

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

25/05/2023 @ Écully



<https://bit.ly/43h2jms>

Rafael Accácio Nogueira

Postdoctoral researcher at LAAS/CNRS

*Guaranteed relative localization and anti collision
scenario for autonomous vehicles*

Project AutOCampus (GIS neOCampus)

Advised by Soheib Fergani



Bachelor Thesis at Escola Politécnica/UFRJ

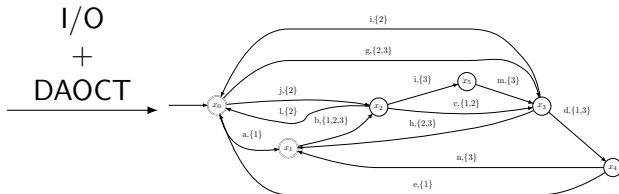
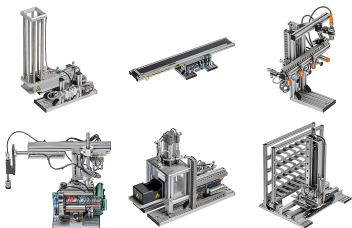
Identification of DES for fault-diagnosis

Advised by Marcos Vicente de Brito Moreira

Politécnica
UFRJ



UFRJ
UNIVERSIDADE FEDERAL
DO RIO DE JANEIRO



About me

Doctoral Thesis at CentraleSupélec/IETR

Security of dMPC under False Data Injection

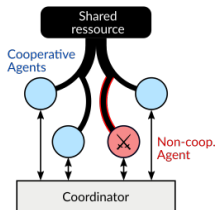
Advised by Hervé Guéguen and Romain Bourdais

CS-Rennes (Expertise in MPC for Smart Buildings)

Brittany Region Interest (Cybersecurity)



CentraleSupélec



Multiple systems interacting



Multiple systems interacting



- Distribution:
 - Electricity
 - Heat
 - Water
- Traffic
- ...

Multiple systems interacting under



- Technical/Comfort Constraints
- We also want
 - Minimize consumption
 - Maximize satisfaction
 - Follow a trajectory
- Solution \rightarrow MPC

Model-based Predictive Control

Brief recap

Find optimal control sequence using predictions based on a model.

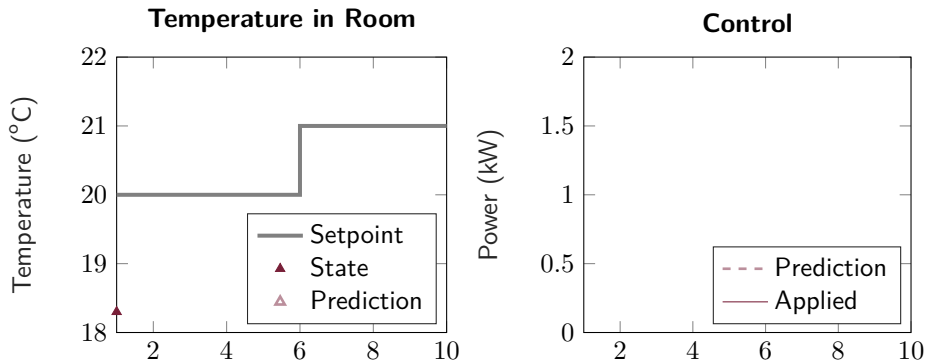
- We need an optimization problem
 - Decision variable is the control sequence calculated over horizon N
 - Objective function to optimize
 - System's Model
 - Other constraints to respect (QoS, technical restrictions, ...)

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

Model Predictive Control

In a nutshell

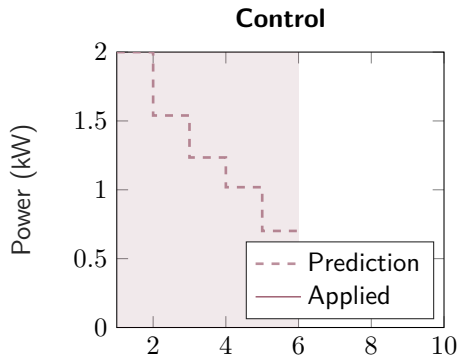
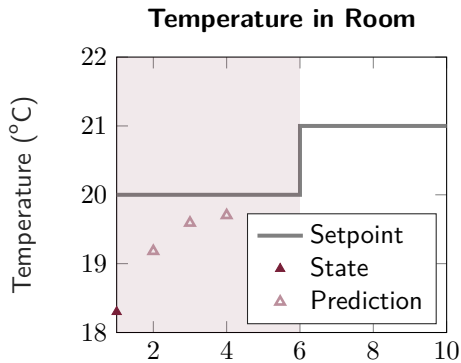
Find optimal control sequence, apply first element, rinse repeat → Receding Horizon



Model Predictive Control

In a nutshell

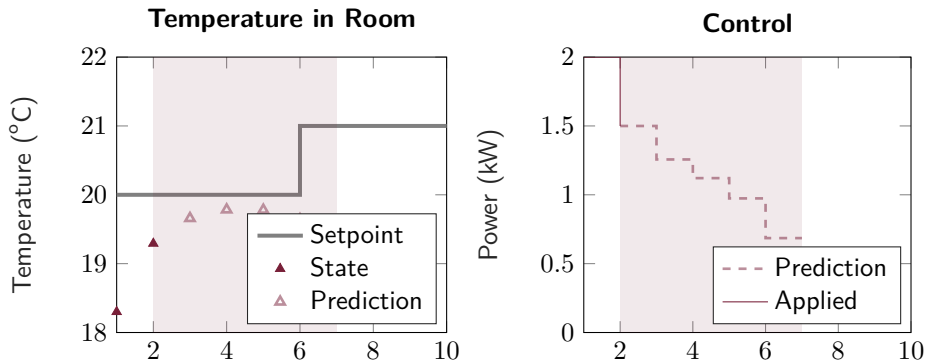
Find optimal control sequence, apply first element, rinse repeat → Receding Horizon



