Decentralized MPC

MPC of Large-Scale Constrained Linear Systems

Cost function: full matrices $Q,\ R,\ P$

$$\sum_{k=0}^{N-1}\left[x_k'Qx_k+u_k'Ru_k\right]+x_N'Px_N$$

Centralized approach

Process model: coupled dynamics

$$x_{k+1}=Ax_k+Bu_k,\ \underbrace{x\in\mathbb{R}^n},\ \underbrace{u\in\mathbb{R}^m}$$

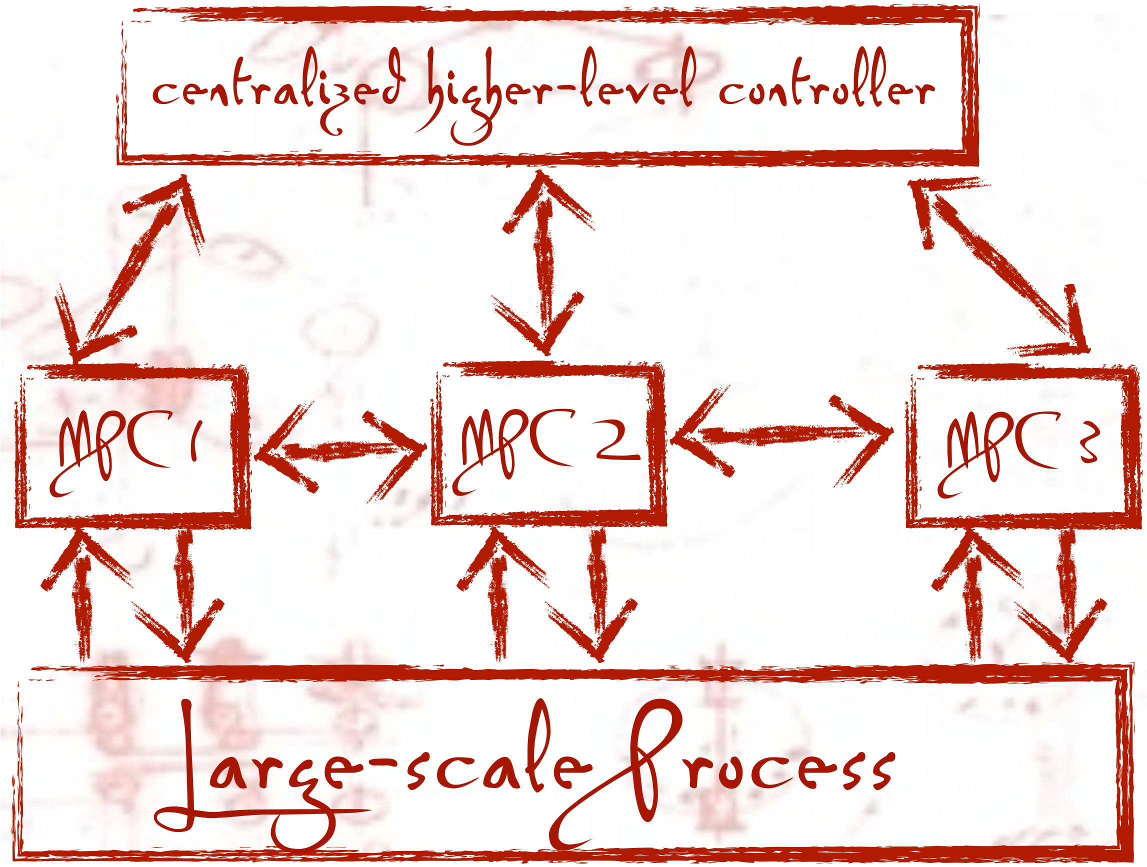
Constraints: $Eu_k+Fx_k\leq G$

Drawbacks

- need **global model**
- **computation** complexity (solve large QP)
- lots of **communicated** data (transmit all states to central unit, transmit back all commands to actuators)
- control design hard to **tune** and **maintain**

Typical DMPC approach

- Measure/estimate **local** states
- Solve local MPC problems
- **Exchange** decisions with neighbors, possibly reiterate local computations
- Apply the current command input to local actuator(s)
- Possibly interact with upper level of decision making (**hierarchical control**)



Main issues: Global closed-loop stability ? Feasibility of global constraints ? Loss of performance w.r.t. centralized control ?

Local DMPC problem

- At time t, each DMPC agent $\#i$ solves the local MPC problem

$$
\begin{aligned}
V_i(x(t)) &= \min_{\{u_k^i\}_{k=0}^{\infty}} \sum_{k=0}^{\infty} (x_k^i)' Q_i x_k^i + (u_k^i)' R_i u_k^i = \\
&= \min_{u_0^i} (x_1^i)' P_i x_1^i + x^i(t)' Q_i x^i(t) + (u_0^i)' R_i u_0^i \\
&\quad \text{s.t. } x_1^i = A_i x^i(t) + B_i u_0^i \\
&\quad\quad x_0^i = W_i' x(t) = x^i(t) \\
&\quad\quad u_{\min}^i \le u_0^i \le u_{\max}^i \\
&\quad\quad u_k^i = 0, \ \forall k \ge 1
\end{aligned}
$$

local weights

$$
\begin{aligned}
Q_i &\triangleq W_i' Q W_i \\
R_i &\triangleq Z_i' R Z_i
\end{aligned}
$$

local states

local optimizer: $u_0^{i*} = \begin{bmatrix} u_0^{i1*} \\ \vdots \\ u_0^{ii*} \\ \vdots \\ u_0^{im_i*} \end{bmatrix}$

this is the only input applied to the system: $u_i(t) = u^{ii*}(t)$

these are not applied, but provide an estimate of neighbors' optimal moves (depending on amount of model mismatch)

The actual input vector $u(t)$ commanded to process is the collection of all optimal inputs $u^{11*}(t)$, $u^{22*}(t)$, … , $u^{mm*}(t)$

Stability results

(Alessio, Bemporad, 2007)

input mismatch

state mismatch

Theorem Assume A and all matrices A_i open-loop as. stable. Let $P_i = A_i' P A_i + Q_i$, $\forall i = 1, \ldots, M$. Define

$$\Delta u^i(t) \triangleq u(t) - Z_i u_0^{*i}(t), \qquad \Delta x^i(t) \triangleq (I - W_i W_i')x(t)$$

$$\Delta A^i \triangleq (I - W_i W_i')A, \qquad \Delta B^i \triangleq B - W_i W_i' B Z_i Z_i'$$

model mismatch

Also, let

$$\Delta Y^i(x(t)) \triangleq W_i W_i'(A \Delta x^i(t) + B Z_i Z_i' \Delta u^i(t)) + \Delta A^i x(t) + \Delta B^i u(t)$$

and

$$\Delta S^i(x(t)) \triangleq \left(2(A_i W_i' x(t) + B_i u_0^{*i}(t))' + \Delta Y^i(x(t))' W_i\right) P_i W_i' \Delta Y^i(x(t))$$

If the condition

$$x' \left(\sum_{i=1}^{M} W_i W_i' Q W_i W_i' \right) x - \sum_{i=1}^{M} \Delta S^i(x) \geq 0, \ \forall x \in \mathbb{R}^n$$

is satisfied then the decentralized MPC scheme is globally asymptotically stabilizing.

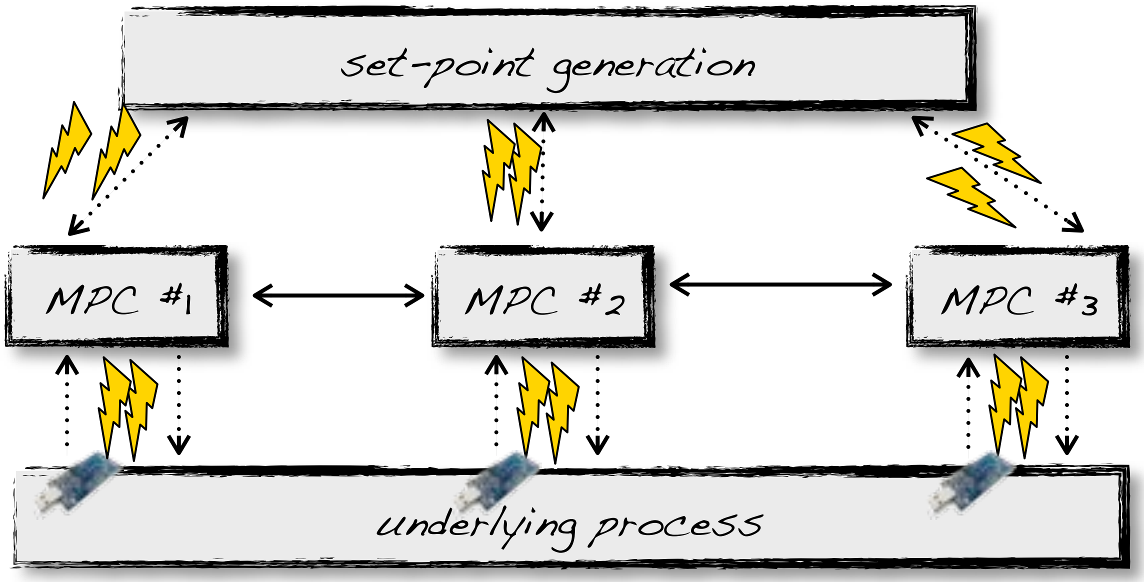
overall mismatch

$\Delta S_i(t)$ depends on amount of mismatch between global and local models.

