
1. Allocate $\boldsymbol{\theta}_i$ for each agent
2. They solve local problems and
3. Send dual variable $\boldsymbol{\lambda}_i$
4. Allocation is updated (respect global constraint)

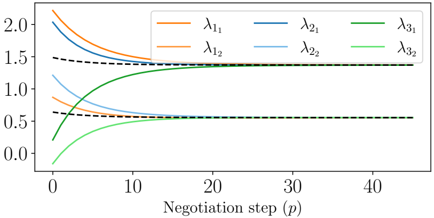
$$\begin{array}{ll} \underset{\boldsymbol{u}_i}{\text{minimize}} & J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s. t.} & \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{\theta}_i : \boldsymbol{\lambda}_i \end{array}$$

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

Example

Until everybody is evenly[5] dissatisfied





[5] For inequality constraints dynamics are more complex

Distributed Model Predictive Control

Negotiation works if agents comply.

Distributed Model Predictive Control

Negotiation works if agents comply.

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

Distributed Model Predictive Control

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

Recent in dMPC literature[6] (First article from 2017[7])

[6] $<$30 documents in scopus

[7] Velarde, Jose Maria Maestre, H. Ishii, et al., "Vulnerabilities in Lagrange-Based DMPC in the Context of Cyber-Security"

Distributed Model Predictive Control

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

Recent in dMPC literature[6] (First article from 2017[7])

- Incentive Brittany Region (Sustainable Energy + cybersecurity)

[6] <30 documents in scopus

[7] Velarde, Jose Maria Maestre, H. Ishii, et al., “Vulnerabilities in Lagrange-Based DMPC in the Context of Cyber-Security”

Distributed Model Predictive Control

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

Recent in dMPC literature[6] (First article from 2017[7])

- Incentive Brittany Region (Sustainable Energy + cybersecurity)
- CentraleSupélec Rennes - MPC for Smart Buildings

[6] <30 documents in scopus

[7] Velarde, Jose Maria Maestre, H. Ishii, et al., "Vulnerabilities in Lagrange-Based DMPC in the Context of Cyber-Security"

Distributed Model Predictive Control

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

Recent in dMPC literature[6] (First article from 2017[7])

- Incentive Brittany Region (Sustainable Energy + cybersecurity)
- CentraleSupélec Rennes - MPC for Smart Buildings

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

---

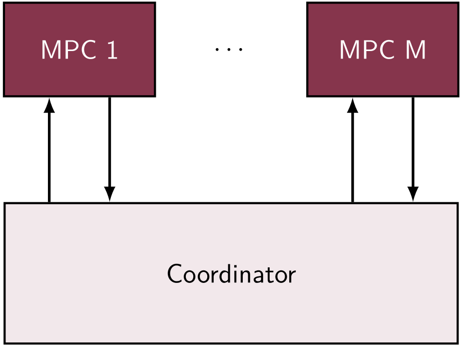
[6] $<$30 documents in scopus
[7] Velarde, Jose Maria Maestre, H. Ishii, et al., "Vulnerabilities in Lagrange-Based DMPC in the Context of Cyber-Security"

Outline

2 Attacks on the dMPC

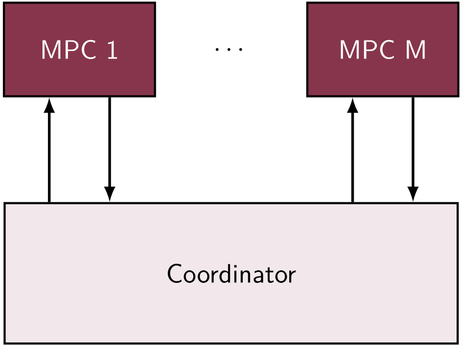
How can a non-cooperative agent attack?

Literature



How can a non-cooperative agent attack?

Literature

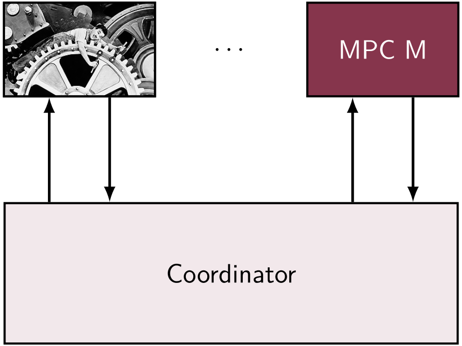


- Common attacks[8]

[8] Velarde, Jose Maria Maestre, Hideaki Ishii, et al., "Scenario-based defense mechanism for distributed model predictive control"

How can a non-cooperative agent attack?

