Linear Time-Varying Model Predictive Control

Linear time-varying MPC

- MPC can easily handle **linear time-varying (LTV)** problems

$$\begin{cases} x_{k+1} = A_k(t,x(t))x_k + B_k(t,x(t))u_k + f_k(t,x(t)) \\ y_k = C_k(t,x(t))x_k + D_k(t,x(t))u_k + g_k(t,x(t)) \end{cases}$$

$$E_k(t,x(t))x_k + F_k(t,x(t))u_k \leq h_k(t,x(t)) \qquad k = 0,1,\ldots,N-1$$

$$\min \sum_{k=0}^{N} \ell_k(y_k, u_k, r(t+k), t, x(t)) \qquad \ell_k = \text{quadratic function of } y_k, u_k$$

$$\begin{array}{ll} \min_U & \frac{1}{2}U'H(t)U + F(t)'U + \cancel{\alpha(t)} \\ \text{s.t.} & G(t)U \leq W(t) \end{array}$$

LTV-MPC still leads to a **Quadratic Program (QP)** !

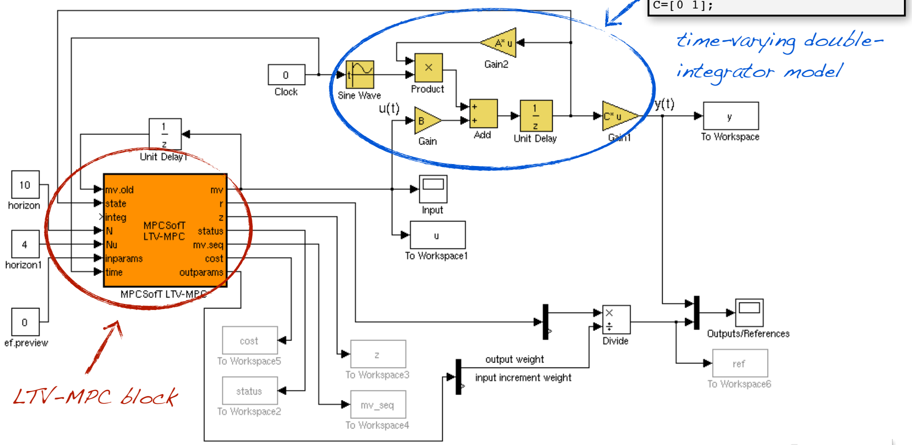
- Applications: time-varying systems (e.g.: aerospace), NL systems

Example: periodic double integrator

- **`mpcsoft_doubleint`** demo

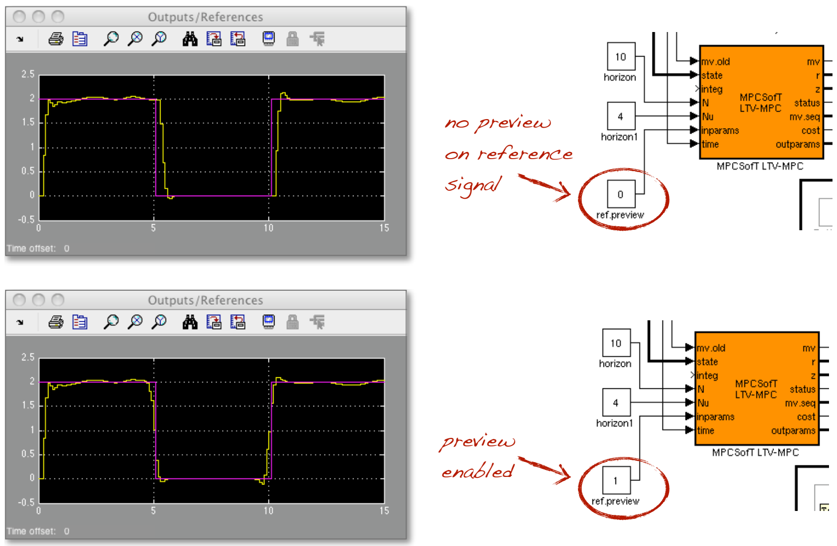
```
A=[1 0;Ts 1]*(1+.5*sin(t+k*Ts));
B=[Ts;0];
f=[0;0];
C=[0 1];
```

time-varying double-integrator model



$$\begin{array}{ll} \min & 10(x_{2,k} - r(t+k))^2 + 0.001\Delta u_k^2 \\ \text{s.t.} & \text{(no constraints)} \end{array}$$

