Distributed and Real-time Predictive Control

Melanie Zeilinger

Christian Conte (ETH)
Alexander Domahidi (ETH)
Ye Pu (EPFL)
Colin Jones (EPFL)

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

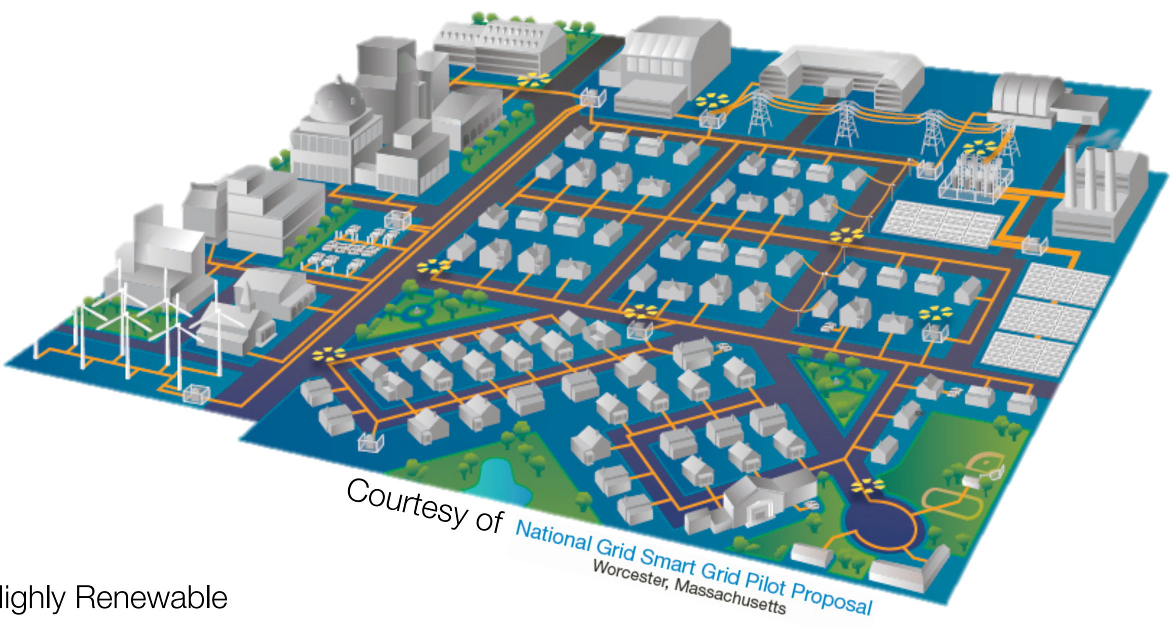
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Challenges in modern control systems

Power system:
- Frequency control
- Voltage control

Customer:
- Control of building networks
- Control of flexible loads and storage capacities

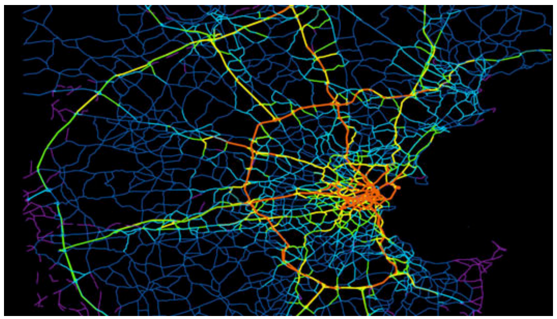
i4Energy seminar: 12pm, 310 SDH
"The Role of Supply-Following Loads in Highly Renewable Electricity Grids", Jay Taneja



- Large-scale, complex system
- Constraints
- Uncertainties
- High performance and safety
- Composed of coupled subsystems
- Often high-speed dynamics
- Computation and communication constraints

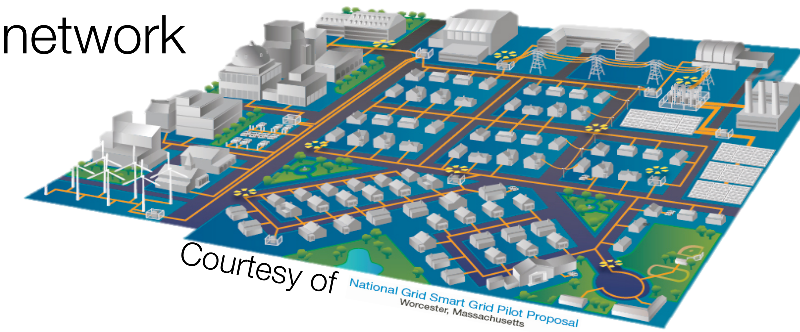
Challenges in modern control systems

Traffic network



Courtesy of Dr. Pu Wang

Power network



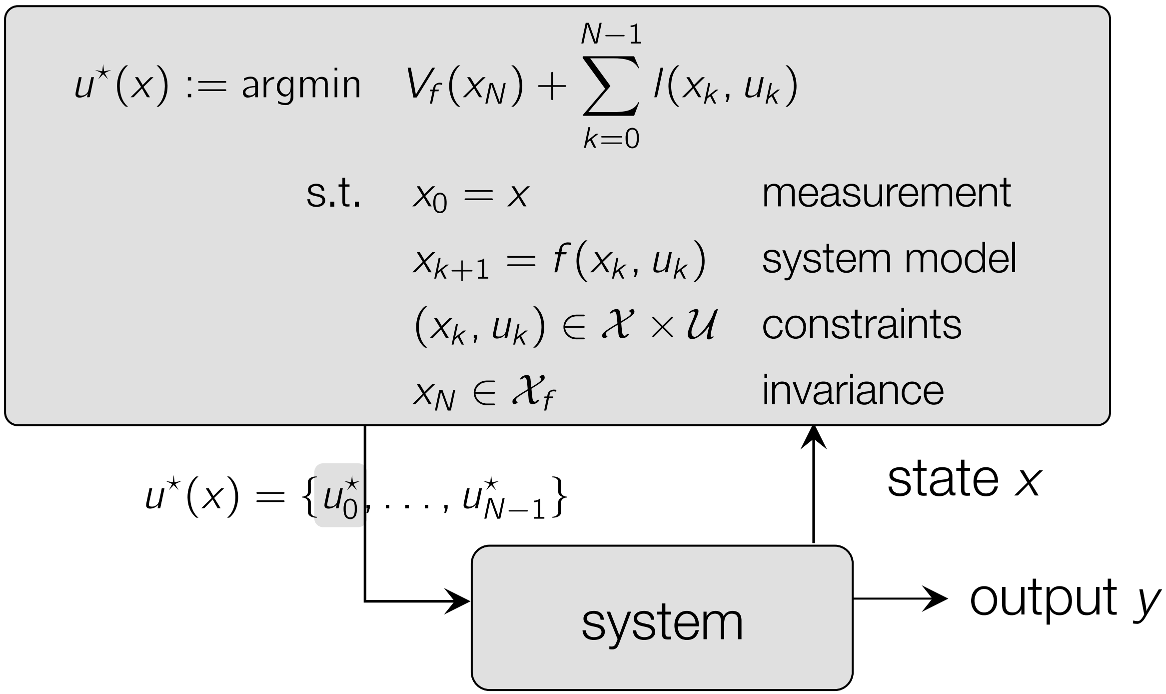
Courtesy of National Grid Smart Grid Pilot Proposal
Worcester, Massachusetts

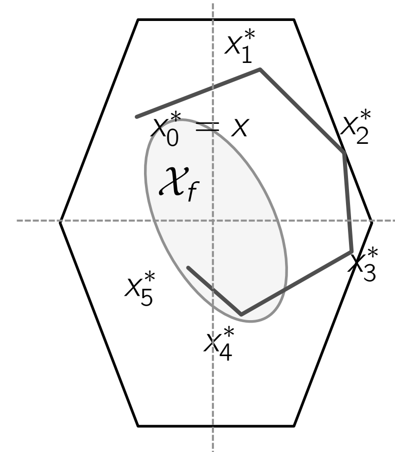
Courtesy of IDSC, ETH

Robotics

- Large-scale, complex system
- Constraints
- Uncertainties
- High performance and safety
- Composed of coupled subsystems
- Often high-speed dynamics
- Computation and communication constraints

Model Predictive Control (MPC)
– A High Performance Method for Constrained Control

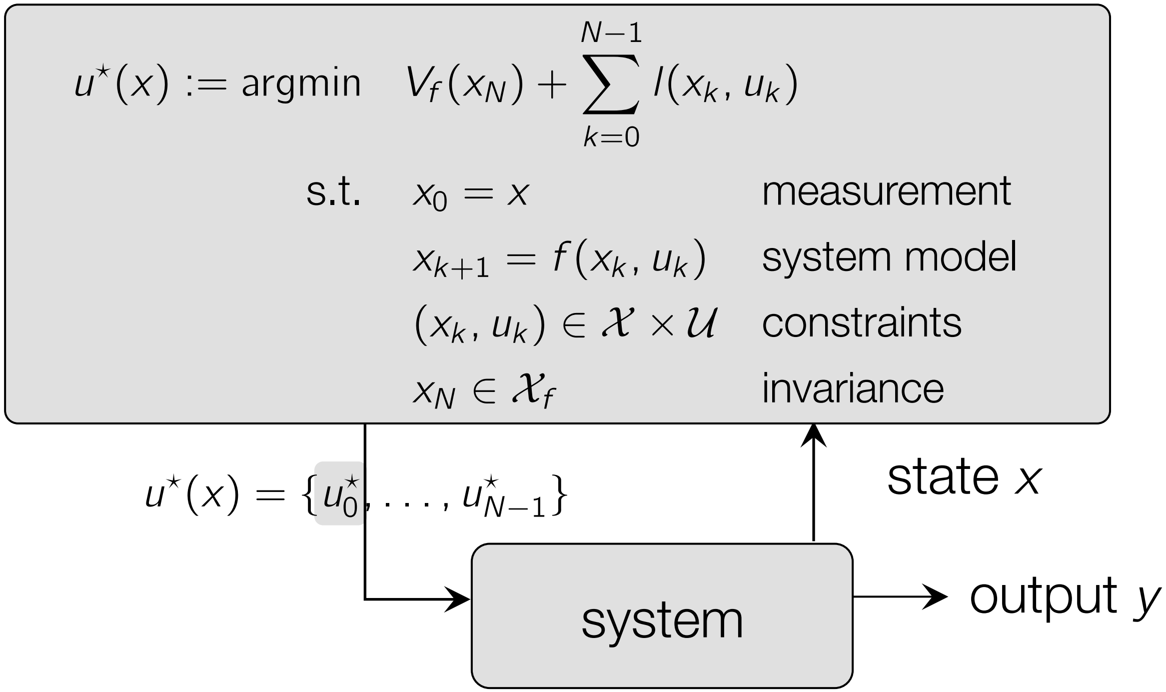




Each sample time:

1. Measure / estimate state
2. Solve optimization problem for entire planning window
3. Implement only the *first* control action

Model Predictive Control (MPC) – A High Performance Method for Constrained Control



Classical MPC theory:

☺ High performance

☺ Recursive constraint satisfaction

☺ Stability by design

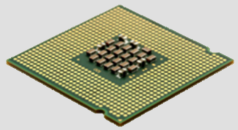
Established approach:

- Optimality
- Terminal cost and constraint

Real-time Model Predictive Control

$$u^\star(x) := \operatorname{argmin} \quad V_f(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k)$$

s.t.	$x_0 = x$	measurement
	$x_{k+1} = f(x_k, u_k)$	system model
	$(x_k, u_k) \in \mathcal{X} \times \mathcal{U}$	constraints
	$x_N \in \mathcal{X}_f$	invariance



Embedded processor

$u^\star(x) = \{u_0^\star, \ldots, u_{N-1}^\star\}$

state x

output y

Classical MPC theory:

☺ High performance

☺ Recursive constraint satisfaction

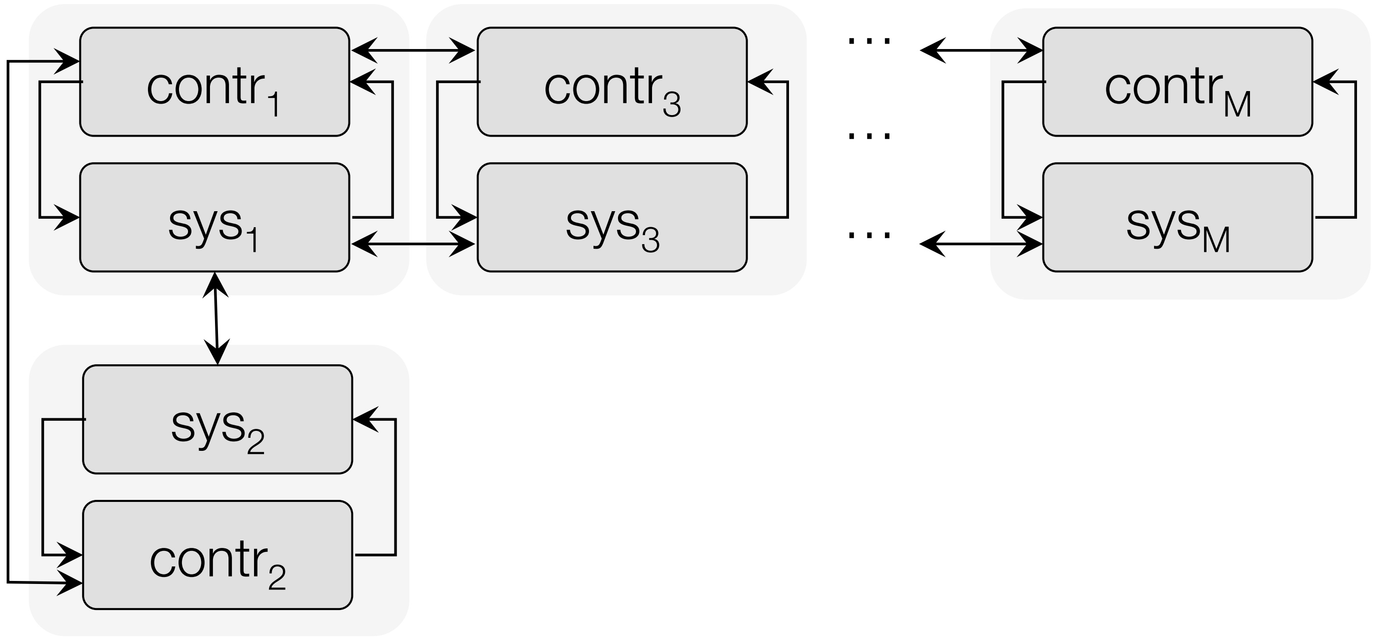
☺ Stability by design

Bounded computation time

→ Early termination

→ Invalidates MPC theory based on optimality

Distributed Model Predictive Control



Classical MPC theory:

☺ High performance

☺ Recursive constraint satisfaction

☺ Stability by design

Local computation and information:

→ Restrictive local terminal conditions

→ Stability in exchange for significant conservatism

Outline: Distributed and Real-time MPC

Centralized MPC theory:

☺ Recursive constraint satisfaction

☺ Stability by design

Established approach:

- Optimality
- Terminal cost and constraint

Real-time MPC:

- Flexibility and fast convergence through interior-point methods
- BUT: Variable solve-times

Outline (Part I):

Stability and constraint satisfaction for any real-time constraint

→ MPC for fast, safety-critical systems

Distributed MPC:

- Reduced conservatism through distributed optimization
- BUT: Global terminal conditions

Outline (Part II):

Stability with larger region of attraction based on local information

→ Plug and Play MPC

Outline: Distributed and Real-time MPC

Centralized MPC theory:

☺ Recursive constraint satisfaction

☺ Stability by design

Established approach:

- Optimality
- Terminal cost and constraint

Real-time MPC:

- Flexibility and fast convergence through interior-point methods
- BUT: Variable solve-times

Outline (Part I):

Stability and constraint satisfaction for any real-time constraint

→ MPC for fast, safety-critical systems

Distributed MPC:

- Reduced conservatism through distributed optimization
- BUT: Global terminal conditions

Outline (Part II):

Stability with larger region of attraction based on local information

→ Plug and Play MPC

Stability and Invariance of Optimal MPC

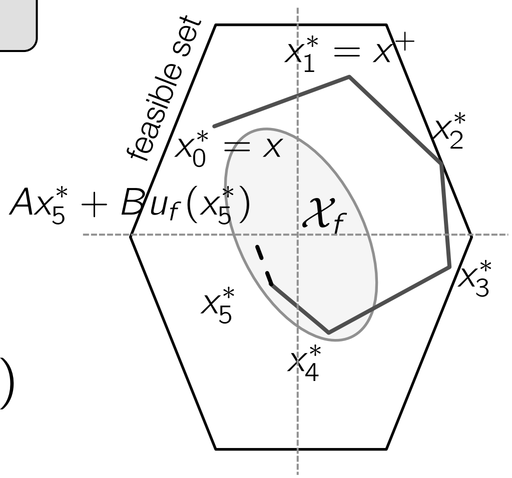
$$
\begin{aligned}
V_N^*(x) = \min \quad & V_N(x,\mathbf{u}) := V_f(x_N) + \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i \\
\text{s.t.} \quad & x_0 = x && \text{measurement} \\
& x_{i+1} = A x_i + B u_i && \text{system model} \\
& C x_i + D u_i \le b && \text{constraints} \\
& x_N \in \mathcal{X}_f && \text{terminal constraint}
\end{aligned}
$$

Assumptions: 1. $\mathcal{X}_f \subset \mathcal{X}$ is invariant

$x \in \mathcal{X}_f \Rightarrow Ax + Bu_f(x) \in \mathcal{X}_f$

2. $V_f(x)$ is a Lyapunov function in $\mathcal{X}_f$

$V_f(Ax + Bu_f(x)) - V_f(x) \le -l(x, u_f(x))$



Theorem:

- $V_N^*(x)$ is a convex Lyapunov function
- The feasible set is invariant under the optimal MPC controller

Stability and Invariance of Optimal MPC

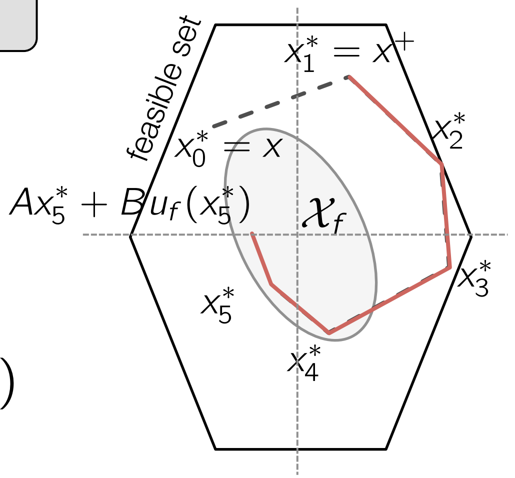
$$
\begin{aligned}
V_N^*(x) = \min \quad & V_N(x, \mathbf{u}) := V_f(x_N) + \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i \\
\text{s.t.} \quad & x_0 = x && \text{measurement} \\
& x_{i+1} = A x_i + B u_i && \text{system model} \\
& C x_i + D u_i \leq b && \text{constraints} \\
& x_N \in \mathcal{X}_f && \text{terminal constraint}
\end{aligned}
$$

Assumptions: 1. $\mathcal{X}_f \subset \mathcal{X}$ is invariant

$x \in \mathcal{X}_f \Rightarrow Ax + Bu_f(x) \in \mathcal{X}_f$

2. $V_f(x)$ is a Lyapunov function in $\mathcal{X}_f$

$V_f(Ax + Bu_f(x)) - V_f(x) \leq -l(x, u_f(x))$



Proof: Shifted sequence $\mathbf{u}^{\text{shift}} = [u_1^*, \ldots, u_{N-1}^*, Kx_N^*]$

- is feasible → Recursive feasibility and invariance
- decreases the cost $V_N^*(x^+) - V_N^*(x) \leq V_N^{\text{shift}}(x^+) - V_N^*(x) \leq -l(x, u_0^*) < 0$

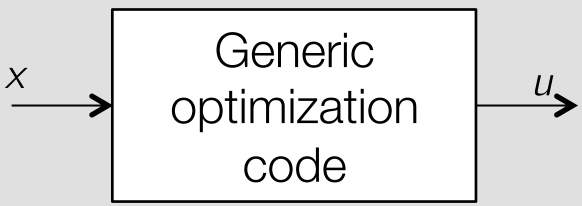
→ $V_N^*(x)$ is a Lyapunov function

Real-Time MPC Controller Synthesis

Large-scale, ms

Interior point methods

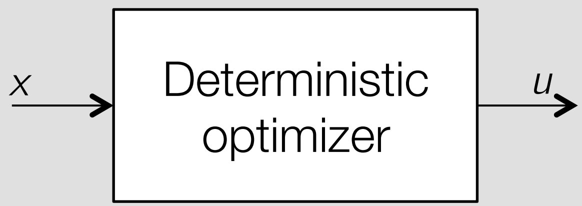
- Modify controller to be robust to time constraints