Model Predictive Control

In a nutshell

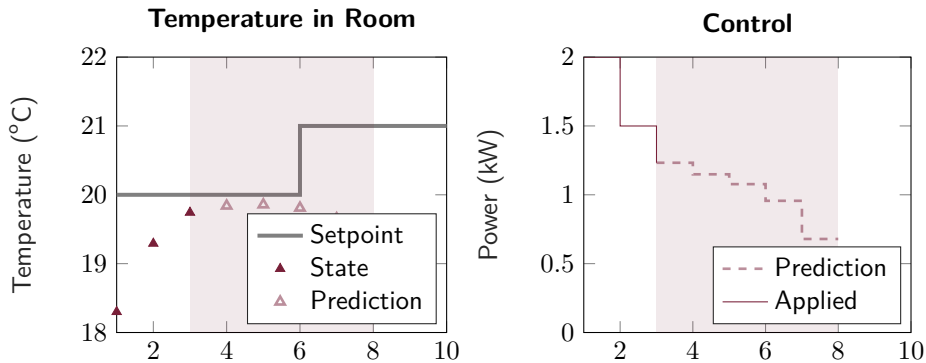
Find optimal control sequence, apply first element, rinse repeat → Receding Horizon



Model Predictive Control

In a nutshell

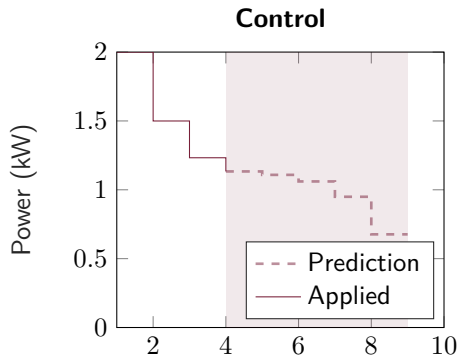
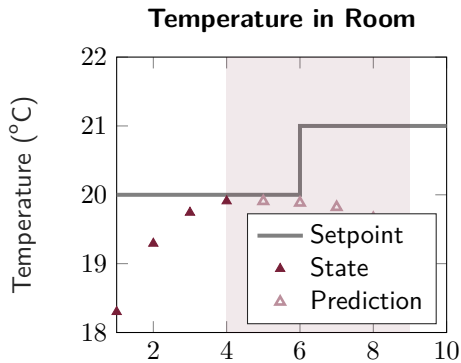
Find optimal control sequence, apply first element, rinse repeat → Receding Horizon



Model Predictive Control

In a nutshell

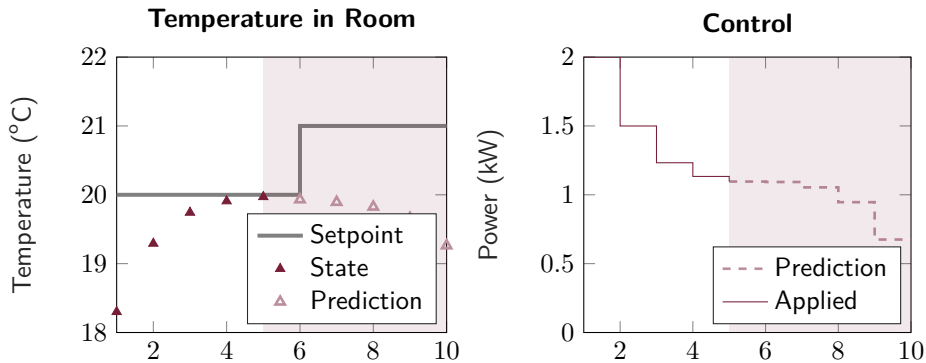
Find optimal control sequence, apply first element, rinse repeat → Receding Horizon



Model Predictive Control

In a nutshell

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



Model Predictive Control

Nothing is perfect

- Issues
 - Topology
 - Complexity of calculation
 - Flexibility (Add/remove parts)
 - Privacy (RGPD)
- Solution: distributed MPC

Objective

Study security in dMPC context

Security in dMPC context is relatively new¹ (First article from 2017²)

- How fragile are dMPC structures?
- How can agents act non-cooperatively?
- How to identify such agents and mitigate the effects?

¹<30 documents in scopus

²Velarde, Jose Maria Maestre, H. Ishii, et al., "Vulnerabilities in Lagrange-Based DMPC in the Context of Cyber-Security"

Outline

- ① Decomposing the MPC
- ② Attacks on the dMPC
- ③ Securing the dMPC
- ④ Conclusion

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 \rightarrow consensus problem

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods³:

- Local problems with auxiliary variables
- Update auxiliary variables


Basically 2 choices⁴:

- Modify based on dual problem⁵ (Solve with dual and send primal)
- Modify based on **primal problem** (Solve with primal and send dual)

Many methods:

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

Security/privacy properties

³  Boyd et al., “Notes on Decomposition Methods”

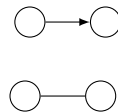
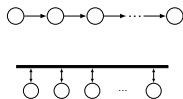
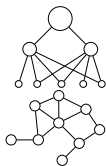
⁴ Other approaches, but similar concepts

⁵ 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. But how?
 - Many flavors to choose from⁶
 - Hierarchical/Anarchical
 - Parallel/Sequential
 - Synchronous/Asynchronous
 - Bidirectional/Unidirectional
 - ...