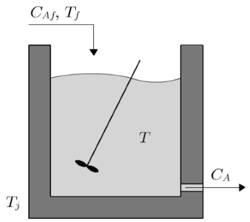
Example: double integrator



Example: LTV-MPC of a nonlinear CSTR system

- MPC control of a diabatic continuous stirred tank reactor (CSTR)
- Process model is rather nonlinear:

$$
\begin{aligned}
\frac{dC_A}{dt} &= \frac{F}{V}(C_{Af}-C_A) - C_A k_0 e^{-\frac{\Delta E}{RT}} \\
\frac{dT}{dt} &= \frac{F}{V}(T_f - T) + \frac{UA}{\rho C_p V}(T_j - T) - \frac{\Delta H}{\rho C_p} C_A k_0 e^{-\frac{\Delta E}{RT}}
\end{aligned}
$$



- T : temperature inside the reactor $[K]$ (state)
- C_A : concentration of the reagent in the reactor $[kgmol/m^3]$ (state)
- T_j : jacket temperature $[K]$ (input)
- T_f : feedstream temperature $[K]$ (measured disturbance)
- C_{Af} : feedstream concentration $[kgmol/m^3]$ (measured disturbance)

- Objective: manipulate T_j to keep C_A on the desired set-point

CSTR model

- Model is nonlinear and continuous in time. We approximate it to a discrete-time LTV model by on-line linearization:

generic NL model

$$\frac{dx}{dt} = f(x(t), u(t), t)$$

linearized model

$$
\begin{aligned}
\frac{dx}{dt}(t+\tau|t) \simeq\ & f(x(t),u(t),t) + \frac{\partial f}{\partial x}(x(t),u(t),t)(x(t+\tau)-x(t)) \\
&+ \frac{\partial f}{\partial u}(x(t),u(t),t)(u(t+\tau)-u(t)) \\
&+ \frac{\partial f}{\partial t}(x(t),u(t),t)\tau
\end{aligned}
$$

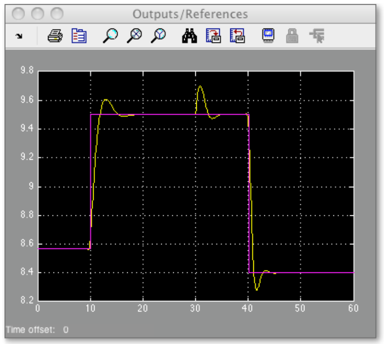
discrete-time LTV model

model matrices depend on current time t

$$
\begin{aligned}
x_{k+1} &= \underbrace{\left(I + T_s \frac{\partial f}{\partial x}(x(t),u(t),t)\right)}_{A(t)} x_k + \underbrace{T_s \frac{\partial f}{\partial u}(x(t),u(t),t)}_{B(t)} u_k + \underbrace{f_k}_{f(t)} \\
f_k &= T_s\Big(f(x(t),u(t),t) - \frac{\partial f}{\partial x}(x(t),u(t),t)x(t) - \frac{\partial f}{\partial u}(x(t),u(t),t)u(t) \\
&\quad + \frac{\partial f}{\partial t}(x(t),u(t),t)kT_s\Big)
\end{aligned}
$$

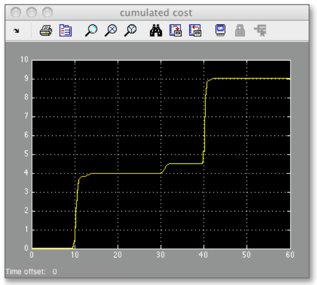
Simulation results

- Closed-loop trajectories: concentration C_A and desired set point r



Cumulated cost J

$$J(t+1) = J(t) + (C_A(t) - r(t))^2 + 10^{-4}\Delta T_j^2(t)$$



Numerical derivatives of cumulated cost J are a good "advisor" to fine tune the LTV-MPC design