$\Delta S_i(t) = 0$ if no mismatch, i.e. all local models are also global (=full feedback)

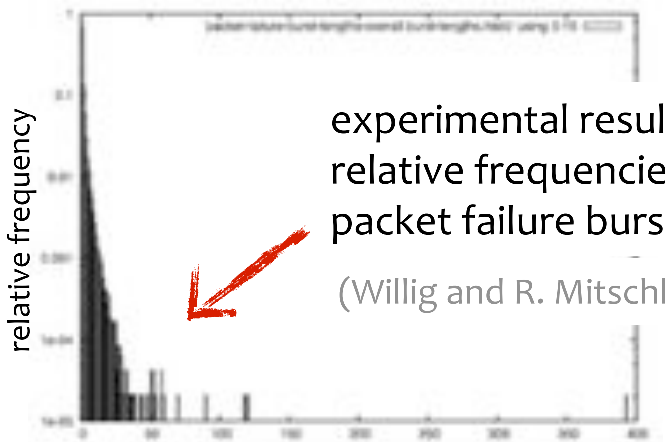
Extension to intermittent measurements

- Assume all data are exchanged on a wireless network
- The network may be congested and packets drop out.
- Assumption: when packets are lost at time t, by default $u_i(t)=0$
- Assumption: at most N packets can be lost consecutively
- Model mismatch grows with the number of consecutive packet losses



(Alessio, Bemporad, 2008)
(Barcelli, Bemporad, 2009)

Packet-loss probability model

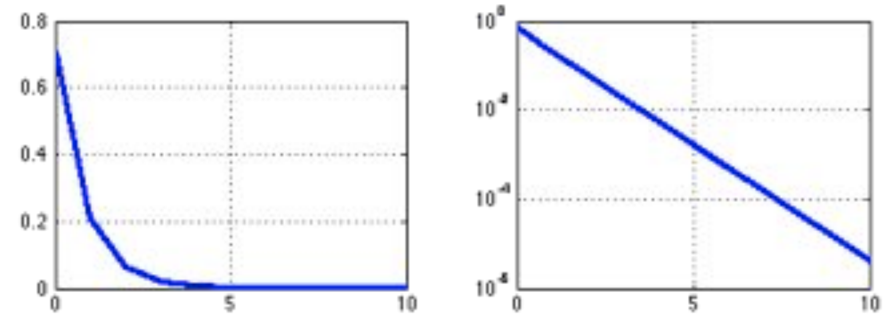


burst length N

experimental results on relative frequencies of packet failure burst length

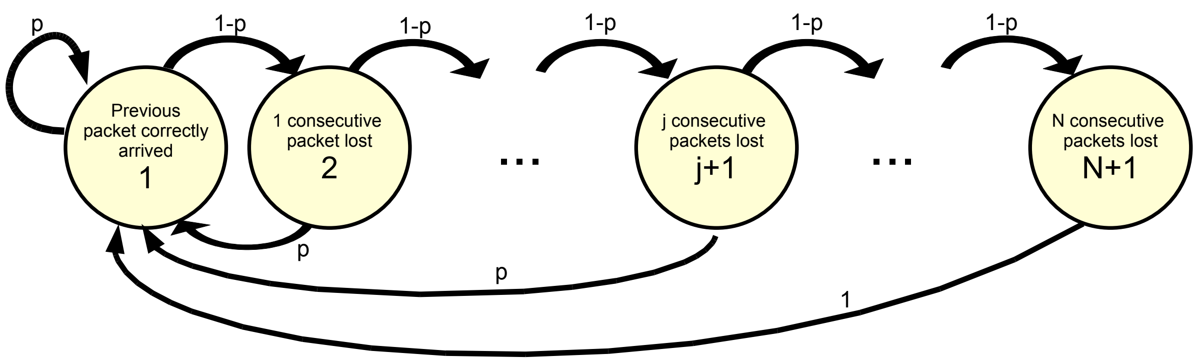
(Willig and R. Mitschke, 2006)

probability π_i of occurrence



burst length N burst length N

Proposed probability model: Markov chain (Barcelli, Bemporad, 2009)



$$P = \begin{bmatrix} p & 1-p & 0 & \cdots & 0 \\ p & 0 & 1-p & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p & 0 & 0 & \cdots & 1-p \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

one-step probability matrix

$$\begin{cases} \pi' = \pi' P \\ \sum_{i=1}^{N} \pi_i = 1 \end{cases}$$

$\pi_i =$ steady-state probability of losing consecutively exactly i-1 packets

Stability results

(Alessio, Barcelli, Bemporad, 2009)

Theorem Let

$$\Delta S_j^i(x) \triangleq [2(A_i W_i' x + B_i u_0^{*i}(x))' W_i' + \Delta Y^i(x)'](A^{j-1})' W_i P_i W_i' A^{j-1} \Delta Y^i(x)$$

and let

$$\xi_i(x) \triangleq A_i W_i' x + B_i u_0^{*i}(x)$$

If the condition

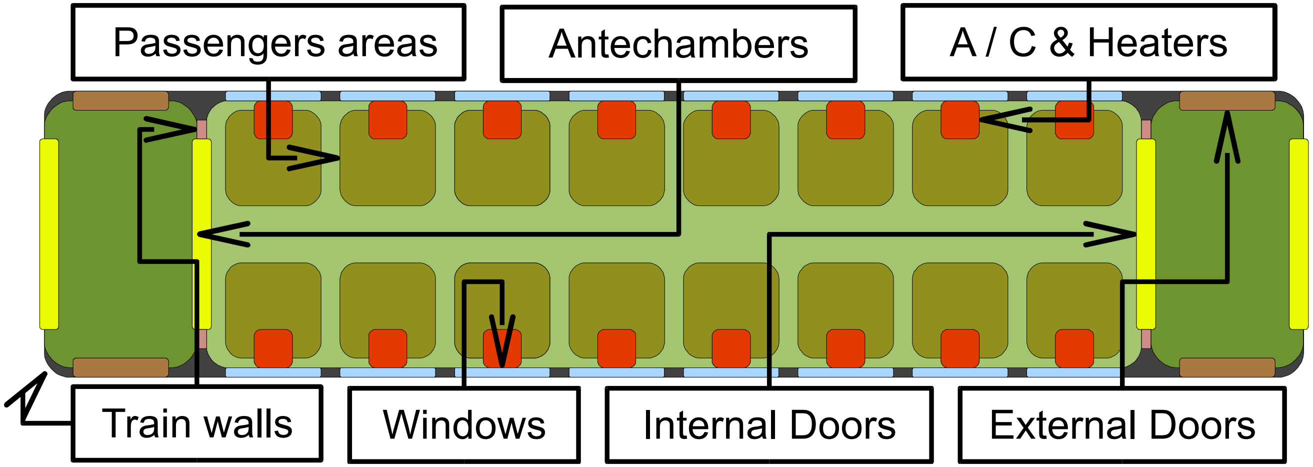
$$\sum_{i=1}^{M} \left(x' W_i W_i' Q W_i W_i' x + \xi_i(x)'(P_i - W_i'(A^{j-1})' W_i P_i W_i' A^{j-1} W_i)\xi_i(x) - \Delta S_j^i(x)\right) \geq 0$$

is satisfied $\forall x \in \mathbb{R}^n$, $\forall j = 1, \ldots, N$ then the decentralized MPC scheme is globally asymptotically stable under packet loss.

- Proof based again on showing that $V(x(t)) = \sum_{i=1}^{M} V_i(x(t))$ is a Lyapunov function
- Note: Proof does **not** depend on ***probability model*** for packet loss !
- **Local asymptotic stability:** check eigenvalues of nxn matrix
- **Global asymptotic stability:** test LMI relaxation of resulting PWA closed-loop