Literature

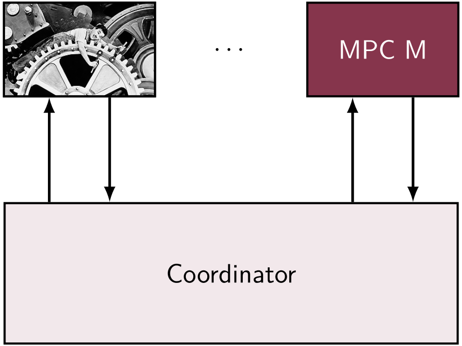


- Common attacks[8]

[8] Velarde, Jose Maria Maestre, Hideaki Ishii, et al., "Scenario-based defense mechanism for distributed model predictive control"

How can a non-cooperative agent attack?

Literature

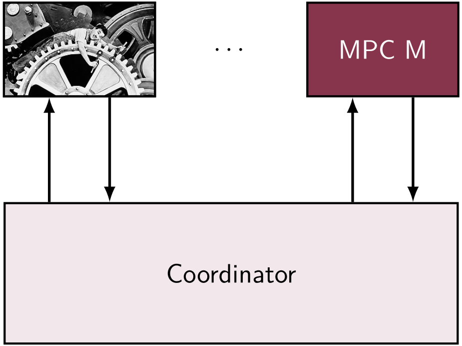


- Common attacks[8]
  - Fake objective function
  - Fake constraints
  - Use different control

[8] Velarde, Jose Maria Maestre, Hideaki Ishii, et al., "Scenario-based defense mechanism for distributed model predictive control"

How can a non-cooperative agent attack?

Literature



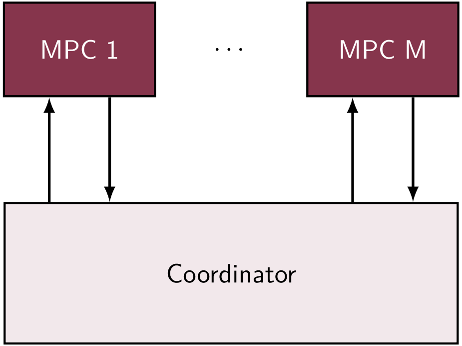
- Common attacks[8]
  - Fake objective function
  - Fake constraints
  - Use different control

} Deception Attacks

[8] Velarde, Jose Maria Maestre, Hideaki Ishii, et al., "Scenario-based defense mechanism for distributed model predictive control"

How can a non-cooperative agent attack?

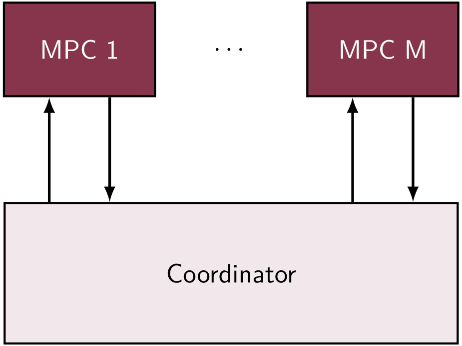
Our approach



- Primal decomposition

How can a non-cooperative agent attack?

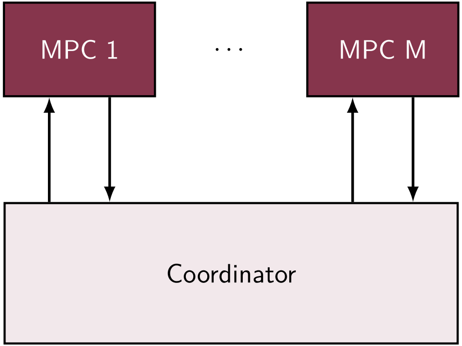
Our approach



- Primal decomposition
  - Maximum resources fixed

How can a non-cooperative agent attack?

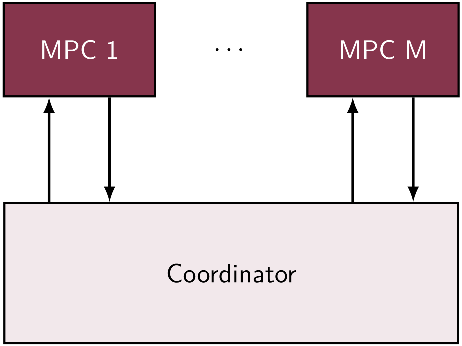
Our approach



- Primal decomposition
    - Maximum resources fixed
- We are in coordinator's shoes

How can a non-cooperative agent attack?