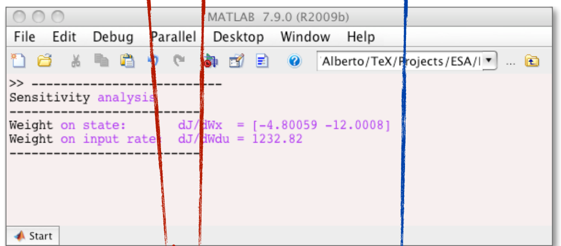
Sensitivities and controller retuning

- Compute numerical derivatives of $J(T_{final})$ with respect to weights on states and input increments:

$$\frac{\partial J}{\partial W_x} = \begin{bmatrix} -4.8 & -12.0 \end{bmatrix}, \quad \frac{\partial J}{\partial W_{\Delta u}} = 1233$$

- Original tuning:

$$W_x = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad W_{\Delta u} = 10^{-2}$$



- Adjust weights according to sensitivity

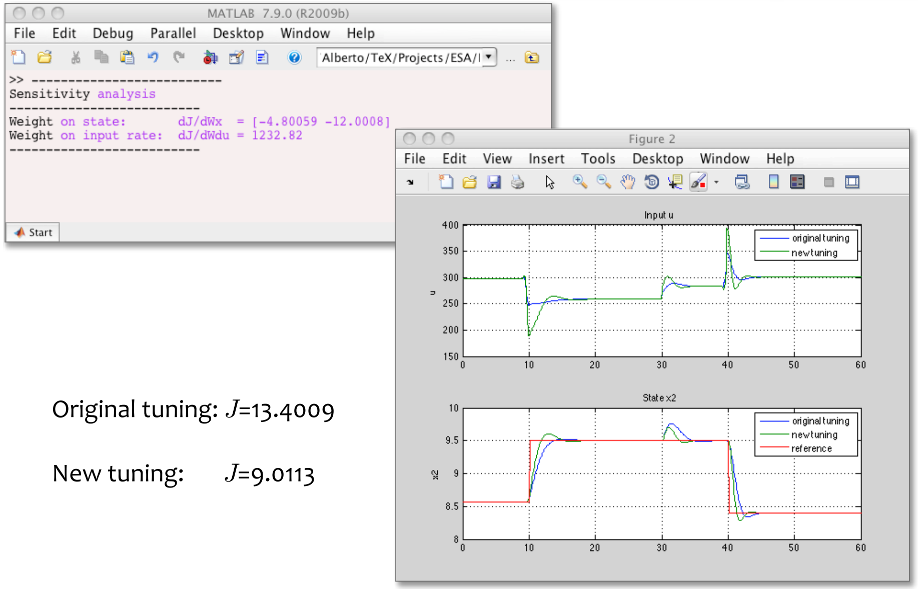
Negative sensitivities suggest to increase state weights

$$W_x = \begin{bmatrix} 10^{-4} & 2 \end{bmatrix}$$

Positive sensitivity suggests to decrease input-rate weight

$$W_{\Delta u} = 0.008$$

Example: CSTR demo - Performance tuning

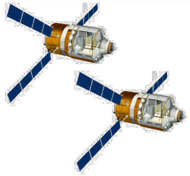


Original tuning: J=13.4009

New tuning: J=9.0113

Example: LTV-MPC for UAV navigation

- Goal: navigate two planar vehicles among obstacles to a target position



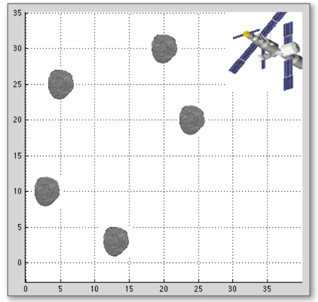
- The dynamical model of each vehicles is described by

$$p(t) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} p_c(t)$$

p= $[x,y]$ vehicle position
p_c= commanded position

- Vehicle dynamics converted to discrete-time by exact sampling

- Five square obstacles placed in workspace

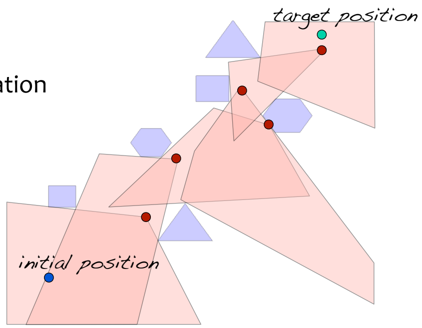


LTV-MPC for obstacle avoidance

- **Goal**: Use **linear time-varying (LTV) MPC** to generate desired positions to stabilizing controller in ***real-time*** to avoid **obstacles** and other UAVs

- **Problem**: feasible space is non-convex !