Medium-scale, us

Gradient approaches

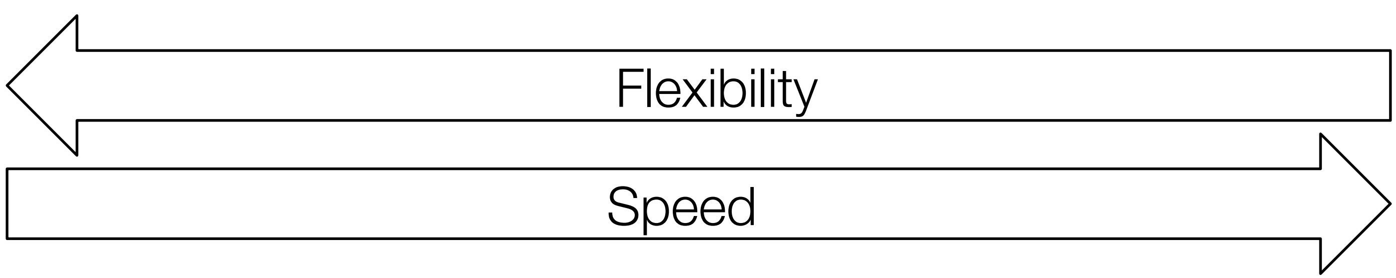
- Bound computation time a priori



Small-scale, ns

Pre-compute controller

- Fixed time online



Ideal approach is problem specific

Real-time MPC using interior-point methods

Real-time online MPC:

Guarantee that

- within the real-time constraint
- a feasible solution
- satisfying stability criteria
- for any admissible initial state

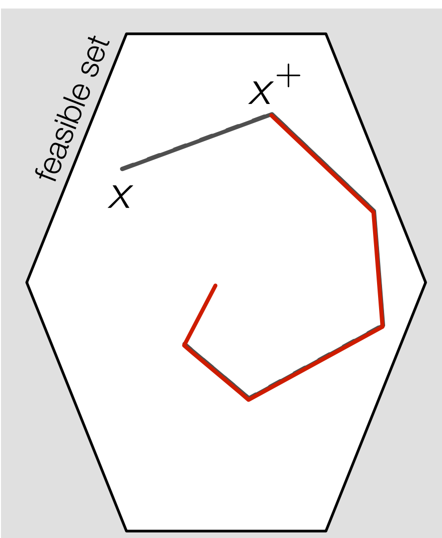
is found.

Real-time MPC using interior-point methods

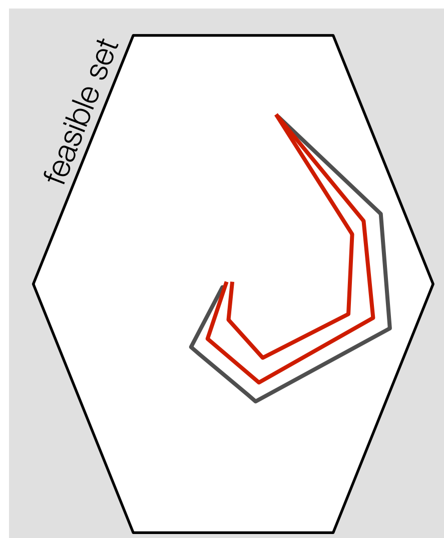
Real-time online MPC:

Guarantee that
- within the real-time constraint ⇐ Early termination
- a feasible solution ⇐ Warm-start
- satisfying stability criteria
- for any admissible initial state

is found.



Warm-start



Online optimization

Suboptimal solution

Real-time MPC using interior-point methods

Real-time online MPC:

Guarantee that

- within the real-time constraint ⇐ Early termination
- a feasible solution ⇐ Warm-start
- satisfying stability criteria
- for any admissible initial state

is found.

Many recent codes have demonstrated that extreme speeds are possible…

OOQP
Object-oriented software for quadratic programming

CVXGEN
Code Generation for Convex Optimization

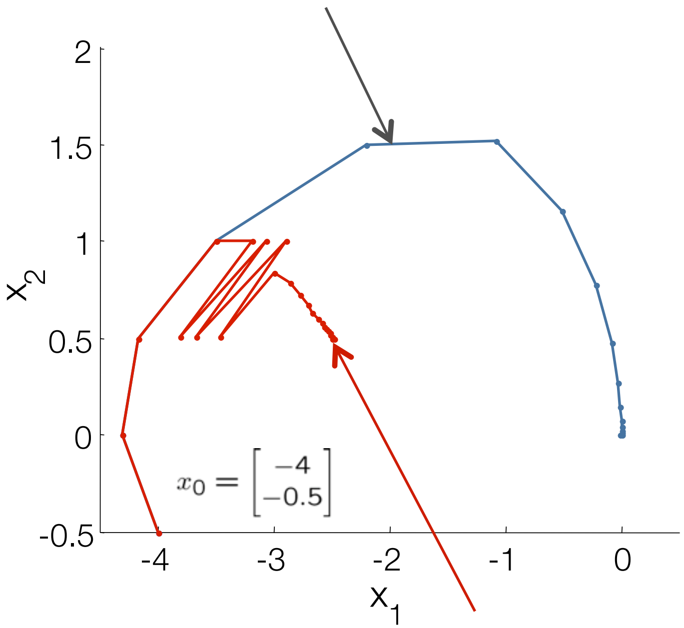
qpOases
Online Active Set Strategy

QPSchur
A dual, active-set, Schur-complement method for quadratic programming

… but cannot guarantee stability in a real-time setting!

Example: Effect of limited computation time

Closed loop trajectory:
Optimal control law



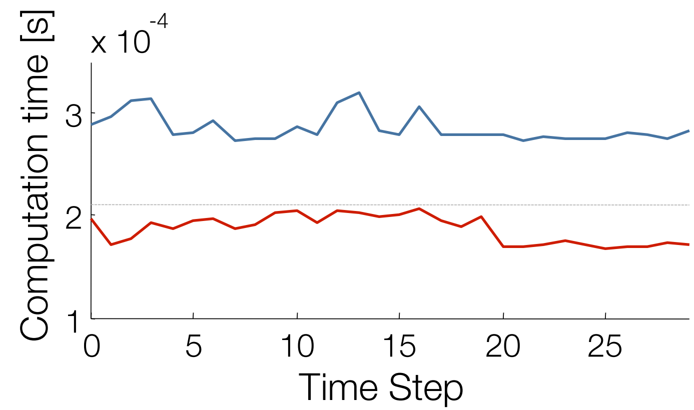
Closed loop trajectory:
Optimization stopped after 4 iterations
= max computation time of 21ms

Unstable example

$$x^+ = \begin{bmatrix} 1.2 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u$$

$|x_1| \leq 5, -5 \leq x_2 \leq 1$

$|u| \leq 1, N = 5, Q = I, R = 1$

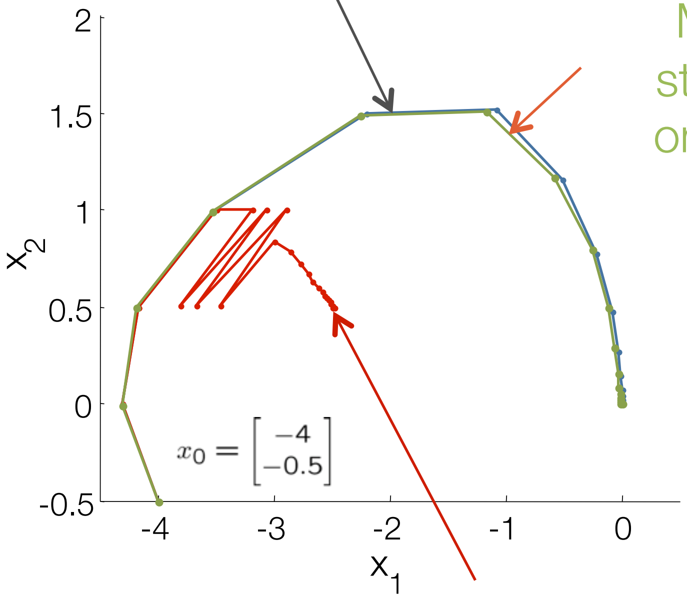


Limited computation time => No stability properties

Example: Stability under proposed real-time method

Closed loop trajectory:
Optimal control law

Proposed real-time MPC method stopped after 4 online iterations



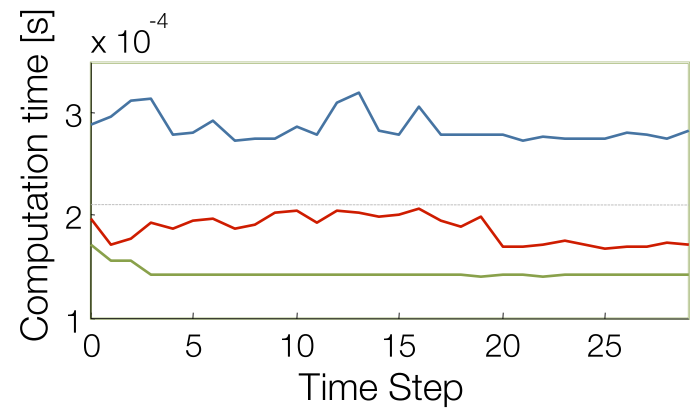
Closed loop trajectory:
Optimization stopped after 4 iterations
= max computation time of 21ms

Unstable example

$$x^+ = \begin{bmatrix} 1.2 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u$$

$$|x_1| \leq 5, -5 \leq x_2 \leq 1$$

$$|u| \leq 1, N = 5, Q = I, R = 1$$



Real-time robust MPC : Nearly optimal and satisfies time constraints

Loss of stability guarantee in real-time

Requirement for stability: Lyapunov function

→ Use of MPC cost as Lyapunov function

→ Key condition: Decrease of MPC cost at every time step

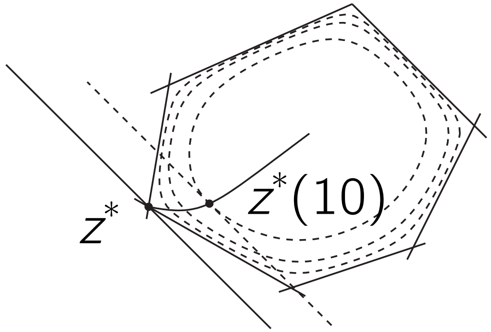
$$V_N(x_t, \mathbf{u}_t) < V_N(x_{t-1}, \mathbf{u}_{t-1})$$