Decentralized temperature control example

Temperature control in different passenger areas in a railcar



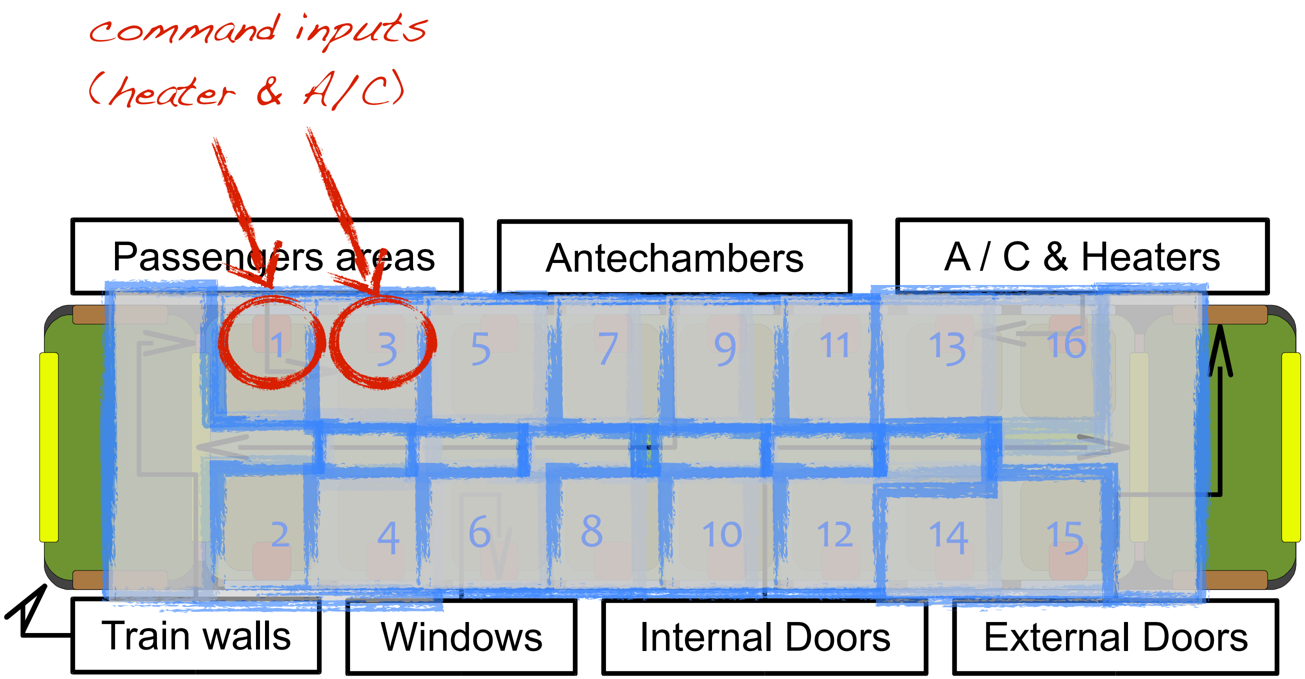
Heat transmission equation

$$\frac{dT_j(\tau)}{d\tau} = \sum_{i=0}^{n} Q_{ij}(\tau) + Q_{uj}$$

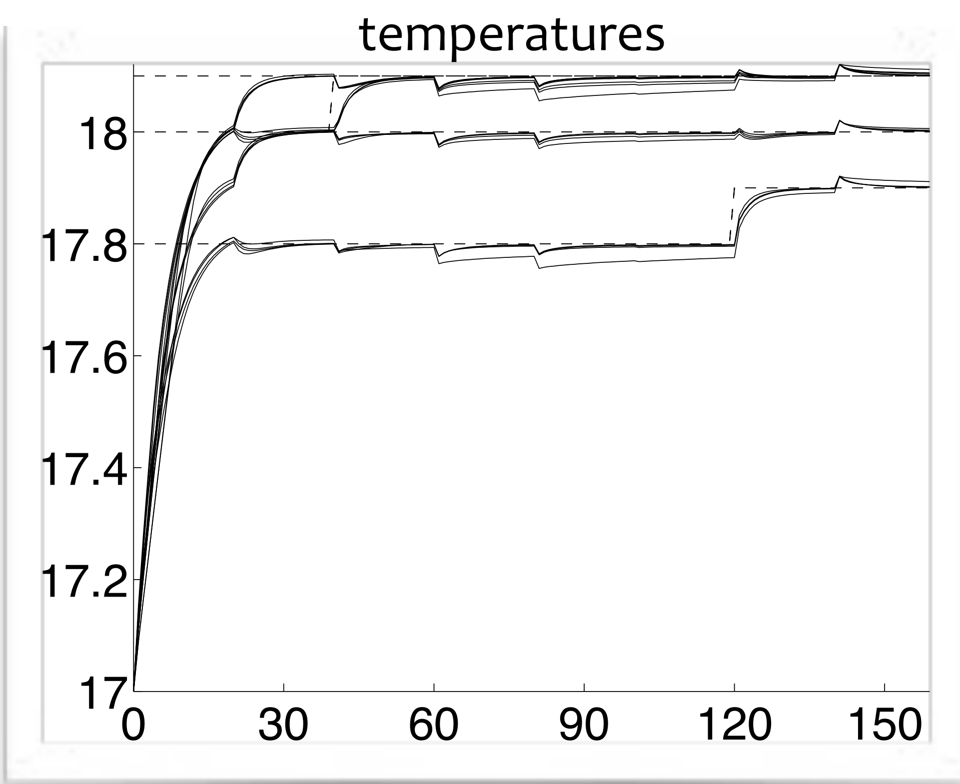
$$Q_{ij}(\tau) = \frac{S_{ij}K_{ij}(T_i(\tau) - T_j(\tau))}{C_j L_{ij}}, \quad j = 1, \ldots, n$$

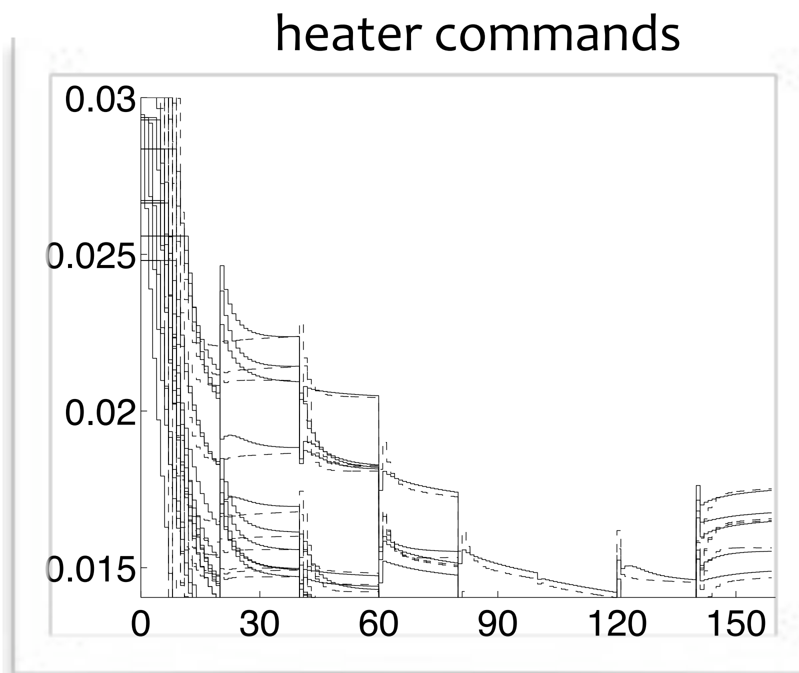
Model decomposition

- Global model: 26 states, 16 inputs
- Define 16 submodels
- Each model has 5 states, 2 or 3 command inputs



DMPC - Simulation results (no packet loss)



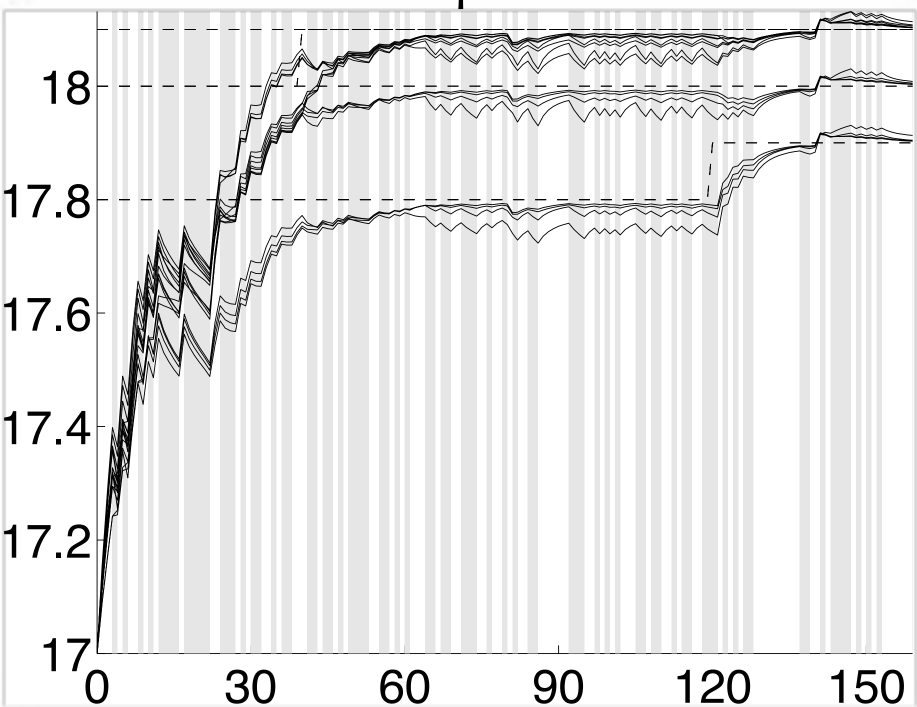


$$Q = 2\begin{bmatrix} 10^2 I_{16} & 0 \\ 0 & 10^{-1} I_{10} \end{bmatrix},\ R = 10^5 I_{16},$$

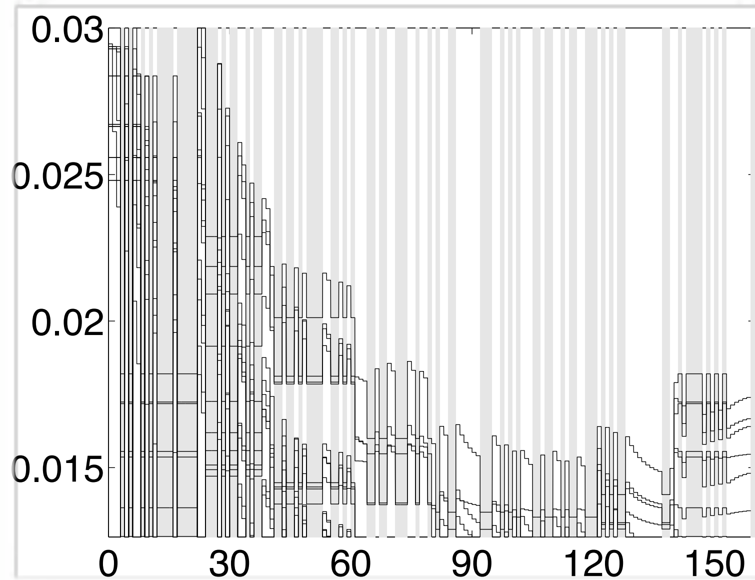
$$u_{min} = -0.03\ \mathrm{W},\ u_{max} = 0.03\ \mathrm{W},\ T_s = 9\ \mathrm{min}$$

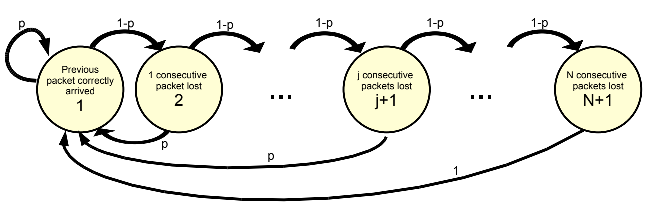
DMPC - Simulation results (with packet loss)

temperatures



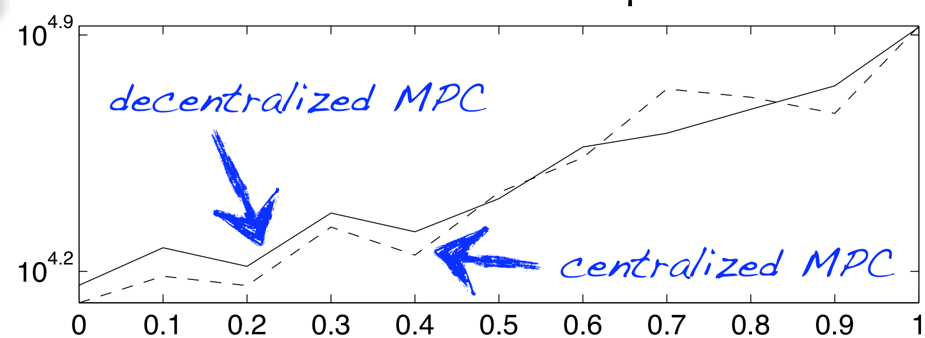
heater commands





Packet loss generated by Markov chain

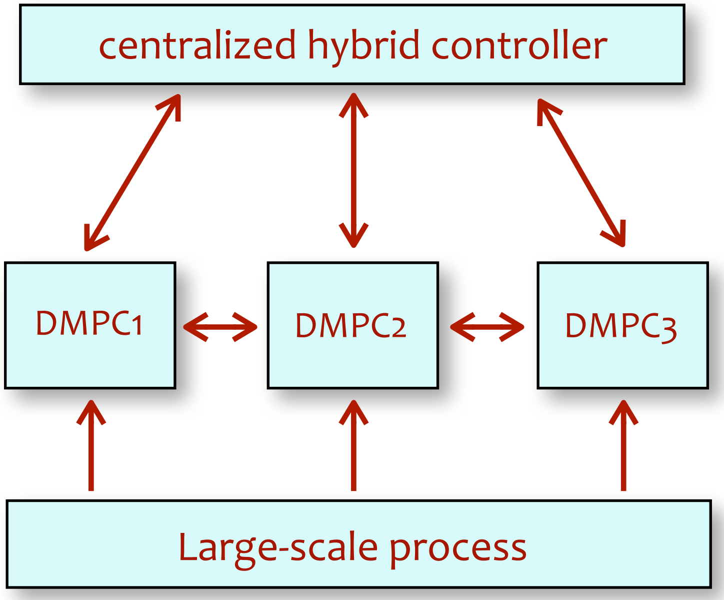
centralized vs. decentralized performance



probability of packet dropout

Hierarchical MPC design

Hierarchical Hybrid MPC



- Centralized hybrid MPC for **global coordination**:
  - Enforce *global constraints* (mixed *linear & logical*)
  - Optimize *global objectives*