- **Main idea:** get a convex inner approximation of the feasible space that is
  - ***Polyhedral*** =linear constraints on predicted states in LTV-MPC problem
  - ***Very simple*** to compute on-line
  - Can handle polytopic obstacles of ***arbitrary shape*** and ***dimension***



- Alternative: non-convex feasible space modeled by mixed-integer constraints, guidance problem solved by (time-invariant) **hybrid MPC**

(Bemporad, Rocchi, IFAC 2011)

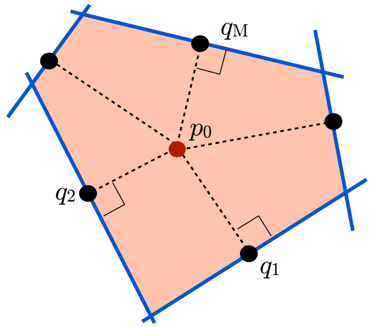
Fast greedy approximation algorithm (1/3)

(Bemporad, Rocchi, CDC 2011)

- Assume (for the moment) that vehicle and obstacles are points in space

$$p_0 \in \mathbb{R}^d = \text{current vehicle position}$$

$$q_1, q_2, \ldots, q_M \in \mathbb{R}^d = \text{obstacle positions}$$



- **Polyhedral approximation:**

$$P = \left\{ p \in \mathbb{R}^d : \begin{bmatrix} (q_1 - p_0)' \\ \vdots \\ (q_M - p_0)' \end{bmatrix} p \leq \begin{bmatrix} (q_1 - p_0)' q_1 \\ \vdots \\ (q_M - p_0)' q_M \end{bmatrix} \right\} \qquad p_0 \neq q_i,\ \forall i = 1, \ldots, M$$

- P contains p_0 and does not contain any of the obstacles q_1, q_2, ..., q_M in its interior

Fast greedy approximation algorithm (2/3)

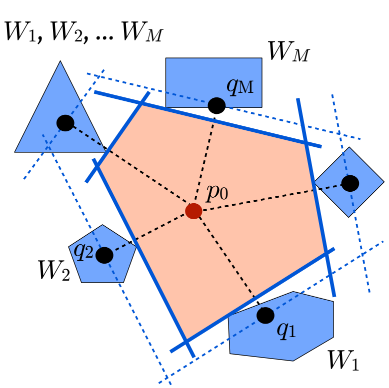
- Assume now **polyhedral obstacles** with shapes $W_1, W_2, \ldots W_M$
- For each obstacle j define the scalar g^j

$$g^j = \begin{array}{ll} \min_{w \in \mathbb{R}^d} & (q_j - p_0)' w \\ \text{s.t.} & w \in W_j \end{array}$$

- If the vertices of W_M are available, simply

$$g^j = \min_{h=1,\ldots,s_j} A_c^j w_{jh} \quad (w_{jh} = \text{vertex of } W_j)$$

- **Polyhedral approximation:**



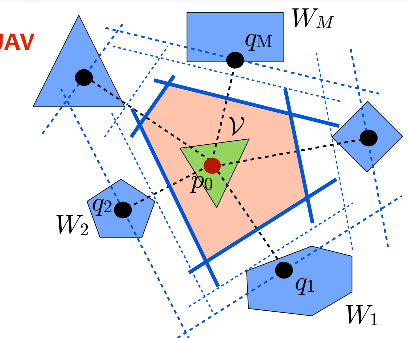
$$P = \left\{ p \in \mathbb{R}^d : \begin{bmatrix} (q_1 - p_0)' \\ \vdots \\ (q_M - p_0)' \end{bmatrix} p \leq \begin{bmatrix} (q_1 - p_0)' q_1 + g^1 \\ \vdots \\ (q_M - p_0)' q_M + g^M \end{bmatrix} \right\}$$

If P is nonempty, then P contains p_0 and does not contain any of the obstacles $B_j = \{q_j\} \oplus W_j$ in its interior

Fast greedy approximation algorithm (3/3)