Using interior-point methods this condition can be violated even when initializing with a stabilizing sequence, e.g. the shifted sequence

Example: Barrier interior-point method

Minimize augmented cost

$$\begin{aligned} \min_z \quad & f(z) \\ \text{s.t.} \quad & Fz = Ex \\ & Gz \leq d \end{aligned} \quad \Rightarrow \quad \begin{aligned} \min_z \quad & f(z) - \mu \sum_{i=1}^{m} \log(-G_i z + d_i) \\ \text{s.t.} \quad & Fz = Ex \end{aligned}$$



→ Decrease in augmented cost does not enforce a decrease in MPC cost

→ Steady-state offset for $\mu \neq 0$

Real-time stability guarantees

Goal: Ensure that suboptimal cost is Lyapunov function

Introduce 'Lyapunov constraint':

Enforces decrease in suboptimal MPC cost *at each iteration*

$$V_N(x_t^{\text{nom}}, \mathbf{u}_t) \leq V_N(x_{t-1}, \mathbf{u}_{t-1}) - \epsilon \|x_{t-1}\|_Q^2 \qquad \text{(Quadratic constraint)}$$

Theorem:

Suboptimal cost for any feasible solution to real-time problem provides Lyapunov function

→ Stability for any real-time constraint

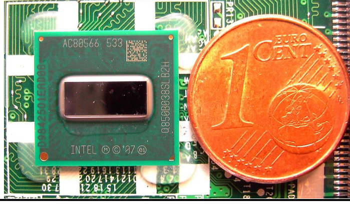
If …

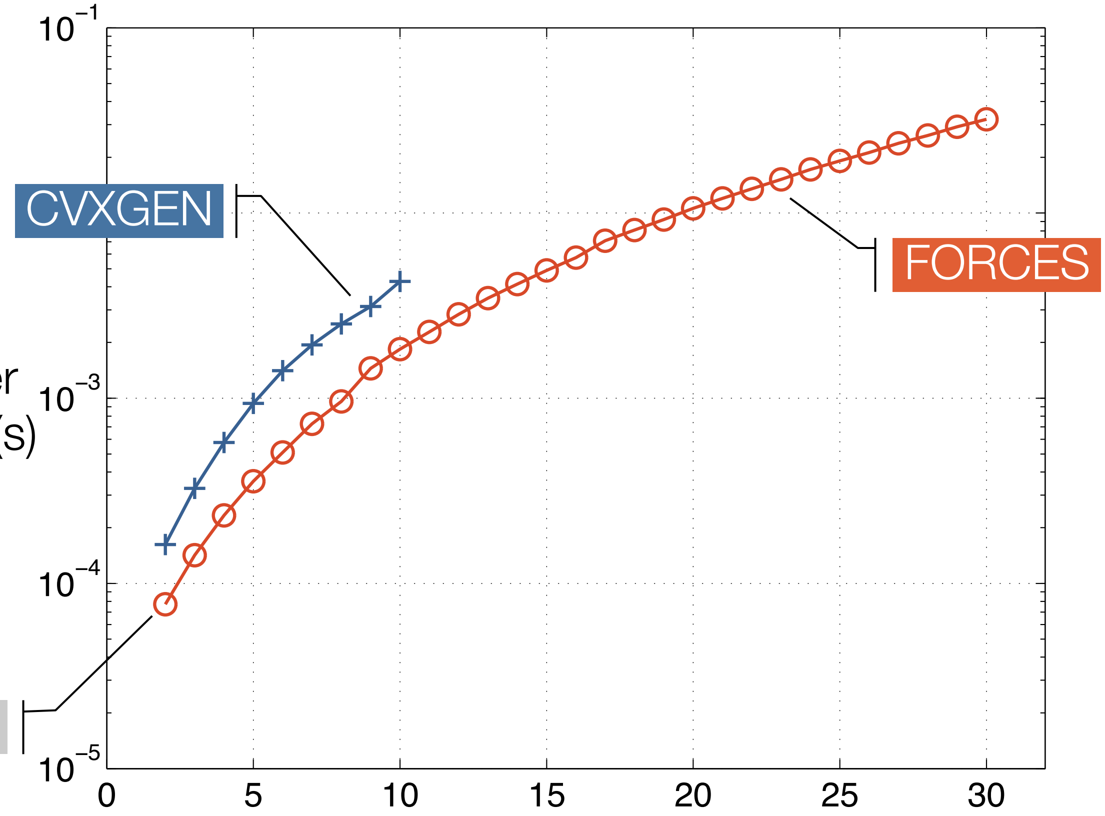
- We can provide (strictly) feasible solution for Lyapunov constraint in real-time
  Key: Ensure that epsilon progress is always possible without optimization

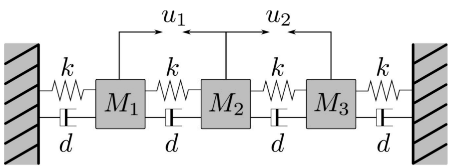
→ Technique based on warm-starting from previous sampling time

- We can solve quadratically constrained QPs with modified structure

Computation times on Intel Atom for QP







Oscillating masses:

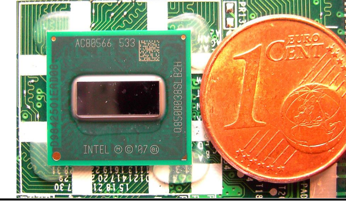
QP with
- box constraints
- diagonal cost

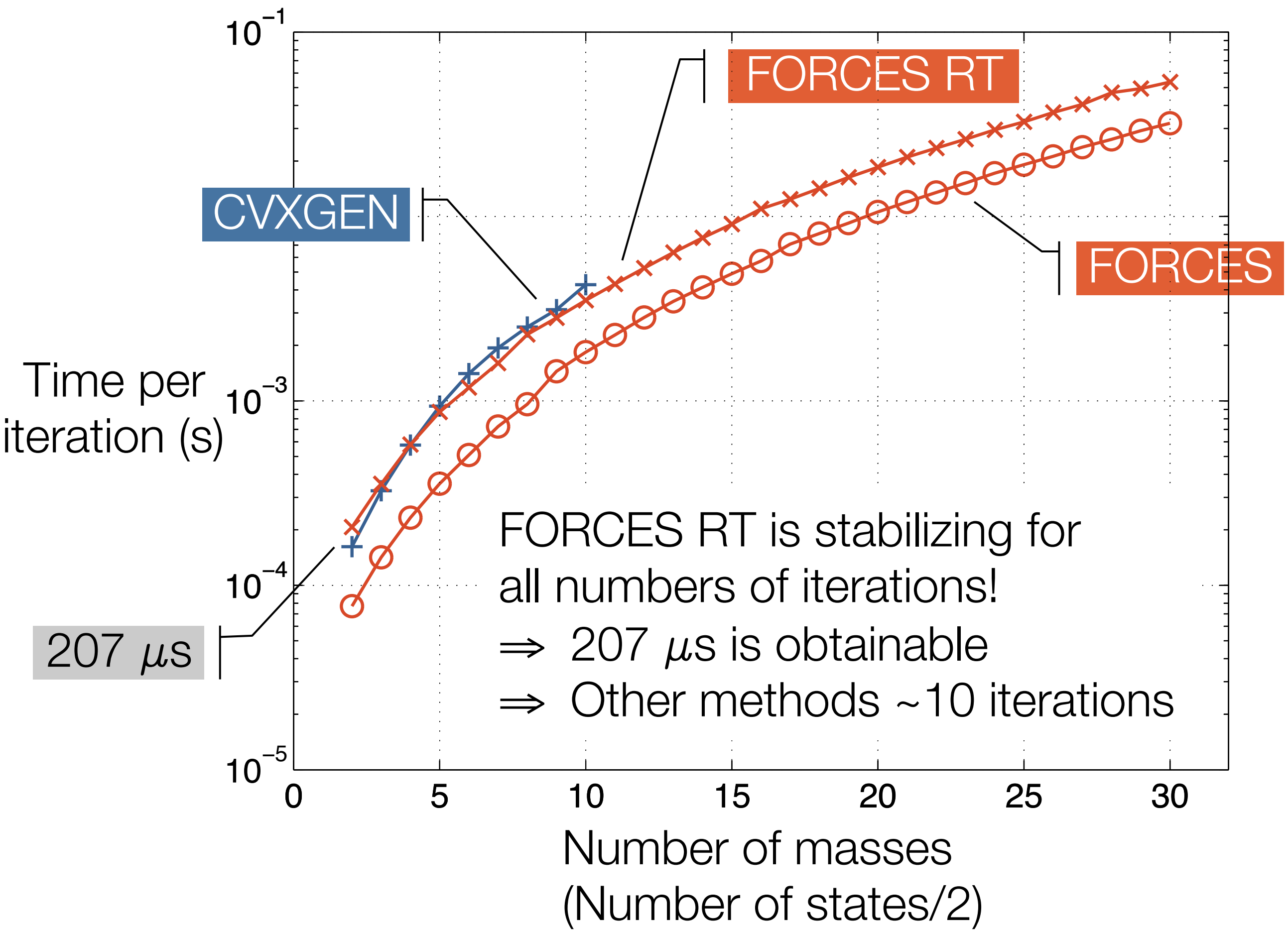
FORCES

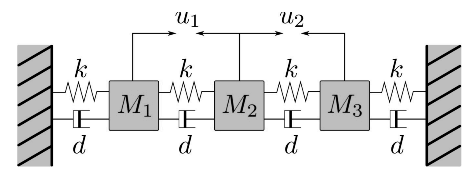
forces.ethz.ch

More details in *[Domahidi, et al., ACC 2012].*

Computation times on Intel Atom for QP







Oscillating masses:

QP with
- box constraints
- diagonal cost

QCQP with
- quadr. terminal set
- real-time constr.

More details in *[Domahidi, et al., ACC 2012].*

FORCES

forces.ethz.ch

Summary: Real-Time MPC

Real-time online MPC:

Guarantee that

- within the real-time constraint ⇐ Early termination
- a feasible solution ⇐ Warm-start
- satisfying stability criteria ⇐ Lyapunov constraint
- for any admissible initial state

is found.