- Need an **abstract (hybrid) model** of underlying closed-loop dynamics (--> full dynamic model and full state information is not needed)

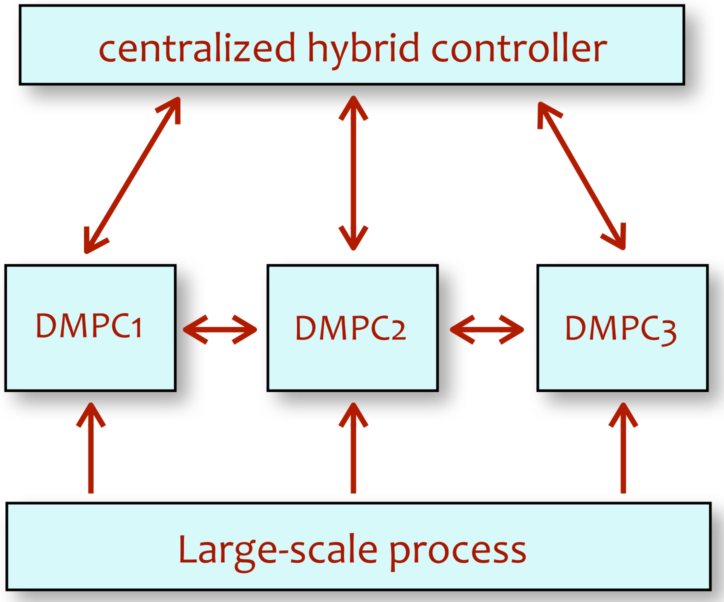
- Run in real-time at a **lower sampling frequency**

Hierarchical Hybrid MPC

Example:

$$y(k+1) = G(1)u(k)$$

- $G(1)$= global DC gain. Underlying closed-loop dynamics abstracted as just one-step delay



$$\begin{array}{ll} \min\limits_{u(k)} & f(y(k+1) - r_d(k), u(k)) \\ \text{s.t.} & g(u(k), y(k), r_d(k)) \leq 0 \end{array}$$

any convex quadratic or PWA function

any set of (mixed-integer) linear constraints

desired reference

output feedback from underlying process

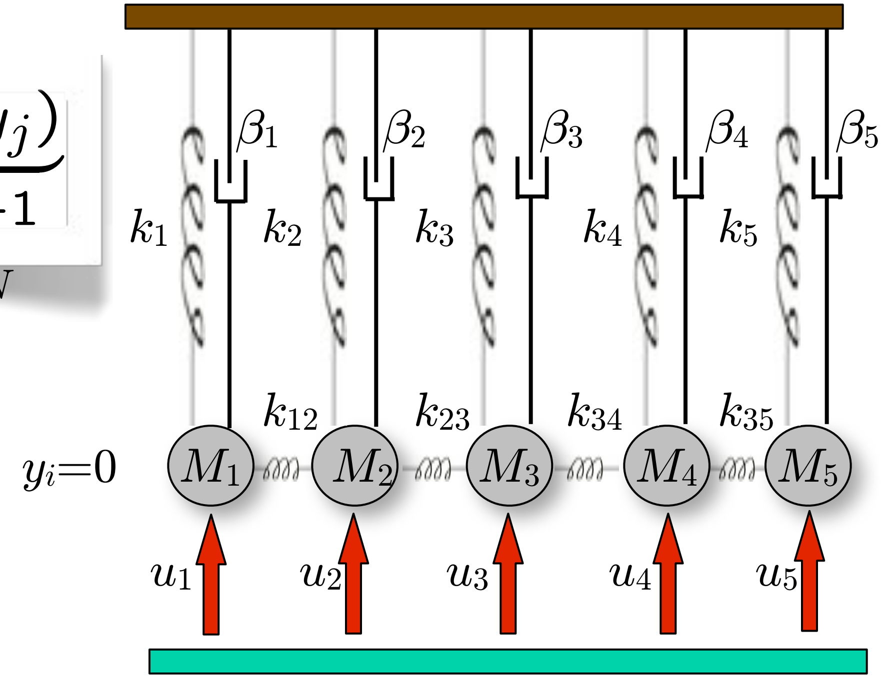
- Define set-points for local DMPC's:

$$r(k) = G(1)u(k)$$

Hierarchical Hybrid MPC - Example

- Multi-mass system, $N=5$ masses, only vertical motion

$$M_i\ddot{y}_i = u_i - \beta_i\dot{y}_i - k_i y_i - \underbrace{k_{ij}(y_i - y_j)}_{\substack{j=i-1,i+1 \\ j\geq 1, j\leq N}}$$



- Local model #i: only considers $y_i, \dot{y}_i, y_{i-1}, y_{i+1}$ as states, assuming $\dot{y}_{i-1} = \dot{y}_{i+1} = 0$

- Local MPC #i:

$$\min_{u_i(k),\dots,u_i(k+N_u-1)} \sum_{j=0}^{Ny-1} (y_i(k+j) - r_i(k))^2$$

$$\text{s.t.} \quad u^i_{\text{min}}(k) \leq u_i(k+j) \leq u^i_{\text{max}}(k)$$

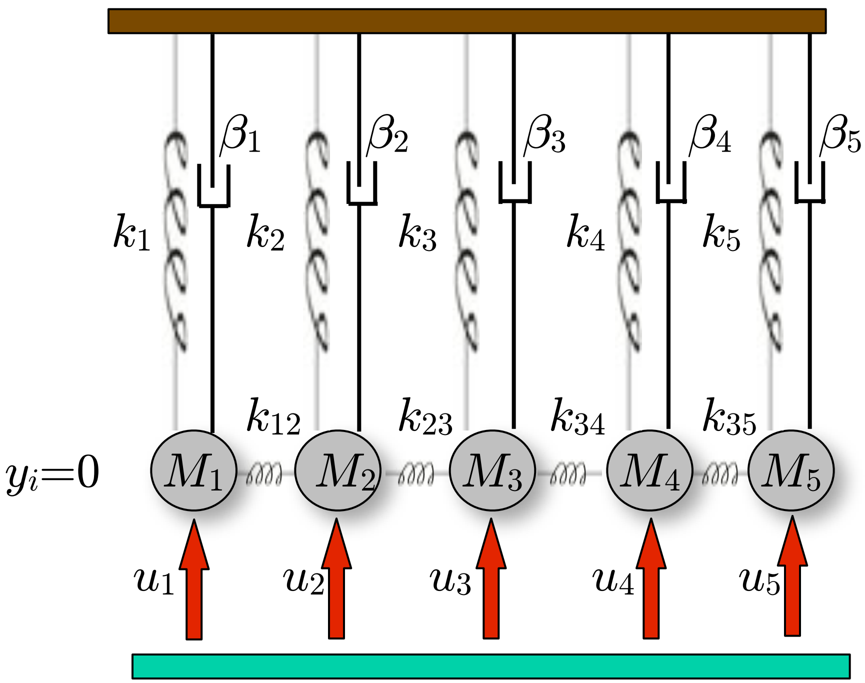
Hierarchical Hybrid MPC - Example

- Hierarchical hybrid MPC based on static global model

$$y(k+1) = G(1)u(k)$$

- Hierarchical hybrid MPC decides input bounds $u^i{}_{max}(k)$ in real-time

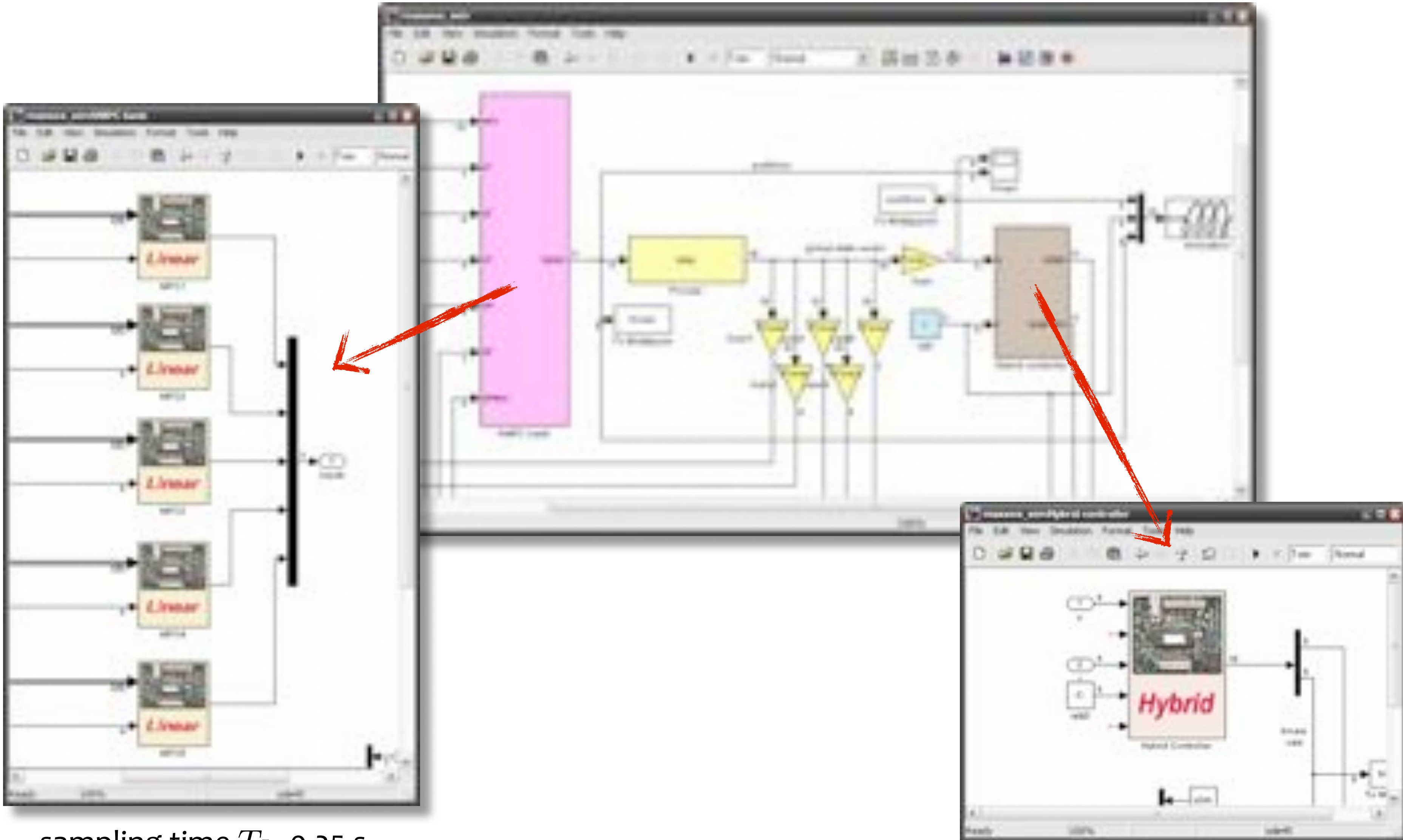
- Hierarchical hybrid MPC decides local set points $r(k)$ in real-time