- Assume polyhedral obstacles and **polytopic UAV**

$$\mathcal{V} \triangleq \text{conv}(p_0 + d_1, \ldots, p_0 + d_r)$$



- **Polyhedral approximation:**

$$P = \left\{ p \in \mathbb{R}^d : \begin{bmatrix} (q_1 - p_0)' \\ \vdots \\ (q_M - p_0)' \end{bmatrix} p \leq \begin{bmatrix} (q_1 - p_0)'(q_1 - d_i) + g^1 \\ \vdots \\ (q_M - p_0)'(q_M - d_i) + g^M \end{bmatrix}, \ i = 1, \ldots, r \right\}$$

If P is nonempty, then P contains $\mathcal{V}$ and does not contain any of the obstacles $B_j = \{q_j\} \oplus W_j$ in its interior

LTV-MPC guidance algorithm: prediction model

- Linear approximation of stabilized dynamics of UAV $\#i$

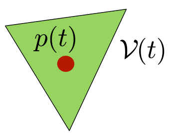
$$p_i(t+1) = A_i p_i(t) + B_i p_{ci}(t)$$

$$p_i(t) = \begin{bmatrix} x_i(t) \\ y_i(t) \\ z_i(t) \end{bmatrix}$$ position of vehicle $\#i$ at time t

$$p_{ci}(t) = \begin{bmatrix} x_{ci}(t) \\ y_{ci}(t) \\ z_{ci}(t) \end{bmatrix}$$ position commanded at time t

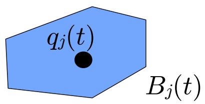
- Polytopic shape of UAV $\#i$ at time t

$$\mathcal{V}(t) = \{p(t)\} + \text{conv}(d_1(t), \ldots, d_r(t))$$



- Polyhedral shape of obstacle $\#j$ at time t

$$B_j(t) = \{q_j(t)\} \oplus W_j(t)$$



- Note: other vehicles in formation are treated as polytopic obstacles ($B_j = \mathcal{V}$)

LTV-MPC guidance algorithm: optimization model

- At time t, select the desired position $p_c(t)$ for the UAV by solving the optimal control problem (=**quadratic programming**)

prediction horizon

desired position

$$\min \quad \rho\epsilon^2 + \sum_{k=0}^{N-1} \|W^y(p_k - p_d(t))\|^2 + \|W^{\Delta u}(p_{c,k} - p_{c,k-1})\|^2$$

s.t. $p_{k+1} = A p_k + B p_{c,k}, \ k = 0, \ldots, N-1$ ← UAV dynamics

bounds on input increments → $\Delta_{\min} \leq p_{c,k} - p_{c,k-1} \leq \Delta_{\max}, \ k = 0, \ldots, N_u - 1$

$A_{c,k} p_k \leq b_{c,k} + g_k - A_{c,k} d_{h,k} + \mathbb{1}\epsilon$ ← (soft) obstacle avoidance constraints

$k = 0, \ldots, N-1, \ h = 1, \ldots, r_i$

$p_{c,k} = p_{c,N_u - 1}, \ k = N_u, \ldots, N$ ← blocked moves

$$A_c = \begin{bmatrix} (q_1 - p_0)' \\ \vdots \\ (q_M - p_0)' \end{bmatrix}, \ b_c = \begin{bmatrix} (q_1 - p_0)' q_1 \\ \vdots \\ (q_M - p_0)' q_M \end{bmatrix}$$

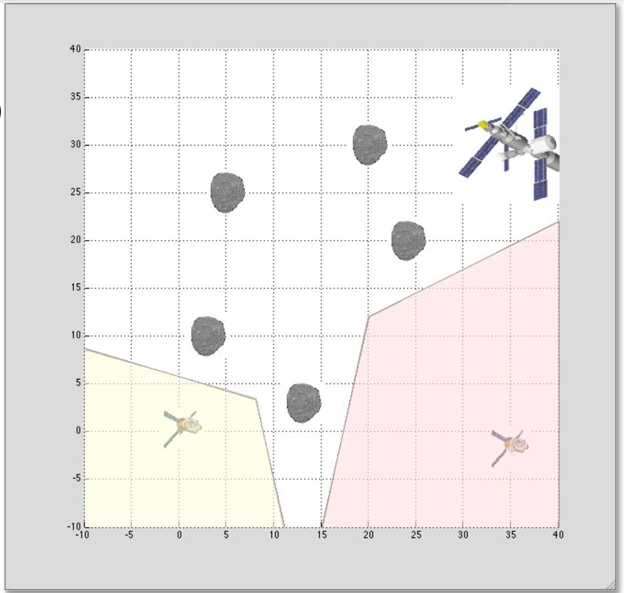
Example: navigation demo

- Initial position #1 = (0,0)
- Target position #1 = (35,30)
- Initial position #2 = (35,-3)
- Target position #2 = (0,20)