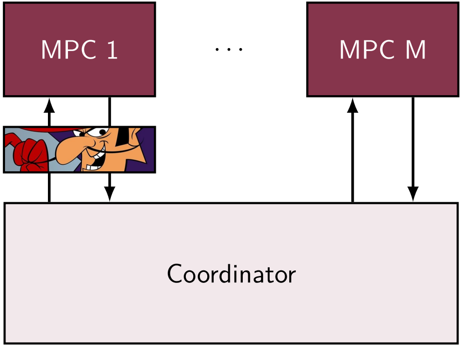
Our approach



- Primal decomposition
  - Maximum resources fixed
- We are in coordinator's shoes
- What matters is the interface

How can a non-cooperative agent attack?

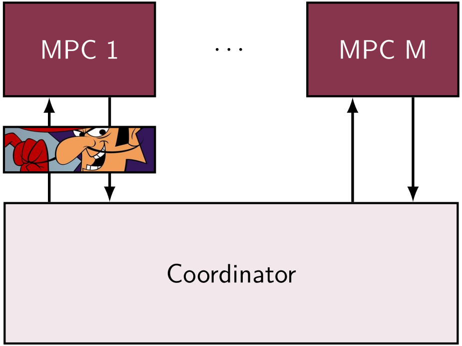
Our approach



- Primal decomposition
    - Maximum resources fixed
- We are in coordinator's shoes
- What matters is the interface
    - Attacker changes communication

How can a non-cooperative agent attack?

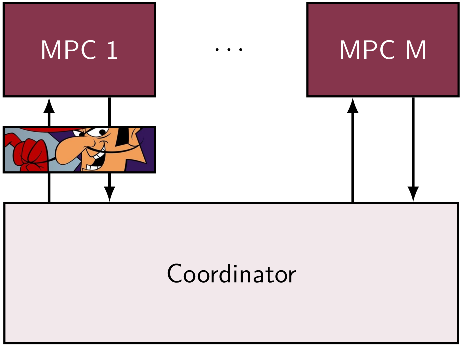
Our approach



- Primal decomposition
  - Maximum resources fixed
- We are in coordinator's shoes
- What matters is the interface
  - Attacker changes communication
    - False Data Injection

How can a non-cooperative agent attack?

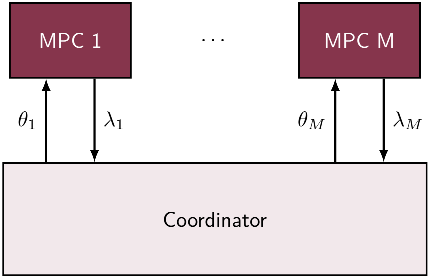
Our approach



- Primal decomposition
  - Maximum resources fixed
- We are in coordinator's shoes
- What matters is the interface
  - Attacker changes communication
    - False Data Injection

How can a non-cooperative agent attack?

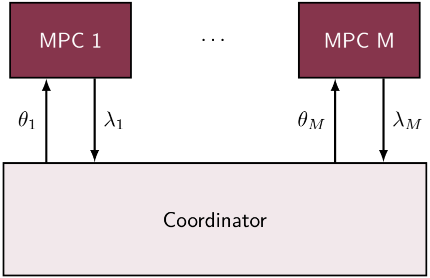
Our approach



- $\boldsymbol{\lambda}_i$ is the only interface

How can a non-cooperative agent attack?

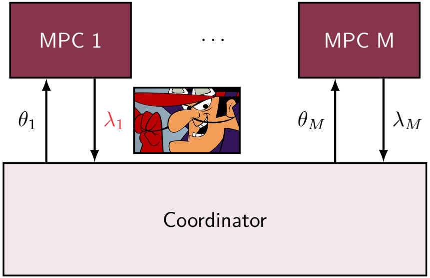
Our approach



- $\boldsymbol{\lambda}_i$ is the only interface
- $\boldsymbol{\lambda}_i$ obfuscate params. (+ Privacy)

How can a non-cooperative agent attack?