- Constraints:
  - **at most** K_u inputs can be over a certain threshold $u_i(k) \geq u_{lim}$
  - set-point changes are bounded $|G(1)u(k) - y_i(k)| \leq \Delta_r$

- Cost function: $$\min_{u(k)} \|G(1)u(k) - r_d(k)\|^2$$ and set $r(k) \triangleq G(1)u(k)$

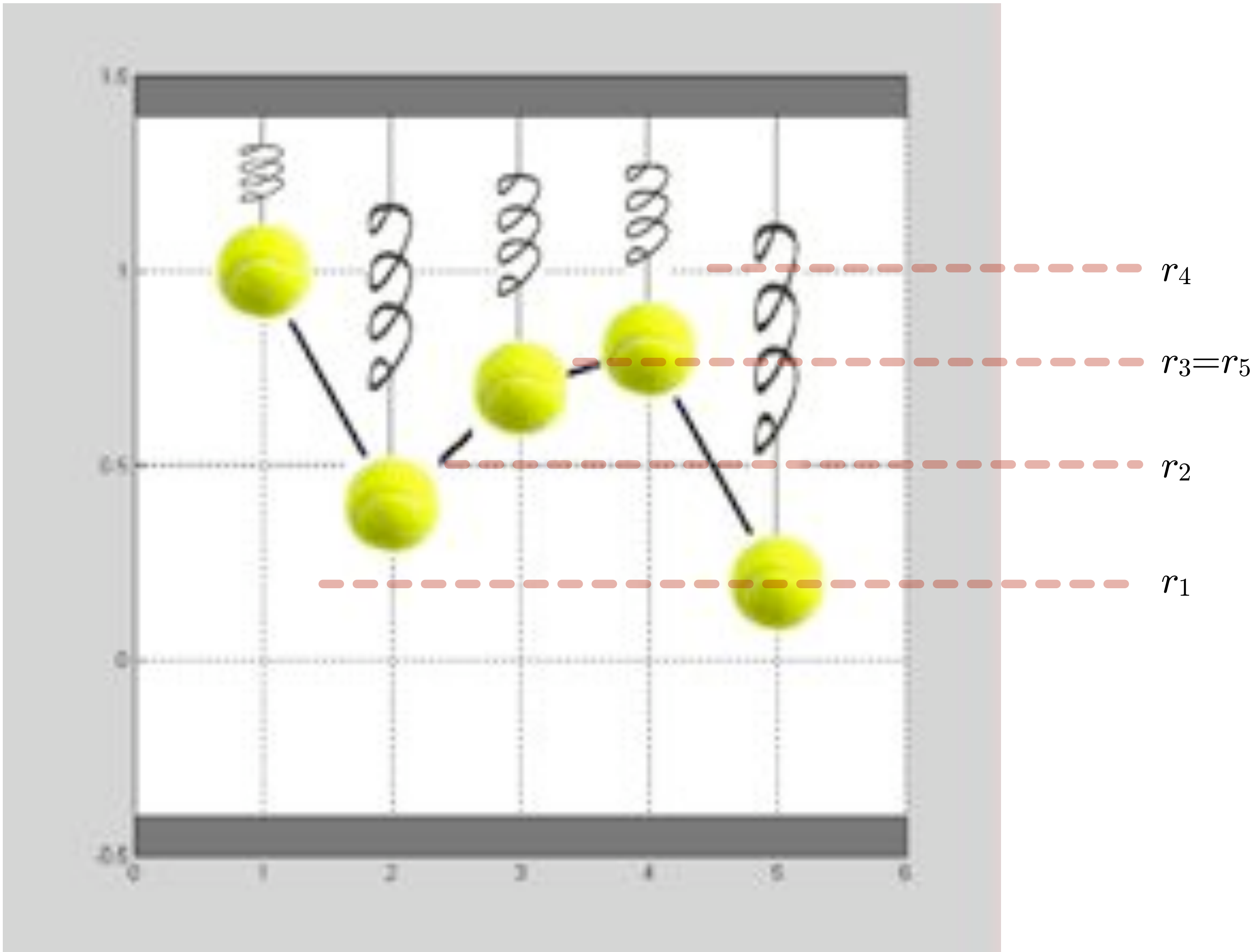
Hierarchical Hybrid MPC - Example



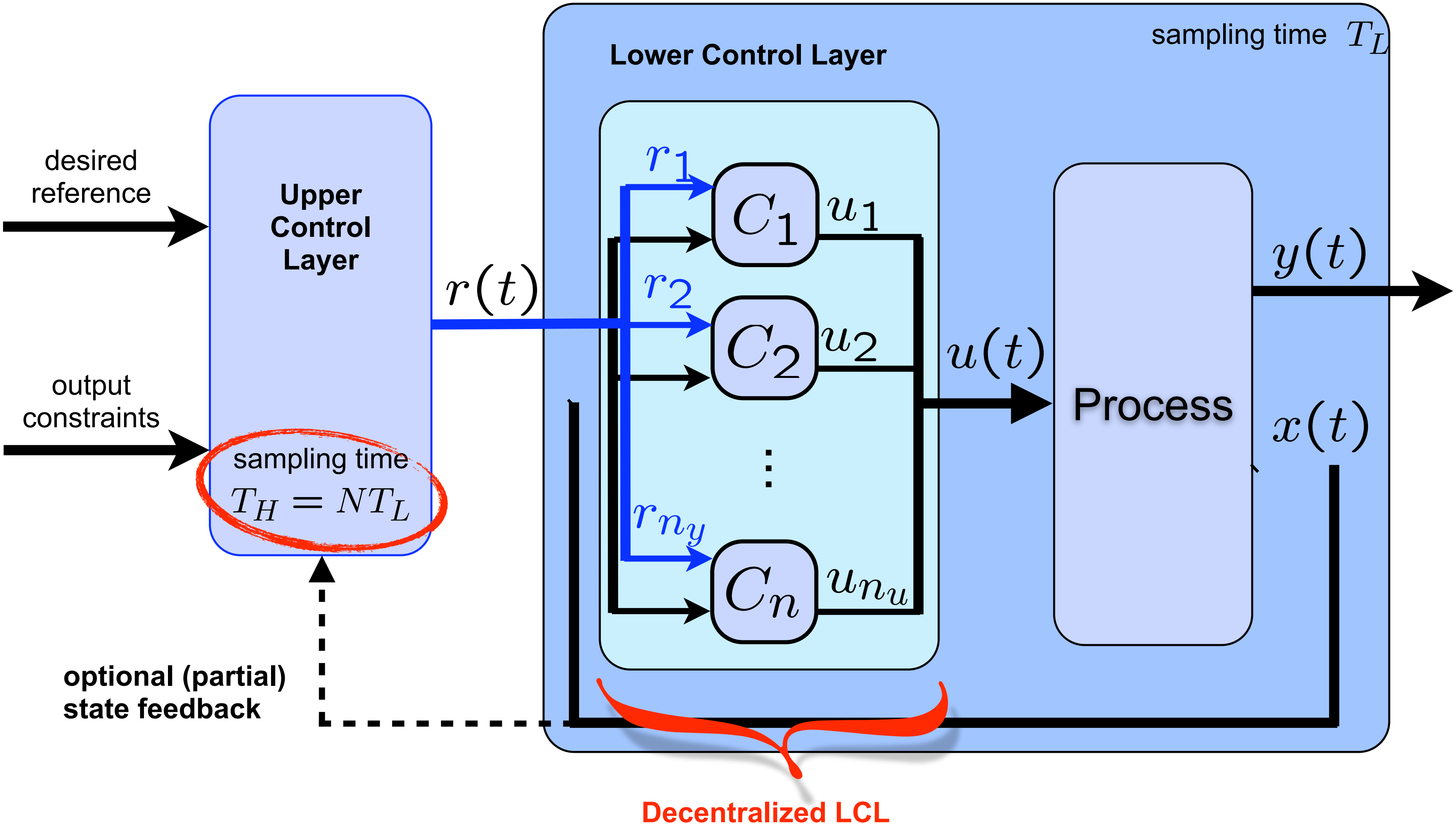
sampling time T_L=0.25 s

sampling time T_H=3 s

Hierarchical Hybrid MPC - Example



Hierarchical structure: optimal choice of sample time



Hierarchical structure: optimal choice of sample time

Plant equations

$$\begin{cases} x(t+1) &= \bar{A}x(t) + \bar{B}u(t) \\ y(t) &= \bar{C}x(t) + \bar{D}u(t) \\ u(t) &= Kx(t) + Er(t) \end{cases}$$