- Optimal MPC requires unknown computation time

  → Fast systems require theory of real-time MPC
- Real-time method provides stability guarantees for arbitrary time constraints
- Extension to robust tube-based MPC
- Extension to tracking (more involved)
- Possible to achieve millisecond solve-times on inexpensive hardware
- Real-time MPC still faster than solvers without guarantees

[Zeilinger, et al., Automatica 2013, accepted], [Domahidi, et al., CDC 2012]

Outline: Distributed and Real-time MPC

Centralized MPC theory:

☺ Recursive constraint satisfaction

☺ Stability by design

Established approach:

- Optimality
- Terminal cost and constraint

Real-time MPC:

- Flexibility and fast convergence through interior-point methods
- BUT: Variable solve-times

Outline (Part I):

Stability and constraint satisfaction for any real-time constraint

Distributed MPC:

- Reduced conservatism through distributed optimization
- BUT: Global terminal conditions

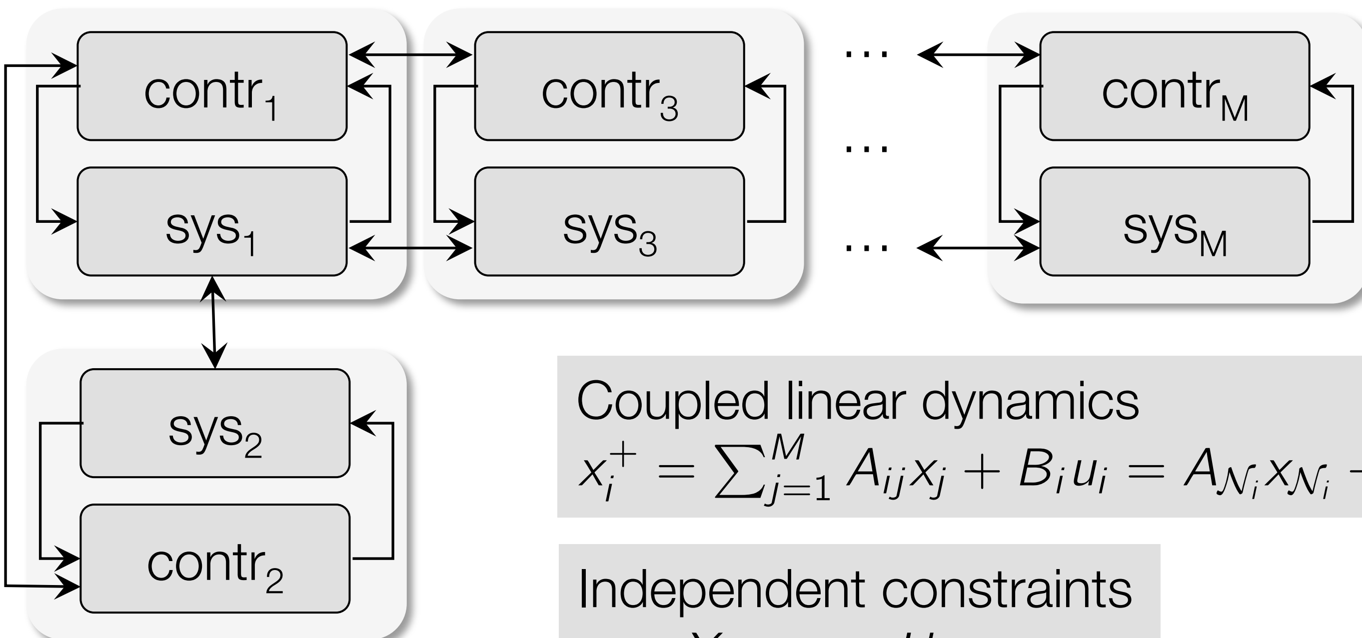
Outline (Part II):

Stability with larger region of attraction based on local information

Distributed Model Predictive Control (MPC)

Communication with neighbours $\mathcal{N}_i$

Cooperative objective
$l(x, u) = \sum_{i=1}^{M} l_i(x_i, u_i)$



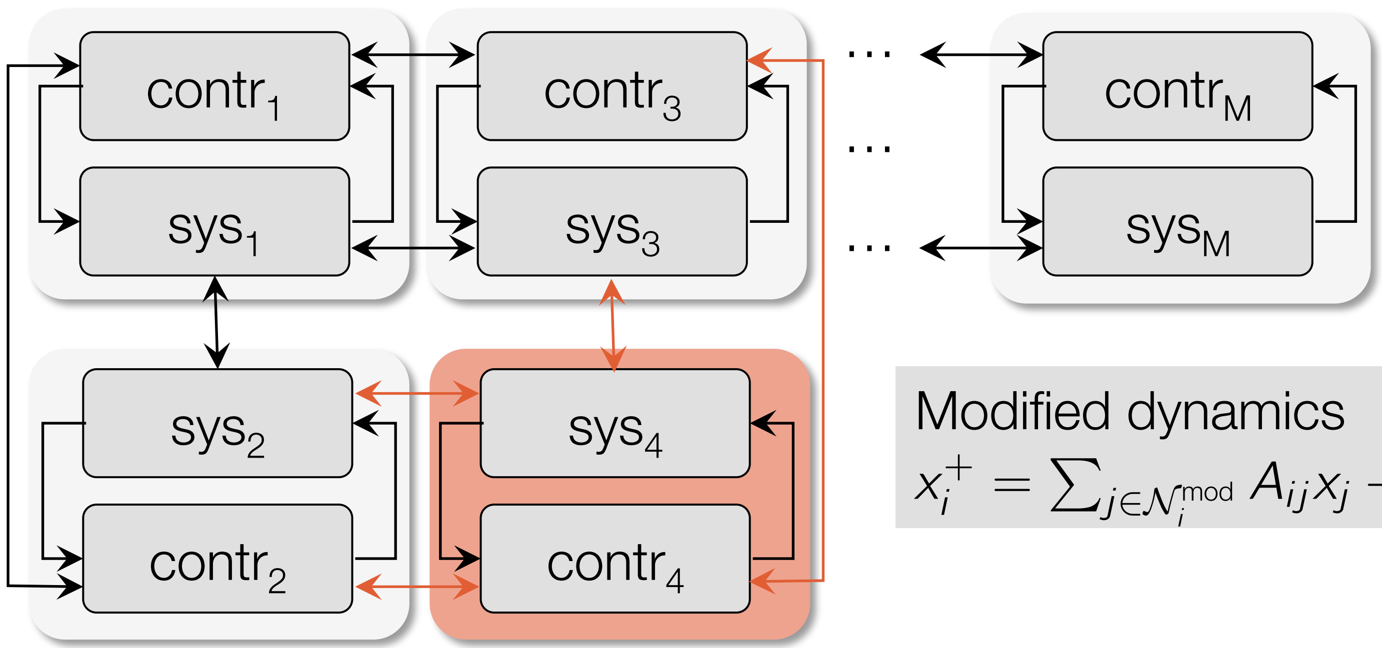
Coupled linear dynamics
$x_i^+ = \sum_{j=1}^{M} A_{ij} x_j + B_i u_i = A_{\mathcal{N}_i} x_{\mathcal{N}_i} + B_i u_i$

Independent constraints
$x_i \in X_i \quad u_i \in U_i$

How to ensure stability and constraint satisfaction without central coordination?

Distributed Model Predictive Control (MPC)

Plug and Play MPC:
Allow subsystems to join or leave the network



Modified dynamics
$x_i^+ = \sum_{j \in \mathcal{N}_i^{\text{mod}}} A_{ij} x_j + B_i u_i$

How to maintain stability and constraint satisfaction during network changes?

Distributed Optimization Requires Structure

$$\min \sum f_i(y_i)$$
$$\text{s.t. } y_i \in Y_i$$
$$\sum A_i y_i = c$$

Distributed optimization requires that the problem is structured

Example: Dual Decomposition

$$g(\lambda) = \min_{y_i \in Y_i} \sum f_i(y_i) + \lambda^T \left(\sum A_i y_i - c \right) = \sum \min_{y_i \in Y_i} f_i(y_i) + \lambda^T A_i y_i$$

Gradient of the dual function: $\nabla g(\lambda) = \sum A_i y_i^\star(\lambda) - c$

Gradient-based approach

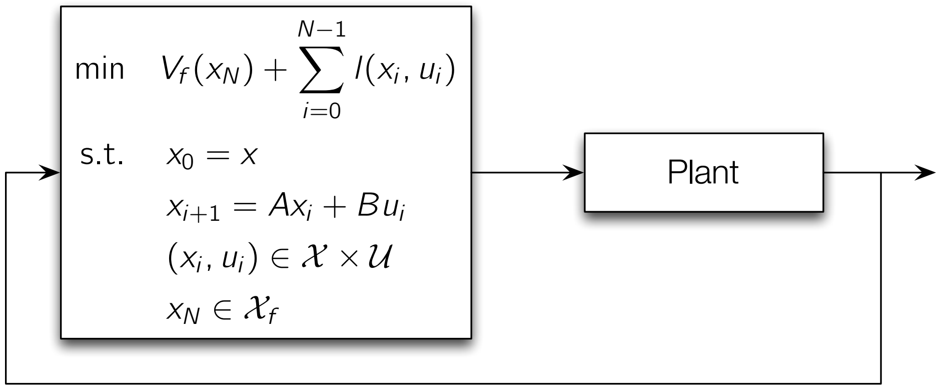
$$\lambda^+ = \lambda + \alpha \nabla g(\lambda)$$

Optimal values y_i^* → Local optimization

Dual update → Consensus

Many variants on this theme (ADMM, AMA,...)