⁶

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

Distributed Model Predictive Control

Optimization Decomposition



MPC

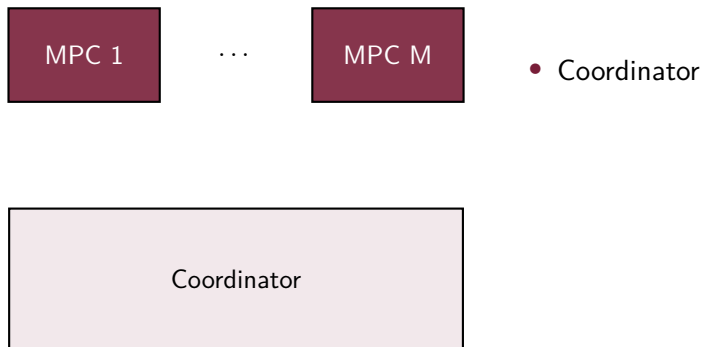
Distributed Model Predictive Control

Optimization Decomposition



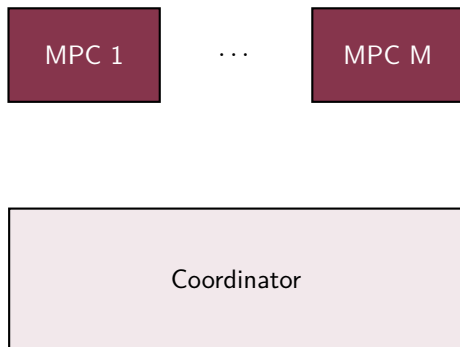
Distributed Model Predictive Control

Optimization Decomposition



Distributed Model Predictive Control

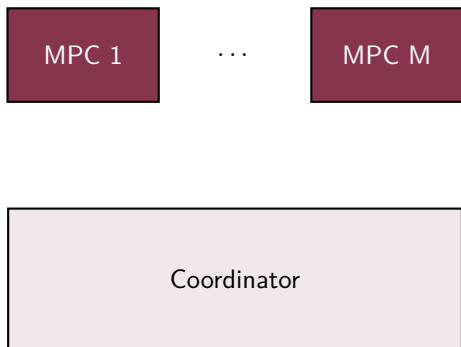
Optimization Decomposition



- Coordinator
 - Enforce global constraints

Distributed Model Predictive Control

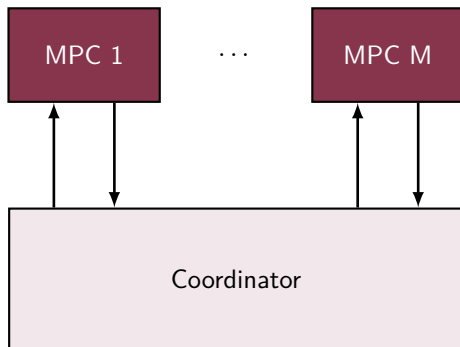
Optimization Decomposition



- Coordinator → Hierarchical
 - Enforce global constraints

Distributed Model Predictive Control

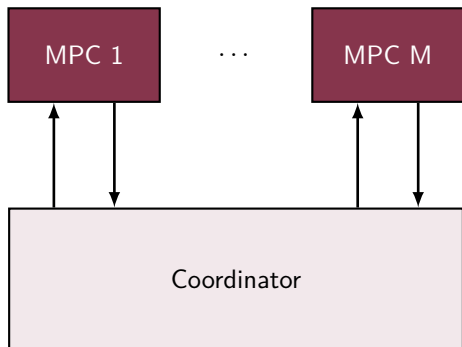
Optimization Decomposition



- Coordinator → Hierarchical
 - Enforce global constraints
- Bidirectional

Distributed Model Predictive Control

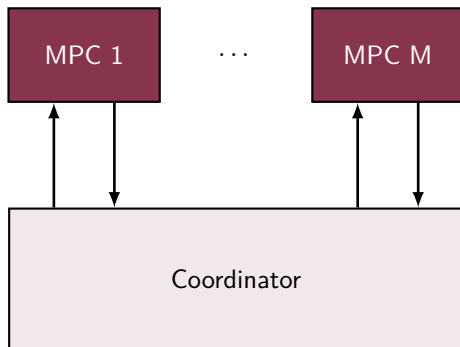
Optimization Decomposition



- Coordinator \rightarrow Hierarchical
 - Enforce global constraints
- Bidirectional
- No delay \rightarrow Synchronous

Distributed Model Predictive Control

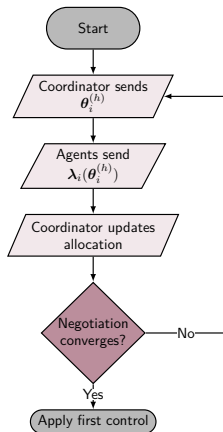
Optimization Decomposition



- Coordinator → Hierarchical
 - Enforce global constraints
- Bidirectional
- No delay → Synchronous
- But what to send?