ROBMPC project
Robust Model Predictive Control (MPC) for Space Constrained Systems

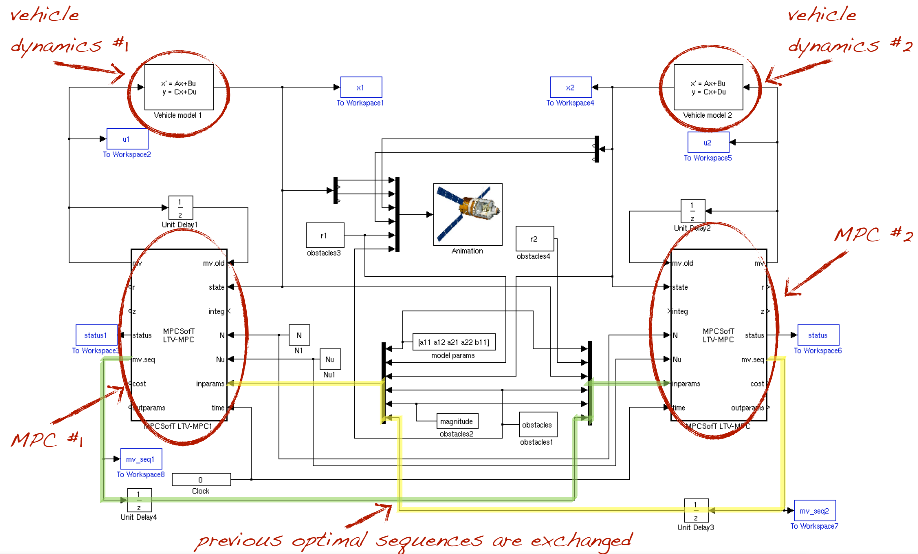
MPCSofT Toolbox for MATLAB

(Bemporad, 2010-2011)

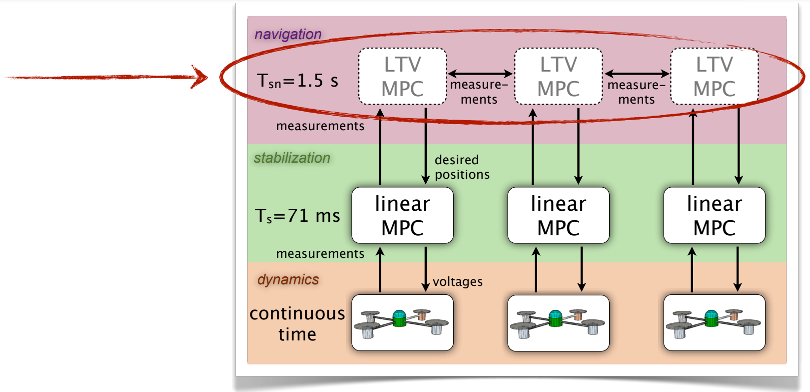


Example: navigation demo (cooperative MPC)

- Two vehicles avoiding each other and obstacles towards their targets



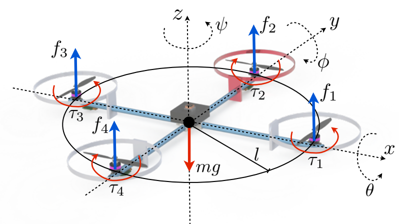
Decentralized LTV-MPC for formation flying



- **Linear-time varying (LTV) MPC** for UAVs flying in formation
- Hierarchical & decentralized structure (leader/follower approach)
- **Convex under-approximation** of the obstacle-free space

Nonlinear dynamical model of a quadcopter

- 4 command inputs: motor voltages $V_{Mi},\ i=1,\dots,4$
- 12 states $\theta,\phi,\psi,x,y,z,\dot\theta,\dot\phi,\dot\psi,\dot x,\dot y,\dot z$