DC Gain

$$E = ((\bar{C} + \bar{D}K)(I - \bar{A} - \bar{B}K)^{-1}\bar{B} + \bar{D})^{-1}$$

LCL specs

$$e(t) \triangleq y(t) - r(t)$$

$$\mathcal{E} = \{e \in \mathbb{R}^{n_y} : \ H_y e \leq \Delta K_y\}$$

Tuning knob

LCL closed-loop

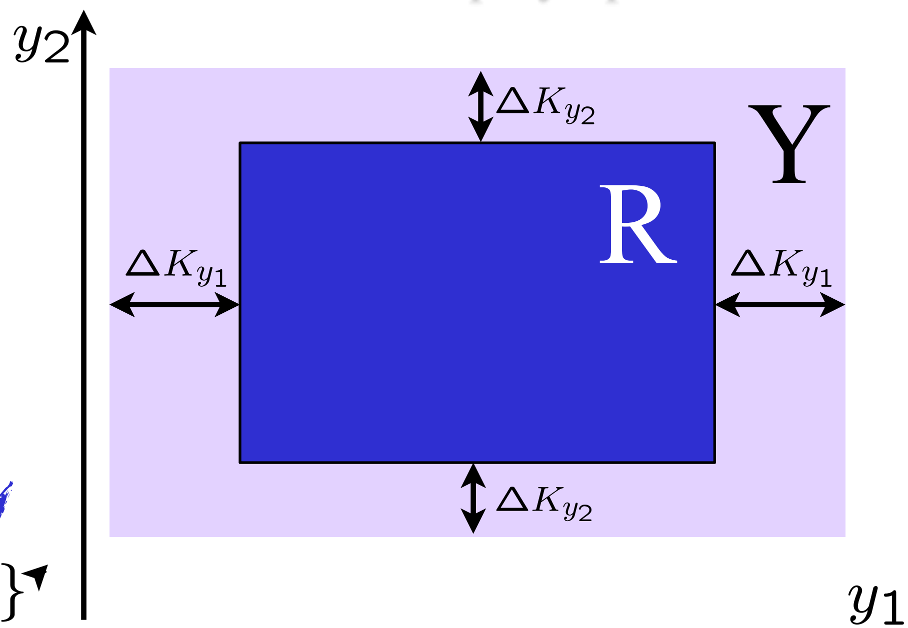
$$\begin{cases} x(t+1) &= Ax(t) + Br(t) \\ y(t) &= Cx(t) + Dr(t) \end{cases}$$

Constraints

$$\mathcal{Y} = \{y \in \mathbb{R}^{n_y} : \ H_y y \leq K_y\}$$

$$\mathcal{R} = \{r \in \mathbb{R}^{n_y} : \ H_y r \leq K_y - \Delta K_y\}$$

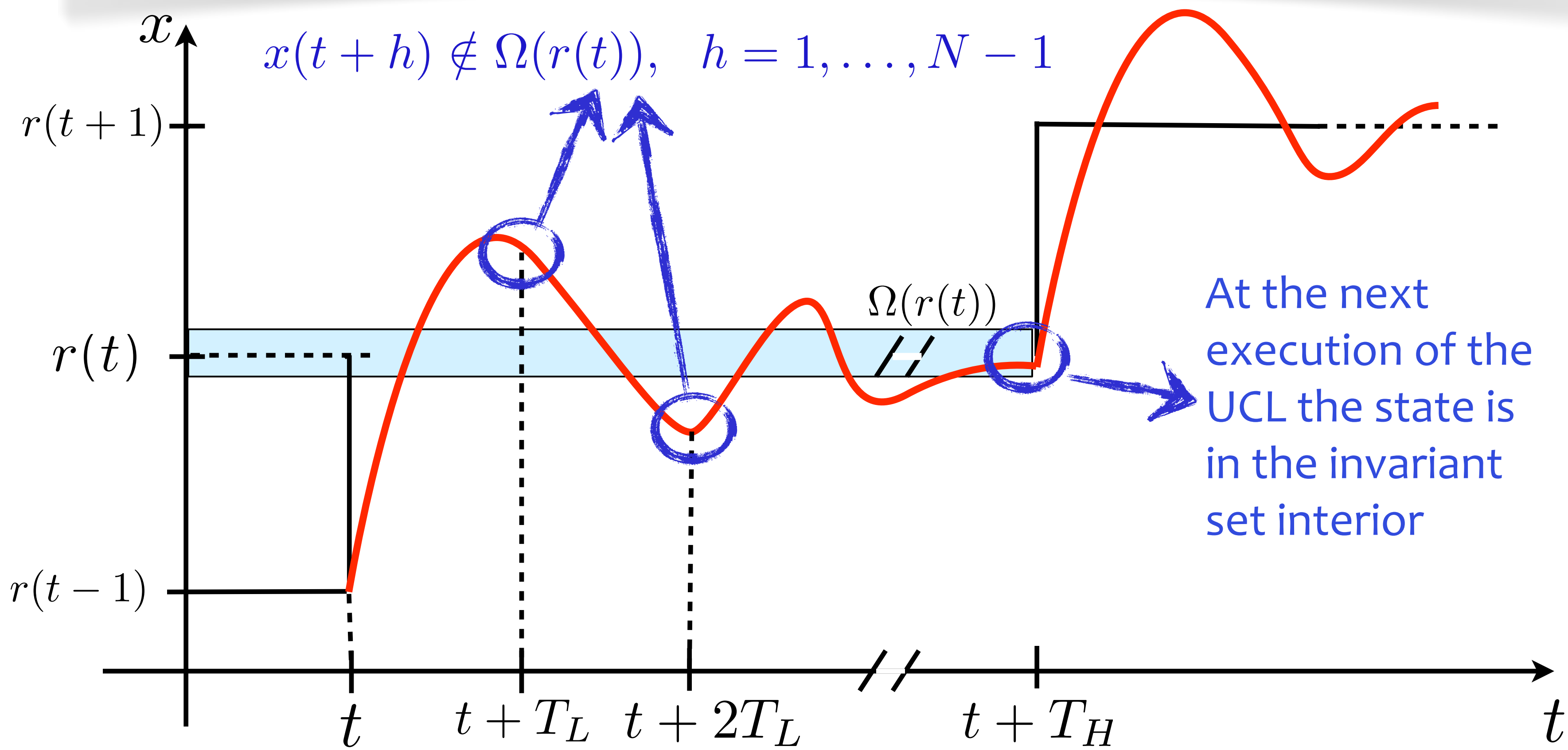
Admissible polytopes



inner closed loop stable

Hierarchical structure: optimal choice of sample time

Main Idea: *at time* t *allow* $\Delta r = ||r(t) - r(t - NT_L)||_\infty$ *"small enough" such that* $x(t + NT_L) \in \Omega(r(t))$

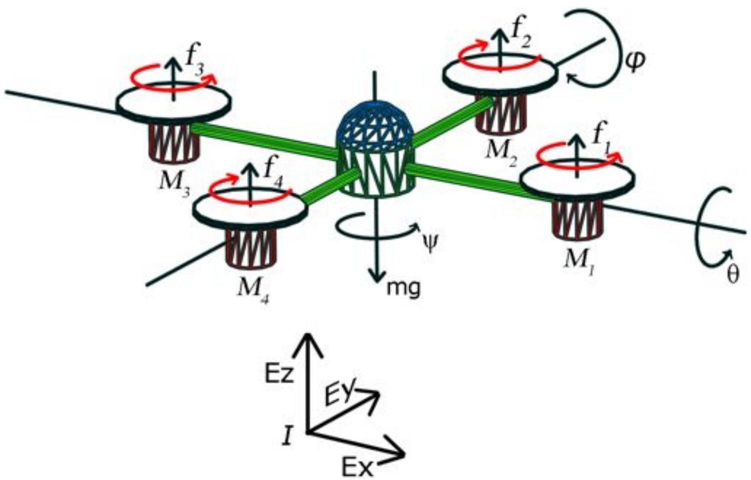


Hierarchical MPC for stabilization of mini-UAVs

(Bemporad, Pascucci, Rocchi, 2009)

Quadcopter model (nonlinear, 6-DOF)

- 4 command inputs:
  $V_{M1}, V_{M2}, V_{M3}, V_{M4}$
- 12 outputs:
  $\theta, \phi, \psi, x, y, z, \dot{\theta}, \dot{\phi}, \dot{\psi}, \dot{x}, \dot{y}, \dot{z}$



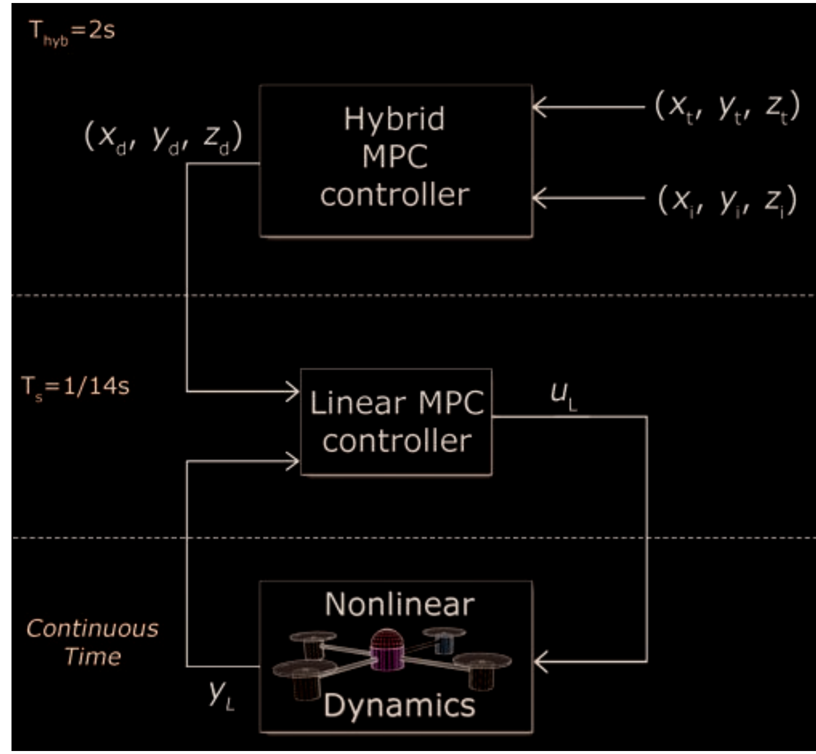
$$
\begin{aligned}
\ddot{x} &= (-u\sin\theta - \beta\dot{x})\frac{1}{m}\\
\ddot{y} &= (u\cos\theta\sin\phi - \beta\dot{y})\frac{1}{m}\\
\ddot{z} &= -g + (u\cos\theta\cos\phi - \beta\dot{z})\frac{1}{m}\\
\ddot{\theta} &= \frac{\tau_\theta}{I_{xx}}\\
\ddot{\phi} &= \frac{\tau_\phi}{I_{yy}}\\
\ddot{\psi} &= \frac{l}{I_{zz}}(-f_1 + f_2 - f_3 + f_4)
\end{aligned}
$$