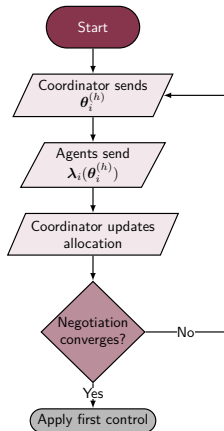
Primal Decomposition

or Quantity Decomposition | or Resource Allocation



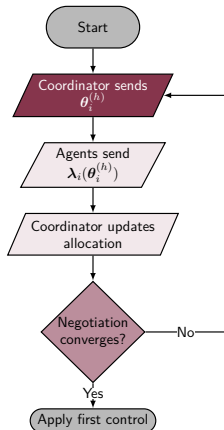
Primal Decomposition

or Quantity Decomposition | or Resource Allocation



Primal Decomposition

or Quantity Decomposition | or Resource Allocation

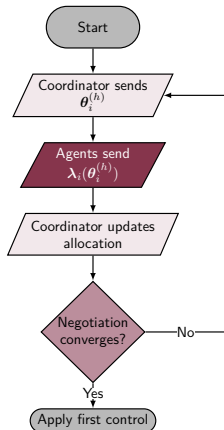


Allocation θ_i



Primal Decomposition

or Quantity Decomposition | or Resource Allocation

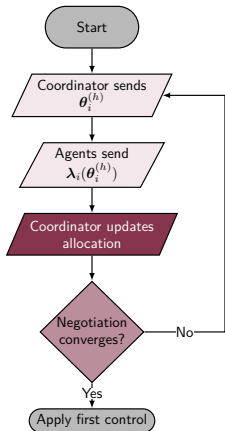


Allocation θ_i
Dissatisfaction λ_i



Primal Decomposition

or Quantity Decomposition | or Resource Allocation



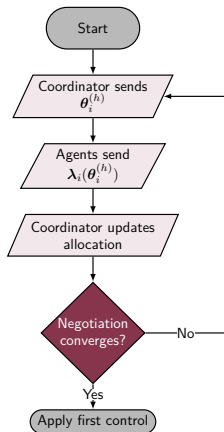
Allocation θ_i
Dissatisfaction λ_i



Update $\theta_i^+ = f_i(\theta_i, \lambda_i)$

Primal Decomposition

or Quantity Decomposition | or Resource Allocation



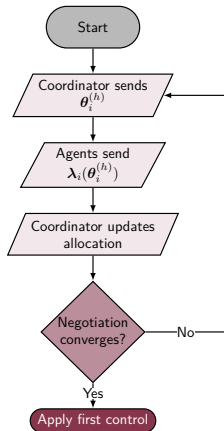
Allocation θ_i
Dissatisfaction λ_i



Update $\theta_i^+ = f_i(\theta_i, \lambda_i)$

Primal Decomposition

or Quantity Decomposition | or Resource Allocation



Allocation θ_i
Dissatisfaction λ_i



Update $\theta_i^+ = f_i(\theta_i, \lambda_i)$

Primal Decomposition

In detail

- ① Allocate θ_i for each agent
- ② They solve local problems and
- ③ Send dual variable λ_i ⁷
- ④ Allocation is updated⁸
(respect global constraint)

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

↓ For each $i \in \mathcal{M}$

$$\begin{aligned}
 & \underset{\mathbf{u}_i}{\text{minimize}} && J_i(\mathbf{x}_i, \mathbf{u}_i) \\
 & \text{s.t.} && \mathbf{h}_i(\mathbf{x}_i, \mathbf{u}_i) \leq \theta_i : \lambda_i
 \end{aligned}$$

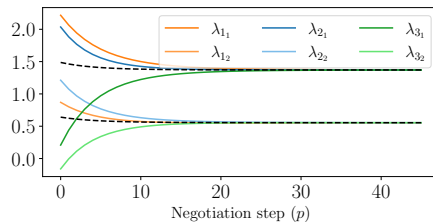
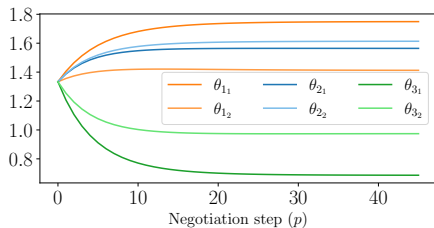
$$\theta[k]^{(p+1)} = \text{Proj}^{\mathcal{S}}(\theta[k]^{(p)} + \rho^{(p)} \lambda[k]^{(p)})$$

⁷It obfuscates system's parameters (+ Privacy)

⁸Only equation to change to add/remove agents

Example

Until everybody is evenly⁹ dissatisfied



⁹For inequality constraints dynamics are more complex

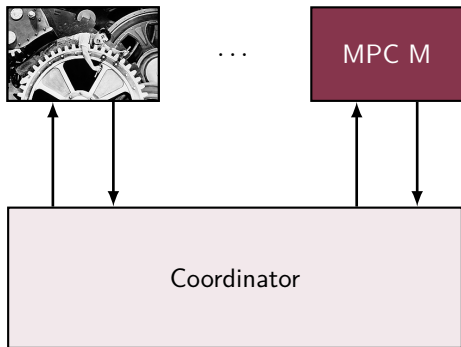
Distributed Model Predictive Control

Negotiation works if agents comply.

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

How can a non-cooperative agent attack?

Literature

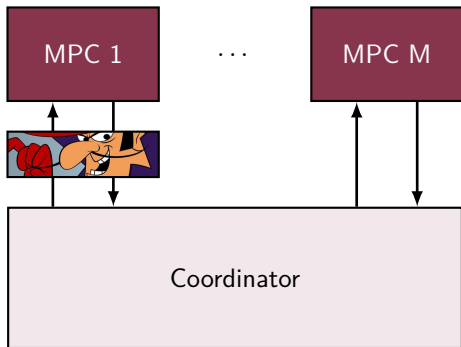


- Common attacks¹⁰
 - Fake objective function
 - Fake constraints
 - Use different control
- } Deception Attacks

¹⁰Velarde, Jose Maria Maestre, Hideaki Ishii, et al., "Scenario-based defense mechanism for distributed model predictive control"

How can a non-cooperative agent attack?

Our approach¹¹

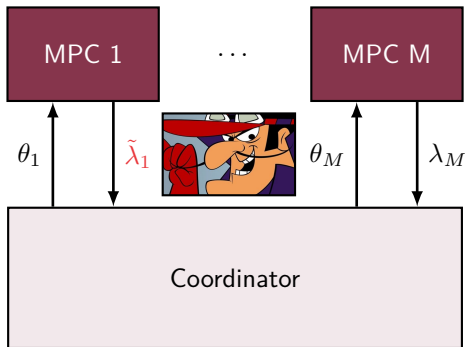


- Primal decomposition
 - Maximum resources fixed
- We are in coordinator's shoes
- What matters is the interface
 - Attacker changes communication
 - **False Data Injection**

¹¹Nogueira, Bourdais, and Guéguen, "Detection and Mitigation of Corrupted Information in Distributed Model Predictive Control Based on Resource Allocation"

How can a non-cooperative agent attack?