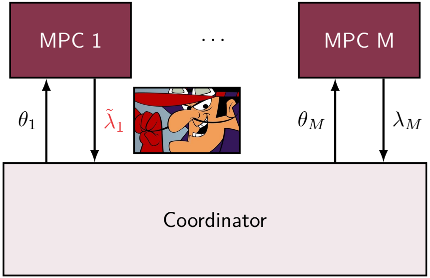
Our approach



- $\boldsymbol{\lambda}_i$ is the only interface
- $\boldsymbol{\lambda}_i$ obfuscate params. (+ Privacy)
- Malicious agent modifies $\boldsymbol{\lambda}_i$

How can a non-cooperative agent attack?

Our approach



- $\boldsymbol{\lambda}_i$ is the only interface
- $\boldsymbol{\lambda}_i$ obfuscate params. (+ Privacy)
- Malicious agent modifies $\boldsymbol{\lambda}_i$

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

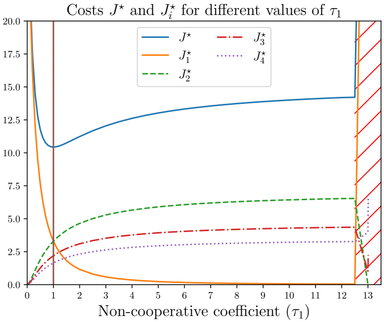
Example

Example

4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$
- Simulate for different τ_1 get J_i

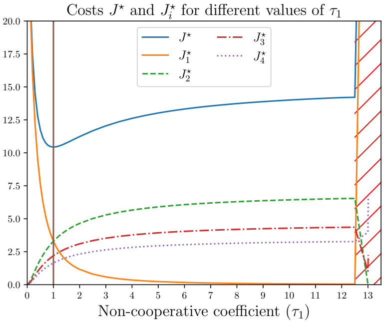
Example



4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$
- Simulate for different τ_1 get J_i

Example

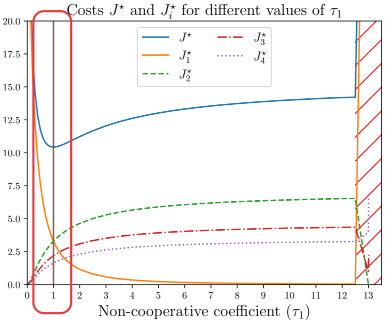


4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$
- Simulate for different τ_1 get J_i

- We can observe 3 things

Example

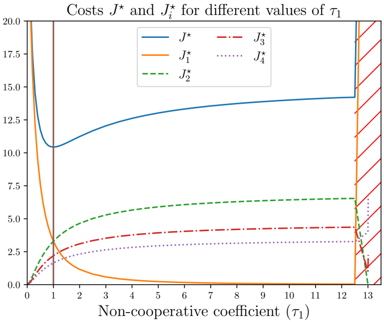


4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$
- Simulate for different τ_1 get J_i

- We can observe 3 things
  - Global minimum when $\tau_1 = 1$

Example

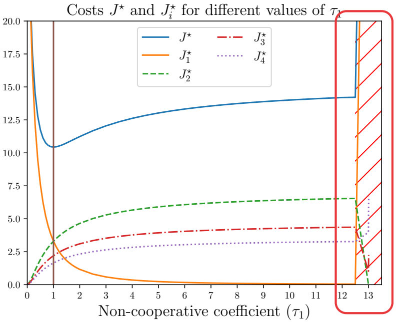


4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$
- Simulate for different τ_1 get J_i

- We can observe 3 things
  - Global minimum when $\tau_1 = 1$
  - Agent 1 benefits if τ_1 increases (inverse otherwise)

Example



4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$
- Simulate for different τ_1 get J_i

- We can observe 3 things
  - Global minimum when $\tau_1 = 1$
  - Agent 1 benefits if τ_1 increases (inverse otherwise)
  - All collapses if too greedy

- But can we mitigate these effects?