Two Conflicting Requirements



$A, \; B$ structured

$\mathcal{X}, \; \mathcal{U}$ distributed

$l(x, u)$ distributed

1 Stability and invariance if:

1. $\mathcal{X}_f \subset \mathcal{X}$ is invariant — Dense
   $x \in \mathcal{X}_f \Rightarrow Ax + Bu_f(x) \in \mathcal{X}_f$
2. $V_f(x)$ is a Lyapunov function in $\mathcal{X}_f$ — Dense
   $V_f(Ax + Bu_f(x)) - V_f(x) \leq -l(x, u_f(x))$

2 Structured optimization

Terminal cost & constraints:

$$\mathcal{X}_f = \mathcal{X}_f^1 \times \cdots \times \mathcal{X}_f$$

$$V_f(x) = \sum_{k=1}^{M} V_f^k(x_{\mathcal{N}_k})$$

$$u_f(x) = [u_f^1(x_{\mathcal{N}_1}), \ldots, u_f^M(x_{\mathcal{N}_M})]^T$$

Goal: Satisfy both requirements without central coordination

→ Online & offline optimization structured according to system coupling

Structured Lyapunov Function

Lyapunov requirement: $V_f(x^+) - V_f(x) \leq -l(x, u_f(x))$

Structure requirement: $V_f(x) = V_f^1(x_1) + \cdots + V_f^M(x_M)$

Idea: Allow local increase while requiring a global decrease

Theorem: *[Jokic, Lazar, 2009]*

$V_f(x) := \sum_{i=1}^{M} V_f^i(x_{\mathcal{N}_i})$ is a Lyapunov function if

$$V_f^i(x_i^+) - V_f^i(x_i) \leq -l_i(x_{\mathcal{N}_i}) + \gamma_i(x_{\mathcal{N}_i})$$

Possible local increase

$$\sum_{i=1}^{M} \gamma_i(x_{\mathcal{N}_i}) \leq 0$$

Global decrease

☺ Global Lyapunov function → Stability

Structured Invariant Set

Invariance requirement: $x \in \mathcal{X}_f \Rightarrow x^+ \in \mathcal{X}_f$

Structure requirement: $\mathcal{X}_f(\alpha) = \mathcal{X}_f^1(\alpha_1) \times \cdots \times \mathcal{X}_f^M(\alpha_M)$

Idea: Level sets of a Lyapunov function are invariant

$$\mathcal{X}_f = \left\{ x \;\middle|\; V_f(x) = \sum_{i=0}^{M} V_f^i(x_{\mathcal{N}_i}) \leq \bar{\alpha} \right\}$$

Problem: This terminal constraint couples all sub-systems

Want a condition that can be tested in a distributed fashion

$$\mathcal{X}_f^i(\alpha_i) = \{x \mid V_f^i(x_{\mathcal{N}_i}) \leq \alpha_i\} \text{ where } \sum_{i=0}^{M} \alpha_i = \alpha \leq \bar{\alpha}$$

Problem: Static sets $\mathcal{X}_f^i(\alpha_i)$ are not invariant...

$$V_f(x_i) \leq \alpha_i \not\Rightarrow V_f(x_i^+) \leq \alpha_i \quad \text{, since}$$

$$V_f^i(x_i^+) - V_f^i(x_i) \leq -l_i(x_{\mathcal{N}_i}, u_f^i(x_{\mathcal{N}_i})) + \gamma_i(x_{\mathcal{N}_i}) \not\leq 0$$

Structured Dynamic Invariant Set

Invariance requirement: $x \in \mathcal{X}_f \Rightarrow x^+ \in \mathcal{X}_f$

Structure requirement: $\mathcal{X}_f(\alpha) = \mathcal{X}_f^1(\alpha_1) \times \cdots \times \mathcal{X}_f^M(\alpha_M)$

- Define auxiliary dynamics, with the same structure as the system dynamics:

$$\alpha_i^+ = \alpha_i + \gamma_i(x_{\mathcal{N}_i})$$

- Choose initial α_i such that $\sum \alpha_i \leq \bar{\alpha}$, $\left\{x \,\middle|\, \sum V_f^i(x_{\mathcal{N}_i}) \leq \bar{\alpha}\right\} \subseteq X$

Theorem:

1. Time-varying terminal set $\mathcal{X}_f^i(\alpha_i) = \{x \mid V_f^i(x_i) \leq \alpha_i\}$ is invariant

$$x_i \in \mathcal{X}_f^i(\alpha_i) \Rightarrow x_i^+ \in \mathcal{X}_f^i(\alpha_i^+)$$

2. All state and input constraints are satisfied in $\mathcal{X}_f(\alpha)$

Proof: From $\sum \alpha_i^+ = \sum \alpha_i + \sum \gamma_i(x_{\mathcal{N}_i}) \leq \sum \alpha_i$

1. $V_f^i(x_i^+) \leq V_f^i(x_i) - l_i(x_{\mathcal{N}_i}, u_f^i(x_{\mathcal{N}_i})) + \gamma_i(x_{\mathcal{N}_i}) \leq \alpha_i + \gamma_i(x_{\mathcal{N}_i}) = \alpha_i^+$

2. $\mathcal{X}_f(\alpha) \subseteq X \Rightarrow \mathcal{X}_f(\alpha^+) \subseteq X$

Structured Dynamic Invariant Set

Invariance requirement: $x \in \mathcal{X}_f \Rightarrow x^+ \in \mathcal{X}_f$

Structure requirement: $\mathcal{X}_f(\alpha) = \mathcal{X}_f^1(\alpha_1) \times \cdots \times \mathcal{X}_f^M(\alpha_M)$

- Define auxiliary dynamics, with the same structure as the system dynamics:

$$\alpha_i^+ = \alpha_i + \gamma_i(x_{\mathcal{N}_i})$$

- Choose initial α_i such that $\sum \alpha_i \leq \bar{\alpha}$, $\left\{x \,\middle|\, \sum V_f^i(x_{\mathcal{N}_i}) \leq \bar{\alpha}\right\} \subseteq X$

Theorem:

1. Time-varying terminal set $\mathcal{X}_f^i(\alpha_i) = \{x \mid V_f^i(x_i) \leq \alpha_i\}$ is invariant

$$x_i \in \mathcal{X}_f^i(\alpha_i) \Rightarrow x_i^+ \in \mathcal{X}_f^i(\alpha_i^+)$$

2. All state and input constraints are satisfied in $\mathcal{X}_f(\alpha)$

☺ Recursive feasibility

Distributed MPC – Online Control

Structured MPC problem

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{M} V_f^i(x_i(N)) + \sum_{k=0}^{N-1} l(x_i(k), u_i(k)) \\
\text{s.t.} \quad & x_i(0) = x_i \\
& x_i(k+1) = A_{ii} x_i(k) + B_i u_i(k) + \sum_{j \in \mathcal{N}_i} A_{ij} x_j(k) \\
& (x_i(k), u_i(k)) \in \mathcal{X}^i \times \mathcal{U}^i \\
& x_i(N) \in \mathcal{X}_f^i(\alpha_i)
\end{aligned}
$$

Distributed control (online for every subsystem):

1. Measure state
2. Solve global MPC problem by distributed optimization, apply input u_i
3. Update $\alpha_i^+ = \alpha_i + \gamma(x_{\mathcal{N}_i})$

Distributed MPC - Synthesis and Online Control

Distributed synthesis in the linear quadratic case (offline):

1. Solve distributed LMI to compute:
   - Local relaxed Lyapunov functions $V_i^f(x_i) = x_i^T P_i x_i$
   - Indefinite coupling $\gamma_i(x_{\mathcal{N}_i}) = x_{\mathcal{N}_i}^T \Gamma_i x_{\mathcal{N}_i}$
   - Local linear control laws $u_i^f(x_{\mathcal{N}_i}) = K_{\mathcal{N}_i} x_{\mathcal{N}_i}$
2. Solve distributed LP to compute initial feasible terminal size $\bar{\alpha}$

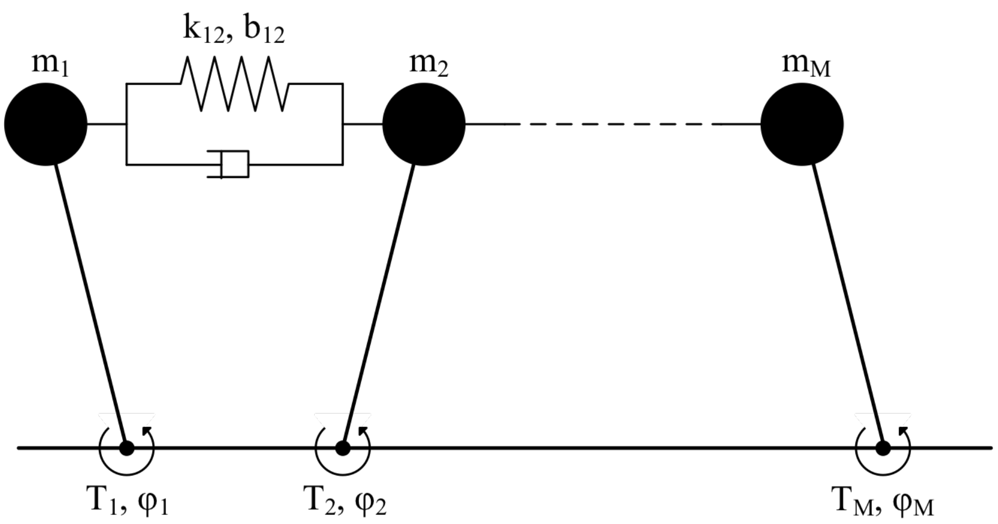
Distributed control (online for every subsystem):

1. Measure state
2. Solve global MPC problem by distributed optimization, apply input u_i
3. Update $\alpha_i^+ = \alpha_i + x_{\mathcal{N}_i}^T(N)\Gamma_{\mathcal{N}_i} x_{\mathcal{N}_i}(N)$