Our approach¹¹



- λ_i is the only interface
- Malicious agent modifies λ_i

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

¹¹Nogueira, Bourdais, and Guéguen, “Detection and Mitigation of Corrupted Information in Distributed Model Predictive Control Based on Resource Allocation”

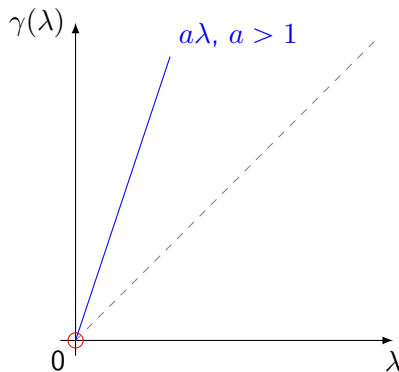
Attack model

Liar, Liar, Pants of fire

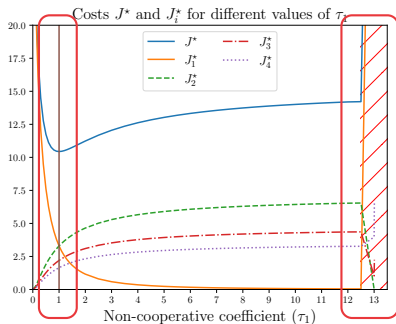
- $\lambda \geq 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Assumptions

- *Same attack during negotiation*
- *Attacker satisfied only if it really is*
 - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- $\tilde{\lambda}_i = T_i[k]\lambda_i$
- Attack is invertible $\rightarrow \exists T_i[k]^{-1}$



Example



4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\lambda}_1 = \gamma_1(\lambda_1) = \tau_1 I \lambda_1$
- 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?
- Yes! (At least in some cases)

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	Detection	Mitigation
¹²	Dual	Robust (Scenario)	NA	NA
¹³	Dual	Robust (f-robust)	NA	NA
¹⁴	Jacobi-Gauß	—	—	—
¹⁵	Dual	Resilient	Analyt./Learn.	Disconnect (Robustness)

¹²José M. Maestre et al., “Scenario-Based Defense Mechanism Against Vulnerabilities in Lagrange-Based Dmpc”.

¹³Velarde, José M. Maestre, et al., “Vulnerabilities in Lagrange-Based Distributed Model Predictive Control”.

¹⁴Chanfreut, J. M. Maestre, and H. Ishii, “Vulnerabilities in Distributed Model Predictive Control based on Jacobi-Gauss Decomposition”.

¹⁵Ananduta et al., “Resilient Distributed Model Predictive Control for Energy Management of Interconnected Microgrids”.

Our Approach

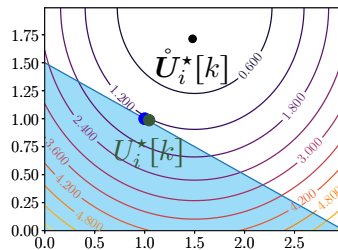
Explore Scarcity

- Resilient
- Analytical/Learning } Parameter
- Data reconstruction } Estimation
- Explore Scarcity

What are deprived systems?

Systems whose optimal solution has all constraints active

- Unconstrained Solution $\mathring{U}_i^*[k]$
- $h_i(\mathring{U}_i^*[k]) > \theta_i[k] \rightarrow$ Scarce resources
 - Solution projected onto boundary
 - Same as with equality constraints¹⁶



$$\begin{array}{ll} \text{minimize} & \frac{1}{2} \|U_i[k]\|_{H_i}^2 + f_i[k]^T U_i[k] \\ \text{subject to} & \bar{\Gamma}_i U_i[k] \leq \theta_i[k] : \lambda_i[k] \end{array}$$

\rightarrow

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} \|U_i[k]\|_{H_i}^2 + f_i[k]^T U_i[k] \\ \text{subject to} & \bar{\Gamma}_i U_i[k] = \theta_i[k] : \lambda_i[k] \end{array}$$

¹⁶If system can have all constraints active simultaneously

Analyzing Deprived Systems

Assumptions

- *Quadratic local problems*
- *Linear inequality constraints*
- *Scarcity*
- Solution is analytical and affine

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

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

$$(\text{local parameters unknown by coordinator}) \left\{ \begin{array}{l} \bullet \mathbf{P}_i \text{ is time invariant} \\ \bullet \mathbf{s}_i[k] \text{ is time variant} \end{array} \right.$$

Deprived Systems

Under attack!

- Normal behavior
 - Affine solution

$$\lambda_i[k] = -P_i \theta_i[k] - s_i[k]$$

- Under attack $\rightarrow \tilde{\lambda}_i = T_i[k] \lambda_i$
 - Parameters modified

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

- But wait! P_i is not supposed to change!
- Change \rightarrow Probably an Attack! Let's take advantage of this!

Detection Mechanism

- We estimate¹⁷ $\hat{P}_i[k]$ and $\hat{\mathbf{s}}_i[k]$ such as:

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

Assumption

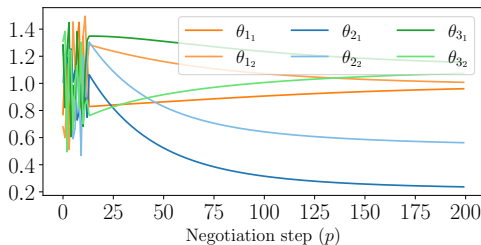
We can estimate \bar{P}_i from a attack free negotiation

- If $\left\| \hat{P}_i[k] - \bar{P}_i \right\|_F > \epsilon_P \rightarrow \text{Attack}$
- Ok, but how can we estimate $\hat{P}_i[k]$?

¹⁷Using Recursive Least Squares for example

Estimating $\hat{P}_i[k]$

- We estimate $\hat{P}_i[k]$ and $\hat{s}_i[k]$ simultaneously using RLS
- Challenge: Online estimation during negotiation fails
 - Update function couples θ_i^p and $\lambda_i^p \rightarrow$ low input excitation
- Solution: Send a random¹⁸ sequence to increase excitation until convergence.