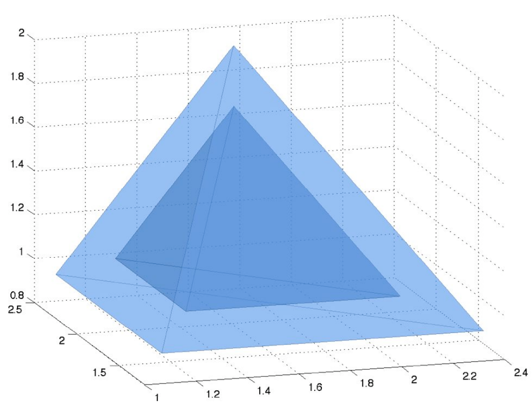
Constraints

- Inputs (motors voltages) saturation
- Altitude $z \geq 0$
- Pitch and roll angles $-\frac{\pi}{6} \leq \theta, \phi \leq \frac{\pi}{6}$ (soft costraints)

Hybrid MPC for obstacle avoidance

- Generate desired position in real-time to avoid obstacles
- Obstacles modeled as polyhedra (tetrahedra)
- HYSDEL for description of hybrid dynamics (quadcopter + obstacles)
- Hybrid Toolbox for conversion to MLD form and hybrid MPC design

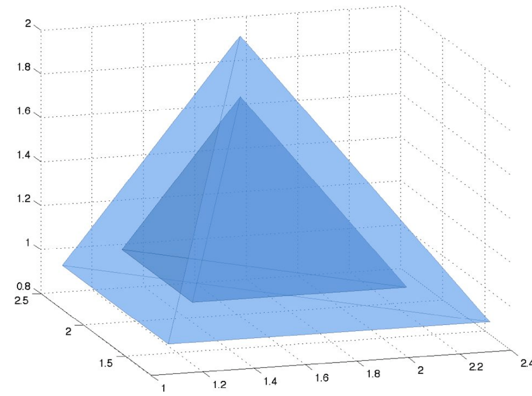




$$\begin{cases} x(k+1) &= \alpha_{1x}x(k) + \beta_{1x}(x_d(k) + \Delta x_d(k)) \\ y(k+1) &= \alpha_{1y}y(k) + \beta_{1y}(y_d(k) + \Delta y_d(k)) \\ z(k+1) &= \alpha_{1z}z(k) + \beta_{1z}(z_d(k) + \Delta z_d(k)) \end{cases}$$

Hybrid modeling of obstacles

- Obstacles modeled as tetrahedra



$$A_{\mathrm{obs}} k_i \begin{bmatrix} x(k)-x_i(k) \\ y(k)-y_i(k) \\ z(k)-z_i(k) \end{bmatrix} \leq B_{\mathrm{obs}}$$

- Staying ``out of the obstacle'' is a nonconvex constraint. Use binary vars

$$[\delta_{ij}(k)=1] \leftrightarrow [A^j_{\mathrm{obs}} k_i \begin{bmatrix} x(k) \\ y(k) \\ z(k) \end{bmatrix} \leq C^j_{\mathrm{obs}}(k)]$$

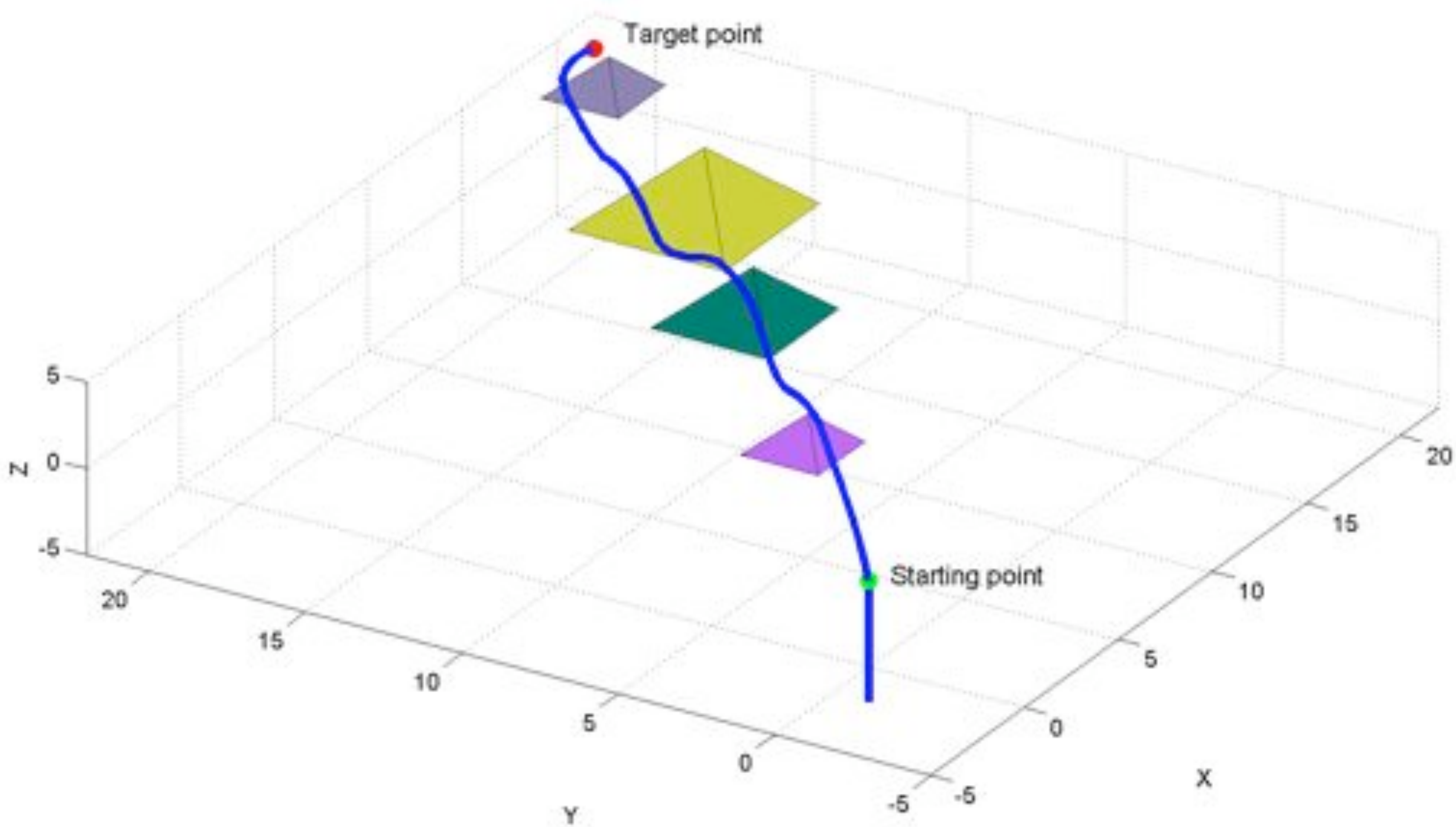
subject to the logical constraint

$$\bigvee_{j=1}^{4} \neg\delta_{ij}(k)=1,\ \forall i=1,\ldots,M$$

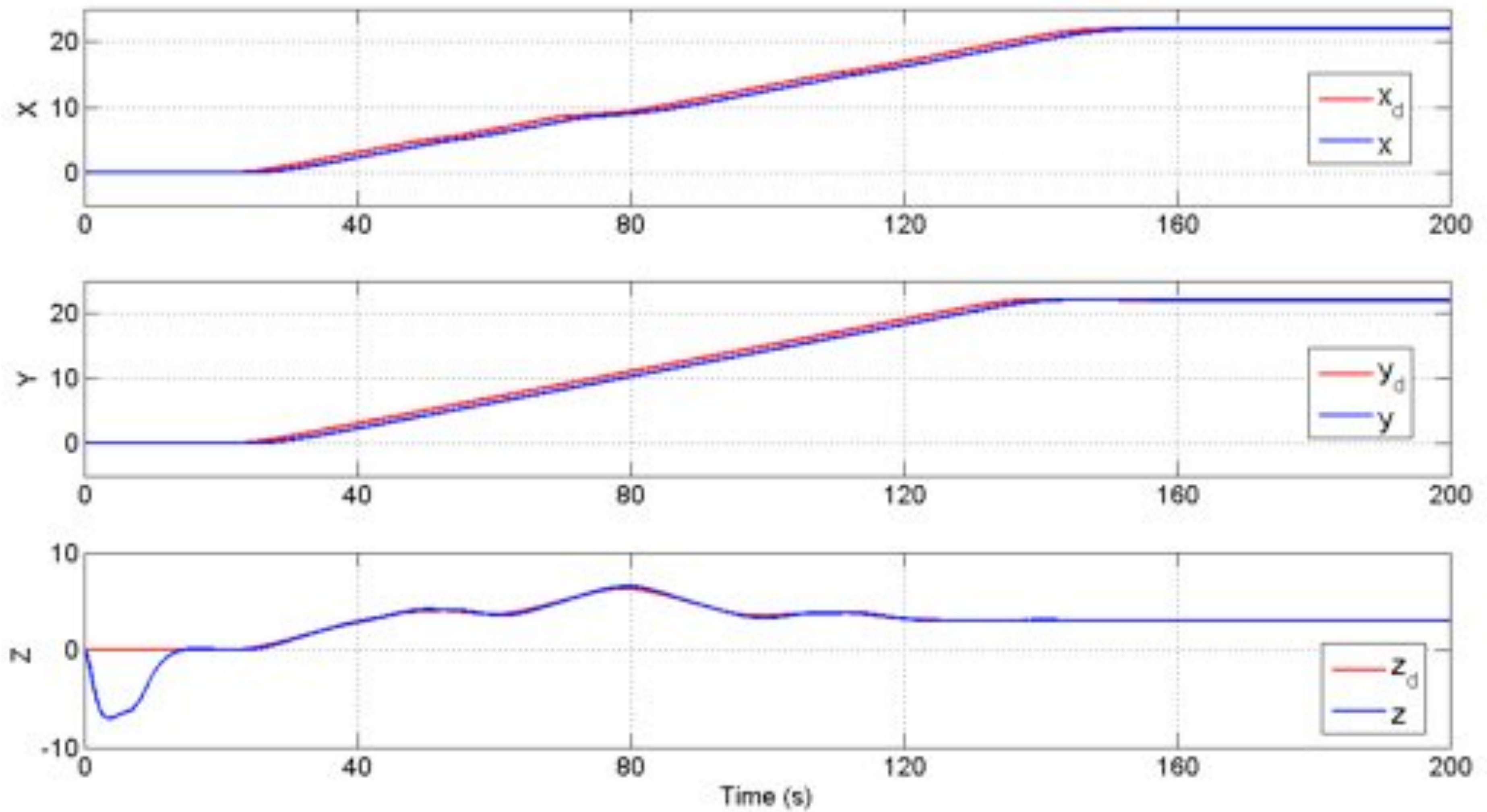
- Use double layer to model an "undesired" area around the obstacle the UAV should avoid