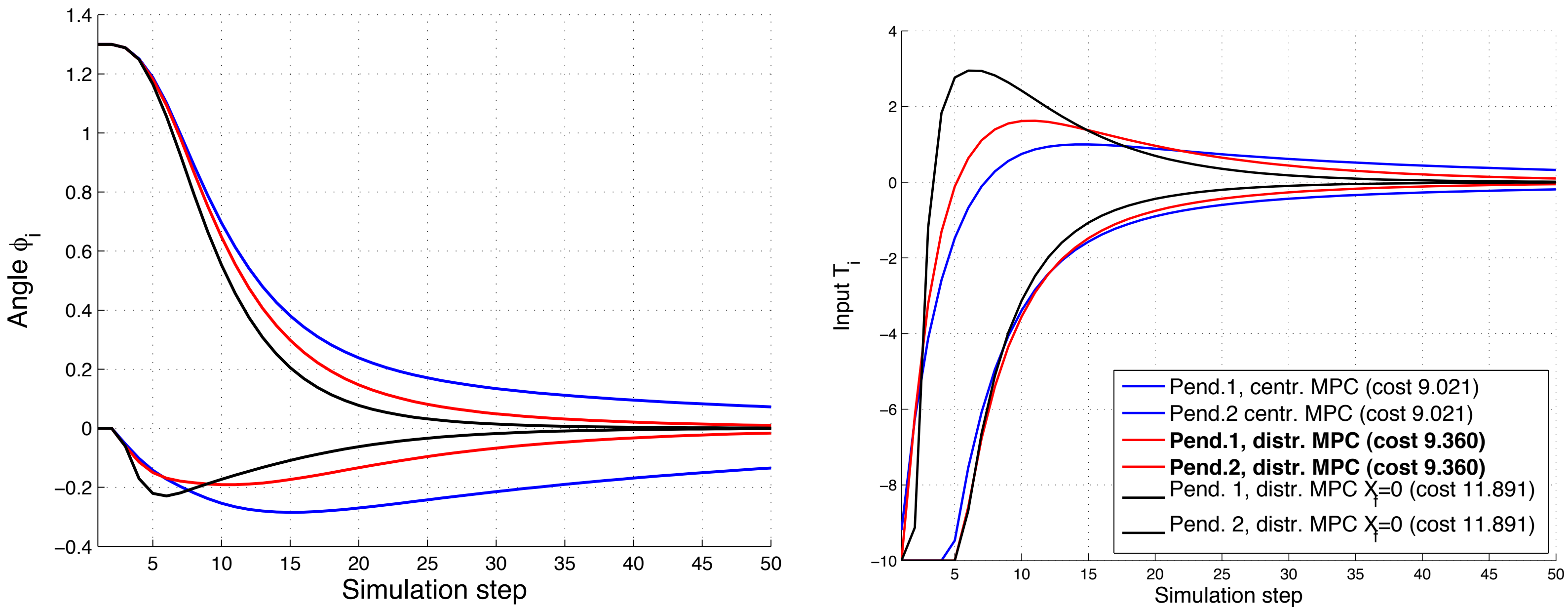
No central coordination required!

Computational example

- Chain of inverted pendulums (unstable)
- Linearized around the origin
- States: Angle and angular velocity of each pendulum
- Inputs: Torque at each pivot

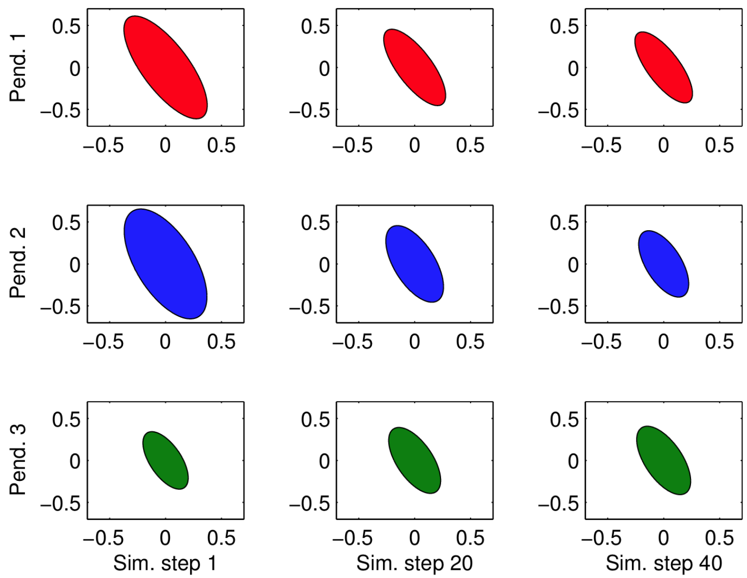


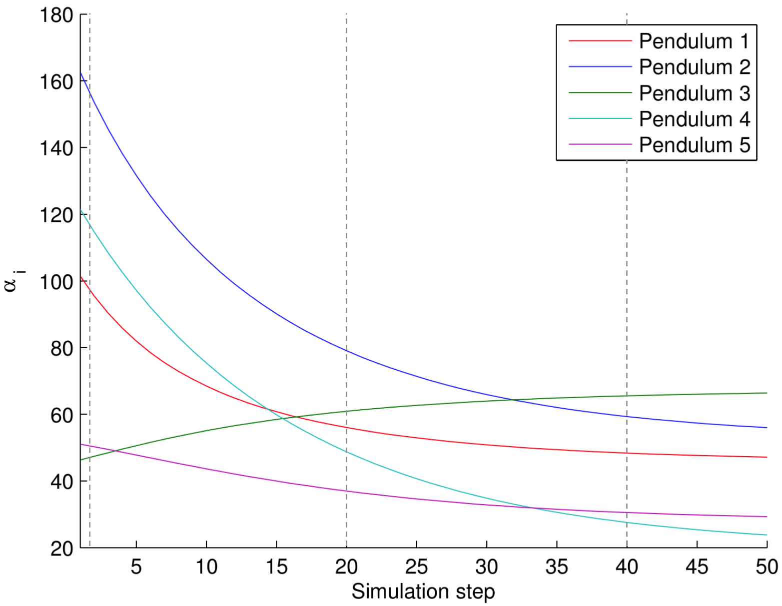
Computational example – Closed-Loop Simulation



- 5 Pendulums, alternating direction method of multipliers, 100 iterations.
- Initially all pendulums in origin, only pendulum 1 is deflected.
- Cost of proposed method only **4%** higher than centralized MPC and **21%** lower than for a trivial terminal set.

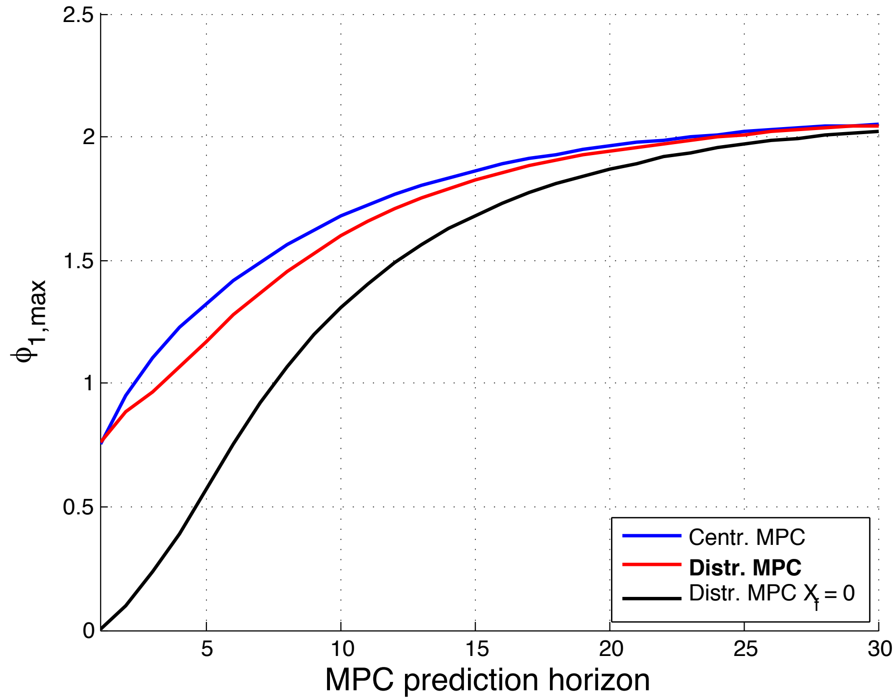
Computational example – Local Terminal Sets





Sizes of local terminal sets change dynamically

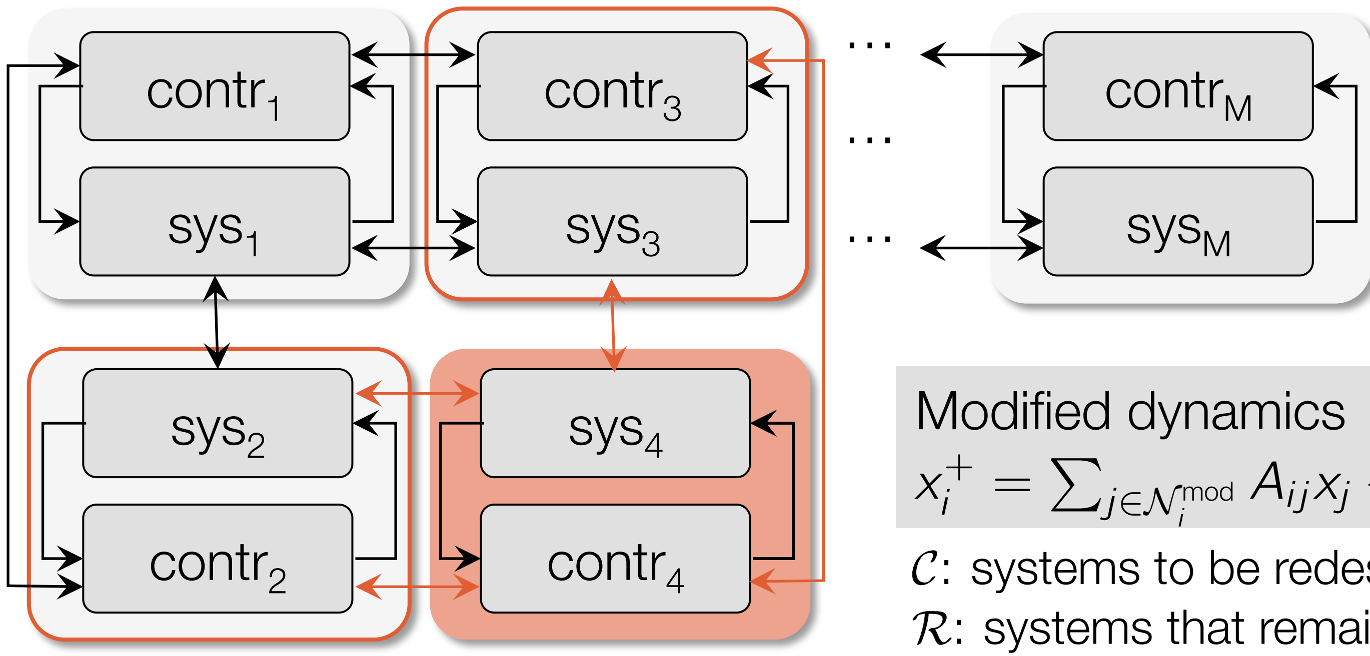
Computational example – Region of Attraction



- Maximum feasible deflection of the first pendulum vs. prediction horizons
- *Short prediction horizons*: Region of attraction for proposed method significantly larger than for trivial terminal set
- *Long prediction horizons*: All methods converge to the same maximum control invariant set

Distributed Model Predictive Control (MPC)