¹⁸A random signal causes persistent excitation of any order ( Adaptive Control)

Classification of mitigation techniques

- Active (Resilient)
 - ① Detection/Isolation ✓
 - ② Mitigation ?

Mitigation mechanism

Reconstructing λ_i

- Now, we have $\hat{\tilde{P}}_i[k]$
 - Since $\tilde{P}_i[k] = T_i[k]\bar{P}_i$
 - We can recover $T_i[k]^{-1}$

$$\widehat{T_i[k]^{-1}} = P_i \hat{\tilde{P}}_i[k]^{-1}$$

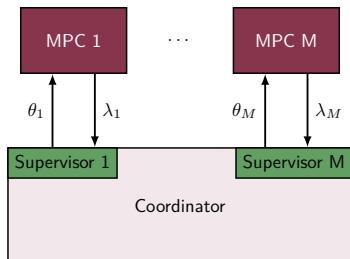
- Reconstruct λ_i

$$\lambda_i^{\text{rec}} = -\bar{P}_i \theta_i - \widehat{T_i[k]^{-1}} \hat{\tilde{s}}_i[k]$$

- Choose adequate version for coordination

$$\lambda_i^{\text{mod}} = \begin{cases} \lambda_i^{\text{rec}}, & \text{if attack detected} \\ \tilde{\lambda}_i, & \text{otherwise} \end{cases}$$

Complete Mechanism



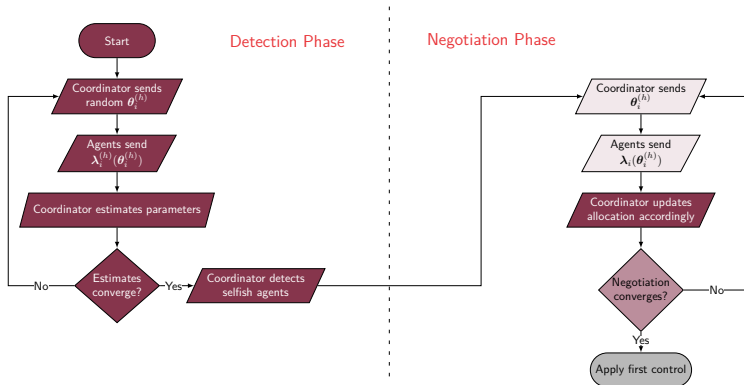
- Supervise exchanges by inquiring the agents
- Estimate how they will behave

Two Phases

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

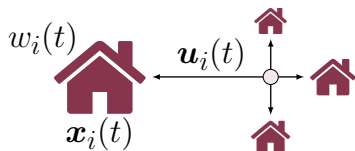
Complete algorithm

RPdMPC-DS¹⁹



¹⁹Nogueira, Bourdais, and Guéguen, "Detection and Mitigation of Corrupted Information in Distributed Model Predictive Control Based on Resource Allocation".

Example

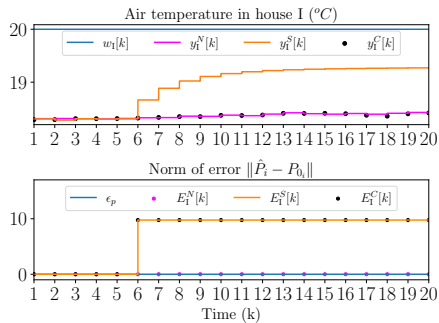


District Heating Network (4 Houses)

- Houses modeled using 3R-2C (monozone)
- Not enough power
- Period of 5h ($T_s = 0.25h \rightarrow k = \{1 : 20\}$)
- Prediction horizon ($N = 4$)
- 3 scenarios
 - Ⓝ Nominal
 - Ⓒ Agent I cheats (dMPC)
 - Ⓢ Agent I cheats (RPdMPC-DS)

Results

Temporal



Temperature in house I.

Error $E_I(k)$.

N Nominal, **S** Selfish, **C** Corrected

- Agent starts cheating in $k = 6$
- S** Agent increases its comfort
- C** Restablish behavior close to **N**



Results

Costs

Objective functions J_i (Normalized error %)

Agent	Selfish	Corrected
I	-36.3	0.5
II	21.67	-0.55
III	17.39	-0.0
IV	17.63	-0.09
Global	3.53	0.02

Relaxing scarcity assumption

- Systems are not completely deprived
 - We can't change our constraints to equality ones anymore
 - Nor use the simpler update equation

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

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

Analyzing System

Solution for $\lambda_i[k]$

Instead of having one single affine solution

$$\lambda_i[k] = -P_i \theta_i[k] - s_i[k]$$

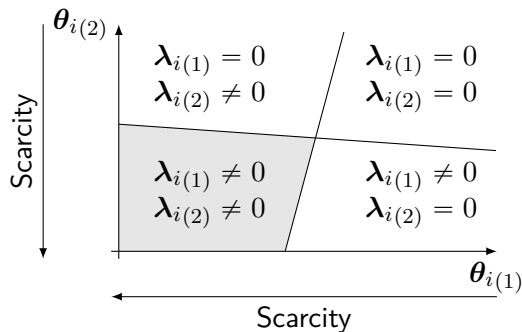
Now, we may have multiple (Piecewise affine function)

$$\lambda_i[k] = \begin{cases} -P_i^{(0)} \theta_i[k] - s_i^{(0)}[k], & \text{if } \theta_i[k] \in \mathcal{R}_{\lambda_i}^0 \\ \vdots & \vdots \\ -P_i^{(Z)} \theta_i[k] - s_i^{(Z)}[k], & \text{if } \theta_i[k] \in \mathcal{R}_{\lambda_i}^Z \end{cases}$$

Still the $P_i^{(z)}$ are time independent

Analyzing System

Solution for $\lambda_i[k]$ (Continued)



Separation surfaces depend on state and local parameters.
Unknown by the coordinator.