- But can we mitigate these effects?
- Yes! (At least in some cases)

Outline

3. Securing the dMPC

Classification of mitigation techniques

Passive (Robust)

Active (Resilient)

Classification of mitigation techniques

Passive (Robust)

- 1 mode

Active (Resilient)

- 2 modes

Classification of mitigation techniques

Passive (Robust)

- 1 mode

Active (Resilient)

- 2 modes
  1. Attack free
  2. When attack is detected

Classification of mitigation techniques

Passive (Robust)

- 1 mode

Active (Resilient)

- 2 modes
  1. Attack free
  2. When attack is detected
     - Detection/Isolation
     - Mitigation

Classification of mitigation techniques

Passive (Robust)

- 1 mode

Active (Resilient)

- 2 modes
  1. Attack free
  2. When attack is detected
     - Detection/Isolation
     - Mitigation

State of art

Security dMPC

	Decomposition	Resilient/Robust
[9]	Dual	Robust (Scenario)
[10]	Dual	Robust (f-robust)
[11]	Jacobi-Gauß	–
[12]	Dual	Resilient

[9] José M. Maestre et al., "Scenario-Based Defense Mechanism Against Vulnerabilities in Lagrange-Based Dmpc".

[10] Velarde, José M. Maestre, et al., "Vulnerabilities in Lagrange-Based Distributed Model Predictive Control".

[11] Chanfreut, J. M. Maestre, and H. Ishii, "Vulnerabilities in Distributed Model Predictive Control based on Jacobi-Gauss Decomposition".

[12] Ananduta et al., "Resilient Distributed Model Predictive Control for Energy Management of Interconnected Microgrids".

State of art

Security dMPC

	Decomposition	Resilient/Robust
[9]	Dual	Robust (Scenario)
[10]	Dual	Robust (f-robust)
[11]	Jacobi-Gauß	–
[12]	Dual	Resilient
Our	Primal	Resilient

[9] José M. Maestre et al., "Scenario-Based Defense Mechanism Against Vulnerabilities in Lagrange-Based Dmpc".

[10] Velarde, José M. Maestre, et al., "Vulnerabilities in Lagrange-Based Distributed Model Predictive Control".

[11] Chanfreut, J. M. Maestre, and H. Ishii, "Vulnerabilities in Distributed Model Predictive Control based on Jacobi-Gauss Decomposition".

[12] Ananduta et al., "Resilient Distributed Model Predictive Control for Energy Management of Interconnected Microgrids".

State of art

Security dMPC

	Decomposition	Resilient/Robust
[9]	Dual	Robust (Scenario)
[10]	Dual	Robust (f-robust)
[11]	Jacobi-Gauß	–
[12]	Dual	Resilient
Our	Primal	Resilient

[9] José M. Maestre et al., "Scenario-Based Defense Mechanism Against Vulnerabilities in Lagrange-Based Dmpc".

[10] Velarde, José M. Maestre, et al., "Vulnerabilities in Lagrange-Based Distributed Model Predictive Control".

[11] Chanfreut, J. M. Maestre, and H. Ishii, "Vulnerabilities in Distributed Model Predictive Control based on Jacobi-Gauss Decomposition".

[12] Ananduta et al., "Resilient Distributed Model Predictive Control for Energy Management of Interconnected Microgrids".

State of art

Security dMPC

	Decomposition	Resilient/Robust
[9]	Dual	Robust (Scenario)
[10]	Dual	Robust (f-robust)
[11]	Jacobi-Gauß	–
[12]	Dual	Resilient
Our	Primal	Resilient

[9] José M. Maestre et al., "Scenario-Based Defense Mechanism Against Vulnerabilities in Lagrange-Based Dmpc".

[10] Velarde, José M. Maestre, et al., "Vulnerabilities in Lagrange-Based Distributed Model Predictive Control".

[11] Chanfreut, J. M. Maestre, and H. Ishii, "Vulnerabilities in Distributed Model Predictive Control based on Jacobi-Gauss Decomposition".

[12] Ananduta et al., "Resilient Distributed Model Predictive Control for Energy Management of Interconnected Microgrids".

State of art

Security dMPC

	Decomposition	Resilient/Robust	Detection	Mitigation
[9]	Dual	Robust (Scenario)	NA	NA
[10]	Dual	Robust (f-robust)	NA	NA
[11]	Jacobi-Gauß	–	–	–
[12]	Dual	Resilient	Analyt./Learn.	Disconnect (Robustness)
Our	Primal	Resilient	Active Analyt./Learn.	Data reconstruction

[9] José M. Maestre et al., “Scenario-Based Defense Mechanism Against Vulnerabilities in Lagrange-Based Dmpc”.

[10] Velarde, José M. Maestre, et al., “Vulnerabilities in Lagrange-Based Distributed Model Predictive Control”.

[11] Chanfreut, J. M. Maestre, and H. Ishii, “Vulnerabilities in Distributed Model Predictive Control based on Jacobi-Gauss Decomposition”.