Plug and Play MPC:
Allow subsystems to join or leave the network



Modified dynamics

$x_i^+ = \sum_{j \in \mathcal{N}_i^{\text{mod}}} A_{ij} x_j + B_i u_i$

$\mathcal{C}$: systems to be redesigned

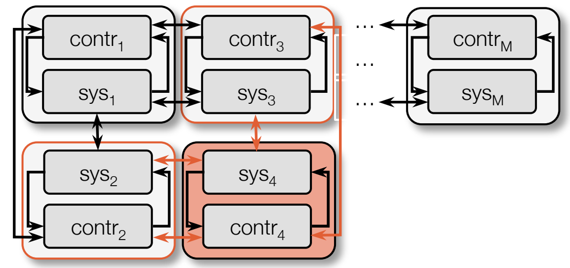
$\mathcal{R}$: systems that remain unchanged

Maintain stability and recursive feasibility during network changes:

- Adapt local control laws of subsystems and neighbours
- Ensure feasibility of the modified control laws

Preparation for Plug and Play operation:

Plug and Play operation
= subsystems join/leave network and modified local control law is applied



Redesign Phase: Adapt local control laws of subsystems and neighbours

- Compute new local terminal costs $\tilde{V}_f^i(x_i)$ and constraint sets $\tilde{\mathcal{X}}_f^i(\alpha_i)$ for (virtually) modified network

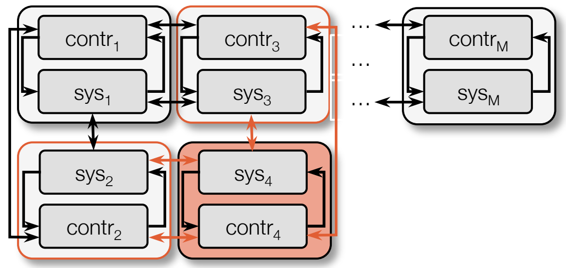
Transition Phase: Ensure feasibility of the modified MPC problem

- Compute a steady-state for Plug and Play operation such that:
  - Steady-state is a feasible initial state for the modified MPC problem
  - System can be controlled to the steady-state from the current state
- *If* steady-state can be found
  - Control system to steady-state
  - Permit plug and play operation

  Else
  - Reject plug and play operation

Preparation for Plug and Play operation:

Plug and Play operation
= subsystems join/leave network and modified
local control law is applied



Redesign Phase: Adapt local control laws of subsystems and neighbours

- Compute new local terminal costs $\tilde{V}_f^i(x_i)$ and constraint sets $\tilde{\mathcal{X}}_f^i(\alpha_i)$ for (virtually) modified network

Transition Phase: Ensure feasibility of the modified MPC problem

- Compute a steady-state for Plug and Play operation such that:
  - Steady-state is a feasible initial state for the modified MPC problem
  - System can be controlled to the steady-state from the current state
- *If* steady-state can be found
  - Control system to steady-state
  - Permit plug and play operation

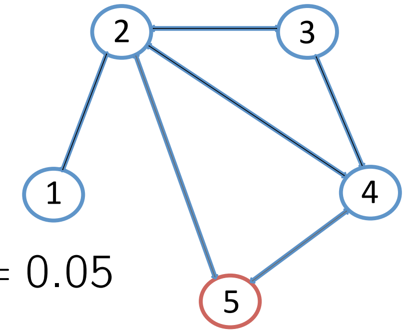
  Else
  - Reject plug and play operation

Plug and and play
synthesis and control
via distributed optimization

Computational example – Area Generation Control

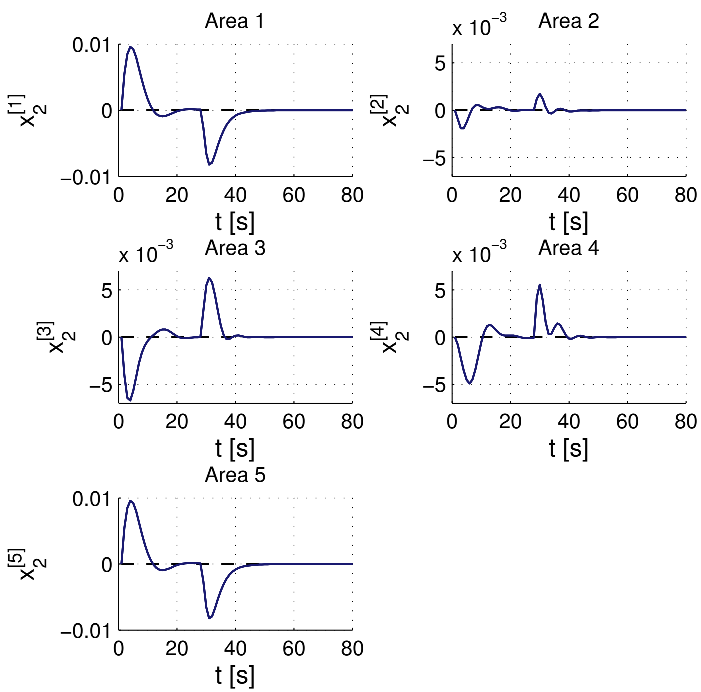
- Four power generation areas with load frequency control
- Model linearized around equilibrium *(Saadat, 2002; Riverso, et al. 2012)*

$$\dot{z}_i = \sum_{j \in \mathcal{N}_i} A_{ij} z_i + B_i v_i + L_i \Delta P_{L_i}$$

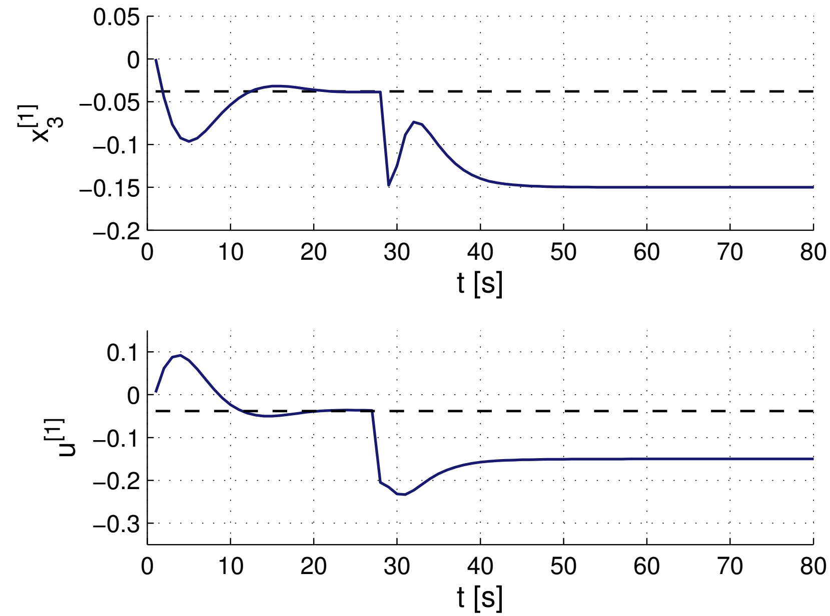


Goals: - Restore frequency, follow load change $\Delta P_{L_1} = -0.15, \Delta P_{L_3} = 0.05$

- Allow fifth area to join the network



Frequency deviation is controlled to zero

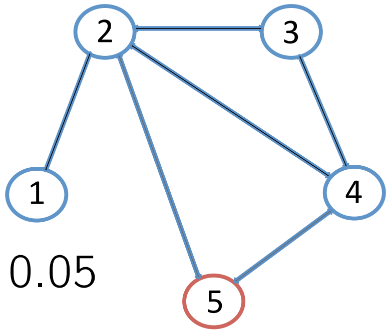


System is first regulated to steady-state and then to the origin

Computational example – Area Generation Control

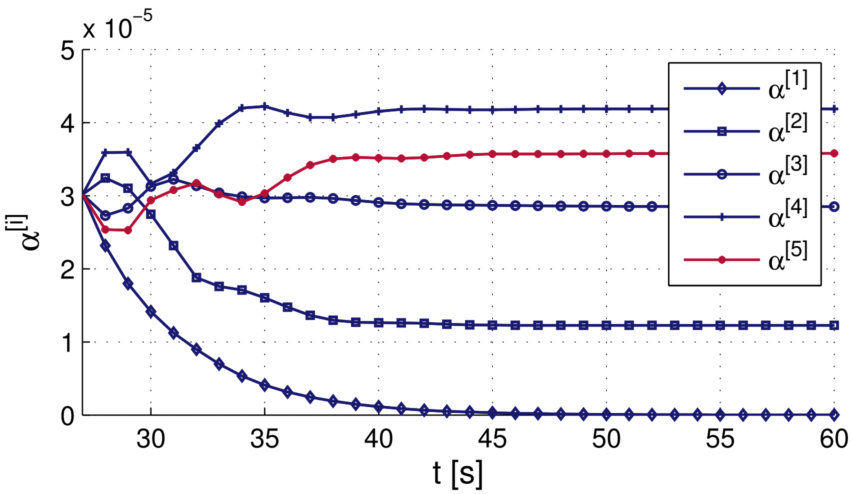
- Four power generation areas with load frequency control
- Model linearized around equilibrium *(Saadat, 2002; Riverso, et al. 2012)*

$$\dot{z}_i = \sum_{j \in \mathcal{N}_i} A_{ij} z_i + B_i v_i + L_i \Delta P_{L_i}$$



Goals: - Restore frequency, follow load change $\Delta P_{L_1} = -0.15, \Delta P_{L_3} = 0.05$

- Allow fifth area to join the network



Terminal set sizes change dynamically

Summary – Distributed MPC

- Structured Lyapunov functions and dynamic invariant sets guarantee stability and invariance by design

- Synthesis and control via distributed optimization

[Conte, et al., ACC 2012], [Conte, et al., CDC 2012]

- Extension to Robust Tube-based MPC and Tracking MPC

[Conte, et al., ECC 2013, Conte, et al., CDC 2013, submitted]

- Plug and Play MPC enables network changes during closed-loop operation

[Zeilinger, et al., CDC 2013, submitted]

Distributed and Real-time MPC

Centralized MPC theory:

☺ Recursive constraint satisfaction

☺ Stability by design

Established approach:

- Optimality
- Terminal cost and constraint

Real-time MPC:

- Flexibility and fast convergence through interior-point methods
- BUT: Variable solve-times

Outline (Part I):

Stability and constraint satisfaction for any real-time constraint

Distributed MPC:

- Reduced conservatism through distributed optimization
- BUT: Global terminal conditions

Outline (Part II):

Stability with larger region of attraction based on local information