Analyzing System

Solution for $\lambda_i[k]$ (Continued) Still?

$$\lambda_i[k] = \begin{cases} -P_i^{(0)} \theta_i[k] - s_i^{(0)}[k], & \text{if } \theta_i[k] \in \mathcal{R}_{\lambda_i}^0 \\ \vdots & \vdots \\ -P_i^{(Z)} \theta_i[k] - s_i^{(Z)}[k], & \text{if } \theta_i[k] \in \mathcal{R}_{\lambda_i}^Z \end{cases} \quad \begin{array}{c} \uparrow \\ \text{Scarcity} \end{array}$$

$$\begin{array}{ll} \text{All constraints active} & -P_i^{(0)} \theta_i[k] - s_i^{(0)}[k] \rightarrow -P_i \theta_i[k] - s_i[k] \\ \text{None constraints active} & -P_i^{(Z)} \theta_i[k] - s_i^{(Z)}[k] \rightarrow \mathbf{0} \end{array}$$

Assumptions

The region $\mathcal{R}_{\lambda_i}^0 \neq \emptyset$ and we known a point $\theta_i^{\emptyset} \in \mathcal{R}_{\lambda_i}^0$

Analyzing System

Under attack!

$$\tilde{\lambda}_i[k] = T_i[k]\lambda_k$$

Parameters are modified. But not the regions' limits

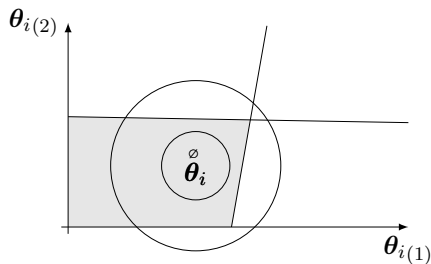
$$\tilde{\lambda}_i[k] = \begin{cases} -\tilde{P}_i^{(0)}\theta_i[k] - \tilde{s}_i^{(0)}[k], & \text{if } \theta_i[k] \in \mathcal{R}^0 \\ \vdots & \vdots \\ -\tilde{P}_i^{(Z)}\theta_i[k] - \tilde{s}_i^{(Z)}[k], & \text{if } \theta_i[k] \in \mathcal{R}_{\lambda_i}^Z \end{cases}$$

- If we can estimate $\tilde{P}_i^{(0)}$ we can use same strategy than before
- Problem: We don't know in which region θ_i is
- Solution: Let's force it using Artificial Scarcity

Artificial Scarcity

What you thought was way too much is not enough

- We use the point θ_i^\emptyset , which activates all constraints²⁰



- Generate points close to θ_i^\emptyset
- Estimate $\hat{P}_i^{(0)}[k]$
- How do we know the radius?
 - Unfortunately we don't.
- How to estimate $\hat{P}_i^{(0)}[k]$ nonetheless?
 - Expectation Maximization

²⁰If we have local constraints, we suppose this point respects them.

Expectation Maximization

- Iterative method to estimate parameters of multimodal models²¹
- We give multiple observations $\theta_i^o[k]$ and $\tilde{\lambda}_i^o[k]$
- At each step we calculate
 - Ⓔ the probability of each $(\hat{P}_i^{(z)}[k], \hat{s}_i^{(z)}[k])$ having generated each $\tilde{\lambda}_i^o[k]$
 - Ⓜ new estimates $(\hat{P}_i^{(z)}[k], \hat{s}_i^{(z)}[k])$ based on the probabilities
- At the end we have
 - Ⓛ Parameters with associated region index
 - Ⓜ Observations with associated region index
- We consult the index associated to θ_i^\emptyset
- We recover the associated parameter, i.e., $\hat{P}_i^{(0)}[k]$

²¹Such as our PWA function after using some tricks

Detection and Mitigation

Same same, but different

Assumption

We estimate nominal $\bar{P}_i^{(0)}$ from attack free negotiation

- Detection

$$\left\| \hat{\bar{P}}_i^{(0)}[k] - \bar{P}_i^{(0)} \right\|_F \geq \epsilon_{P_i^{(0)}}$$

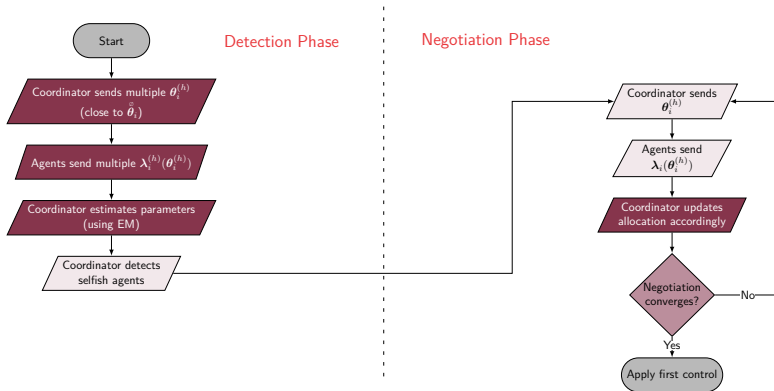
- Mitigation

$$\widehat{T_i[k]^{-1}} = \bar{P}_i^{(0)} \hat{\bar{P}}_i^{(0)}[k]^{-1}.$$

$$\lambda_i^{\text{rec}} = \widehat{T_i[k]^{-1}} \tilde{\lambda}_i.$$

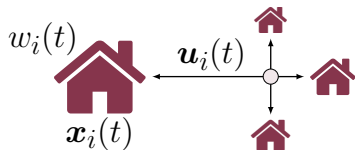
Complete algorithm

RPdMPC-AS²²



²²Nogueira, Bourdais, Leglaive, et al., “Expectation-Maximization Based Defense Mechanism for Distributed Model Predictive Control”.