Navigation results (hybrid + linear MPC)

- Obstacles positions are known at each sample step
- Trajectory generated in real-time



Tracking the references generated by hybrid MPC



Average CPU time for hybrid MPC = 135 ms per time step (T_{hyb}=1.5 s), using the MIQP solver of CPLEX 11.2

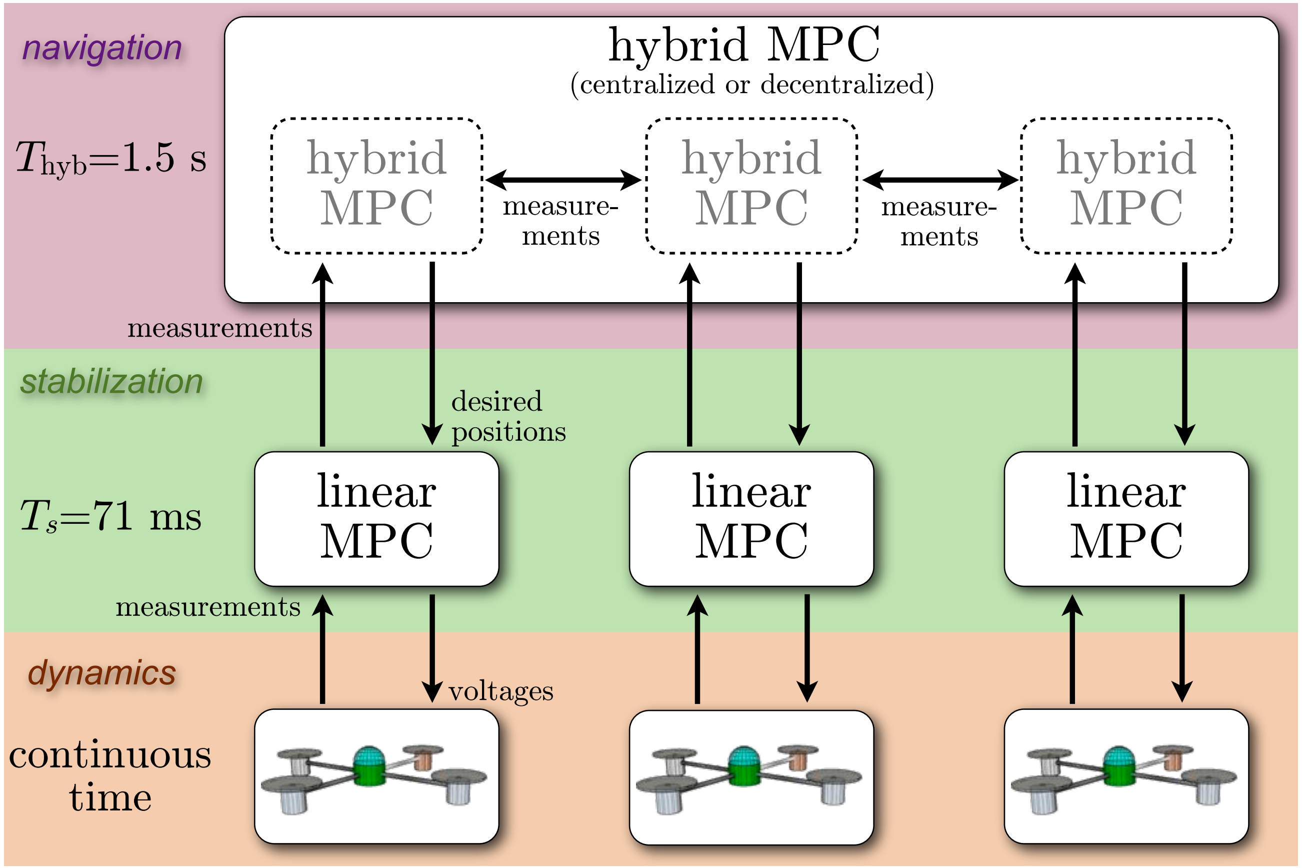
Now going into experiments …

(C.A. Pascucci, master thesis, 2010)

Hybrid MPC for formation flight

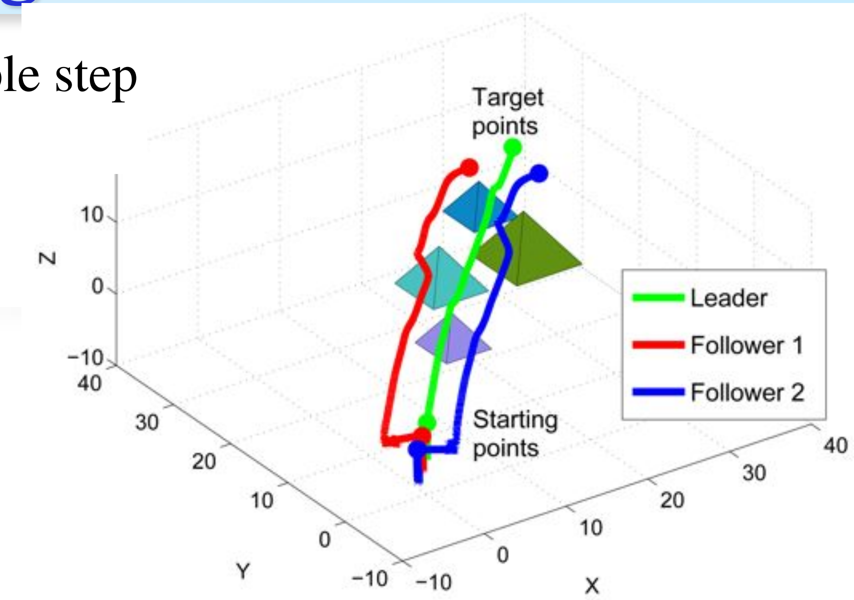
(Bemporad, Rocchi, 2010)

- Leader-follower approach
- Two control scheme:
  - Decentralized hierarchical hybrid + linear MPC
  - Centralized hybrid MPC + decentralized linear MPC

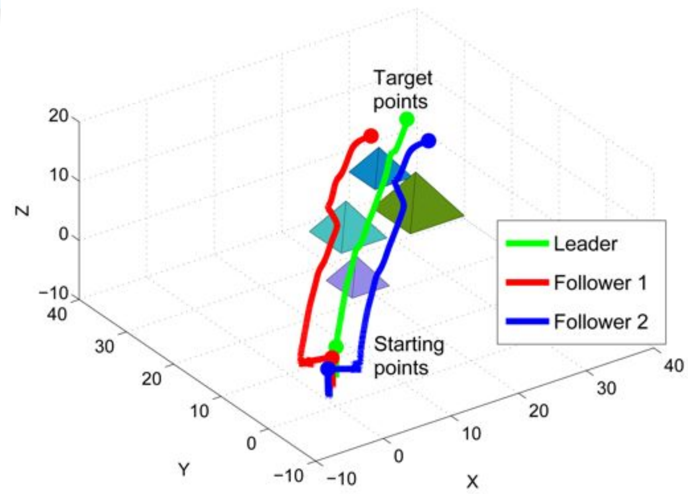


Hybrid MPC for formation flight

- Obstacles positions are known at each sample step
- Trajectory generated in real-time



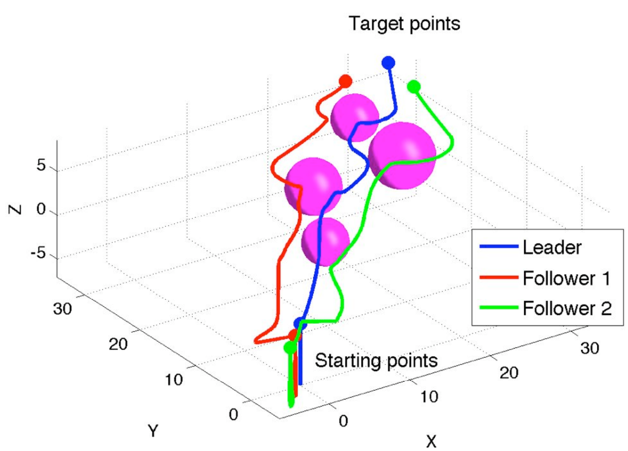
Hybrid MPC for formation flight



$$J_{\text{tt}} = \sum_{k=250}^{4200} \|p_L(k) - p_t\|_2^2$$

$$J_{\text{fpt}} = \sum_{k=250}^{4200} \|p_L(k) - p_{F1}(k) - p_{d1}\|_2^2 + \|p_L(k) - p_{F2}(k) - p_{d2}\|_2^2$$

$$J_{\text{u}} = \sum_{k=250}^{4200} \|u(k) - u(k-1)\|_1$$



	J_{tt}	J_{fpt}	J_{u}
centralized hybrid MPC	1	1	1
decentralized hybrid MPC	-0.09%	-0.51%	-0.03%
potential fields	+210.32%	+294,30%	+212.75%

Average CPU time for hybrid MPC: decentralized 73 ms, centralized 466 ms.