[12] Ananduta et al., “Resilient Distributed Model Predictive Control for Energy Management of Interconnected Microgrids”.

State of art

Security dMPC

	Decomposition	Resilient/Robust	Detection	Mitigation
[9]	Dual	Robust (Scenario)	NA	NA
[10]	Dual	Robust (f-robust)	NA	NA
[11]	Jacobi-Gauß	–	–	–
[12]	Dual	Resilient	Analyt./Learn.	Disconnect (Robustness)
Our	Primal	Resilient	Active Analyt./Learn.	Data reconstruction

[9] José M. Maestre et al., "Scenario-Based Defense Mechanism Against Vulnerabilities in Lagrange-Based Dmpc".

[10] Velarde, José M. Maestre, et al., "Vulnerabilities in Lagrange-Based Distributed Model Predictive Control".

[11] Chanfreut, J. M. Maestre, and H. Ishii, "Vulnerabilities in Distributed Model Predictive Control based on Jacobi-Gauss Decomposition".

[12] Ananduta et al., "Resilient Distributed Model Predictive Control for Energy Management of Interconnected Microgrids".

Attack model

Liar, Liar, Pants of fire

Attack model

Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction

Attack model

Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Attack model

Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Assumptions

Attack model

Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Assumptions

- *Same attack during negotiation*

Attack model

Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Assumptions

- *Same attack during negotiation*
- *Attacker satisfied only if it really is*

Attack model

Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Assumptions

- *Same attack during negotiation*
- *Attacker satisfied only if it really is*
    - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$

Attack model

Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Assumptions

- *Same attack during negotiation*
- *Attacker satisfied only if it really is*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- $\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i$

Attack model

Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Assumptions

- *Same attack during negotiation*
- *Attacker satisfied only if it really is*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- $\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i$

- Attack is invertible $\rightarrow \exists T_i[k]^{-1}$

Attack model

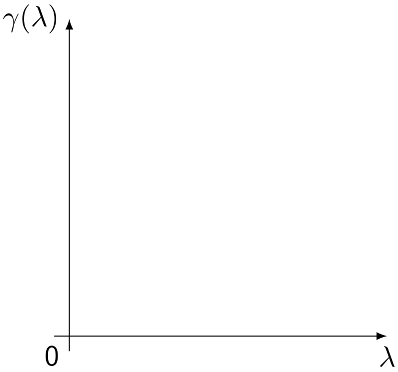
Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Assumptions

- *Same attack during negotiation*
- *Attacker satisfied only if it really is*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- $\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i$

- Attack is invertible $\rightarrow \exists T_i[k]^{-1}$



Attack model

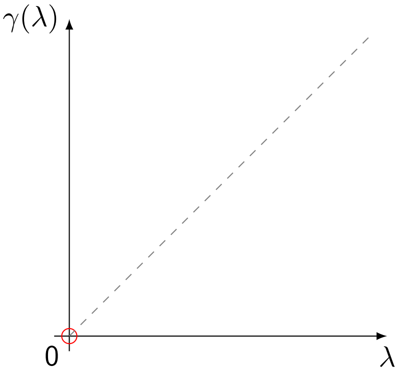
Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Assumptions

- *Same attack during negotiation*
- *Attacker satisfied only if it really is*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- $\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i$

- Attack is invertible $\rightarrow \exists T_i[k]^{-1}$



Attack model

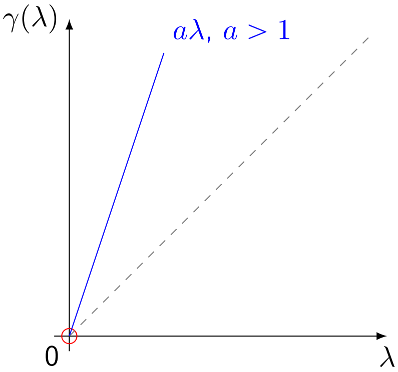
Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Assumptions

- *Same attack during negotiation*
- *Attacker satisfied only if it really is*
    - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- $\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i$

- Attack is invertible $\rightarrow \exists T_i[k]^{-1}$



For Further Reading I

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

Nogueira, Rafael Accácio. "Security of DMPC under False Data Injection". 2022CSUP0006. PhD thesis. CentraleSupélec, 2022. URL: http://www.theses.fr/2022CSUP0006.