Example

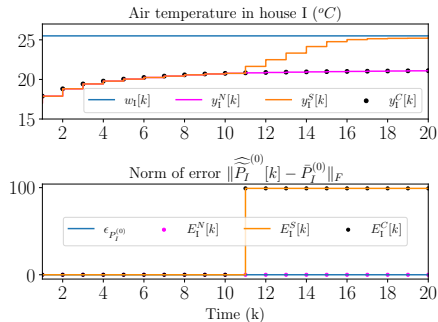


District Heating Network (4 Houses)

- Houses modeled using 3R-2C
- Not enough power ~~Not enough power~~
(Change (x_0, w_0))
- Period of 5h ($T_s = 0.25h \rightarrow k = \{1 : 20\}$)
- Prediction horizon ($N = 4$)
- 3 scenarios
 - Ⓝ Nominal
 - Ⓒ Agent I cheats (dMPC)
 - Ⓢ Agent I cheats (RPdMPC-AS)

Results

Temporal



Temperature in house I.

Error $E_I(k)$.

N Nominal, **S** Selfish **C** Corrected

Results

Costs

Objective functions J_i (Normalized error %)

Agent	Selfish	Corrected
I	-36.49	$-4.12e - 05$
II	35.81	$1.74e - 05$
III	29.22	$2.14e - 05$
IV	37.54	$1.73e - 05$
Global	10.69	$-6e - 07$

Too good to be true!

It's a kind of magic!~~It's a kind of magic!~~

- No disturbance in communication
- Unfortunately EM is not magic
 - Slow convergence
 - Dependency on initialization
 - No guarantees of achieving global optimal
- Some “solutions”:
 - Force some parameters to converge faster (case dependant)
 - Run multiple times with different initialization and pick best
 - Associate with other methods of the same family

Conclusion




Main takeaways

- Distributed MPC
 - increases privacy and flexibility
 - reduces complexity of calculation
 - in security context, it still is in its baby steps
- Primal decomposition
 - prevents agent to use more resources than agreed upon
 - increases privacy by communicating dual variables instead of primal
- Security for DMPC
 - Attacker can change the communication to receive more resources.
 - The consequences of an attack are suboptimality and instability
 - We can explore scarcity information to mitigate

Open questions/Future directions

- Reconstruction with partial information (Current work)
- Study of error propagation (Current work)
- Robustness when add noise
- Estimation as Switched Auto-Regressive Exogenous System
- Sensibility to other topologies (more/less vulnerable?)
- Study of security on similar problems
(flocking/consensus/averaging/federated learning etc)
- ...

For Further Reading I

-  Åström, K.J. and B. Wittenmark. Adaptive Control. Addison-Wesley series in electrical and computer engineering: Control engineering. Addison-Wesley, 1989. ISBN: 9780201097207. DOI: 10.1007/978-3-662-08546-2_24.
-  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>.

Questions? Comments?

Repository

<https://github.com/Accacio/thesis>



Contact

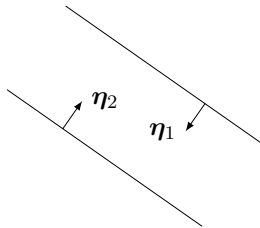
rafael.accacio.nogueira@gmail.com



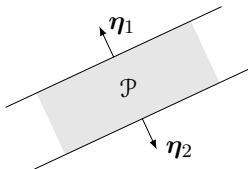
Conditions

◀ back

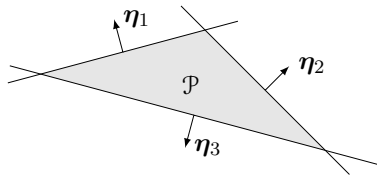
One way to ensure this, is to make the original constraints to form a cone.



No intersection



$$\angle \eta_2 = 180^\circ$$



A 3-sided polyhedron.

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

where

$$\mathcal{A}_\theta = \begin{bmatrix} I - \frac{M-1}{M} \rho^{(p)} P_1 & \frac{1}{M} \rho^{(p)} P_2 & \dots & \frac{1}{M} \rho^{(p)} P_M \\ \frac{1}{M} \rho^{(p)} P_1 & I - \frac{M-1}{M} \rho^{(p)} P_2 & \dots & \frac{1}{M} \rho^{(p)} P_M \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{M} \rho^{(p)} P_1 & \frac{1}{M} \rho^{(p)} P_2 & \dots & I - \frac{M-1}{M} \rho^{(p)} P_M \end{bmatrix}$$
$$\mathcal{B}_\theta[k] = \begin{bmatrix} -\frac{M-1}{M} \rho^{(p)} \mathbf{s}_1[k] + \frac{1}{M} \rho^{(p)} \mathbf{s}_2[k] \dots - \frac{1}{M} \rho^{(p)} \mathbf{s}_M[k] \\ \frac{1}{M} \rho^{(p)} \mathbf{s}_1[k] - \frac{M-1}{M} \rho^{(p)} \mathbf{s}_2[k] \dots - \frac{1}{M} \rho^{(p)} \mathbf{s}_M[k] \\ \vdots \\ \frac{1}{M} \rho^{(p)} \mathbf{s}_1[k] + \frac{1}{M} \rho^{(p)} \mathbf{s}_2[k] \dots - \frac{M-1}{M} \rho^{(p)} \mathbf{s}_M[k] \end{bmatrix}$$

Parameters estimated depending on Prediction Horizon N

constraints depend on # global constraints c and prediction horizon N

- Number of Regions = 2^{Nc}
- Parameters in each region = Matrix $P_i^{(z)} = (Nc)^2$ + vector $\mathbf{s}_i^{(z)}[k] = Nc$
 - Total $((Nc)^2 + Nc)2^{Nc}$

Some examples

- 1 constraint
 - $N = 3 \rightarrow 96$ elements
 - $N = 4 \rightarrow 320$ elements

Remark

*We can reduce number of elements estimated from $P_i^{(z)}$ if we assume $P_i^{(z)} \in \mathbb{S}$
New total $\rightarrow ((Nc)^2 + 3Nc)2^{Nc-1}$*