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

$$
\begin{aligned}
f &= f_1+f_2+f_3+f_4\\
\tau_\theta &= (f_2-f_4)l\\
\tau_\phi &= (f_3-f_1)l
\end{aligned}
$$

$$
f_i = \frac{9.81(22.5V_{Mi}-9.7)}{1000},\ i=1,\dots,4
$$

- Highly nonlinear and coupled dynamics

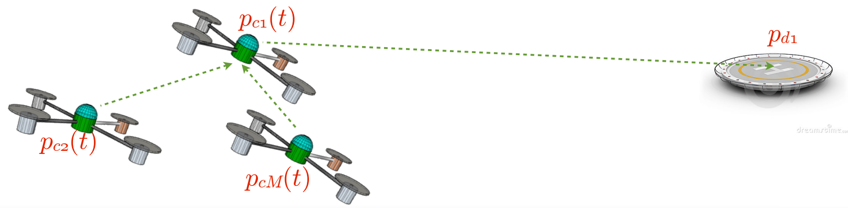
Linear MPC for stabilization

- Linear MPC control design based on MATLAB MPC Toolbox
- Linearization around the hovering condition and time-discretization
- Linear MPC setup:
  - Constraints:
    - Motor voltage saturation $0 \leq V_{Mi} \leq 11.1V$
    - Altitude $z \geq 0$
    - Pitch and roll angles $-\frac{\pi}{6} \leq \theta, \phi \leq \frac{\pi}{6}$ (soft constraints)
  - Sampling time $T_s = \frac{1}{14}s$
  - Prediction horizon $N_L = 20$
  - Control horizon $N_{Lu} = 3$
- Excellent stabilization properties, despite NL dynamics and constraints. Many other techniques available for stabilization purposes

(Castillo, Dzul, Losano, 2004)

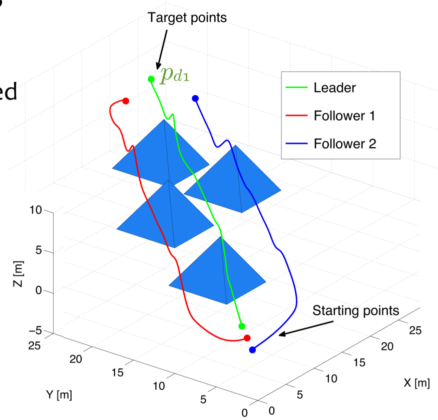
LTV-MPC guidance algorithm: formation flying

- At time t, each UAV $\#i$ solves its own LTV-MPC problem
- Vehicle #1 considers arrival point p_{d1} as desired target, $p_{d1}(t) \equiv p_{d1},\ \forall t \geq 0$
- Vehicle $\#j$ considers leader's position as desired target, $p_{dj}(t) = p_{c1}(t),\ \forall j > 1$
- Previous optimal sequences $p_{cj}(t|t-1), p_{cj}(t+1|t-1), \ldots, p_{cj}(t+N-2|t-1)$ are exchanged to predict future positions of other UAVs $\#j$



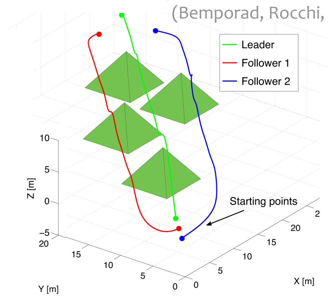
Example: LTV-MPC for formation flying

- 4 tetrahedral obstacles, $W = \mathrm{conv}\left(\begin{bmatrix}-1/3\\-1/3\\-1/2\end{bmatrix}, \begin{bmatrix}2/3\\-1/3\\-1/2\end{bmatrix}, \begin{bmatrix}-1/3\\2/3\\-1/2\end{bmatrix}, \begin{bmatrix}0\\0\\1/2\end{bmatrix}\right)$
- UAVs modeled as small parallelepipeds
- *MPCSofT Toolbox* used for LTV-MPC design, simulation, and code generation
- QP problem builder and solver implemented in Embedded MATLAB
- CPU time = ~**75 ms** (per time-step) (MATLAB R2009b, Macbook Air)



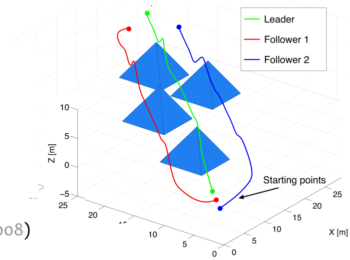
Comparison with other guidance methods

decentralized **hybrid** MPC (Bemporad, Rocchi, IFAC 2011)



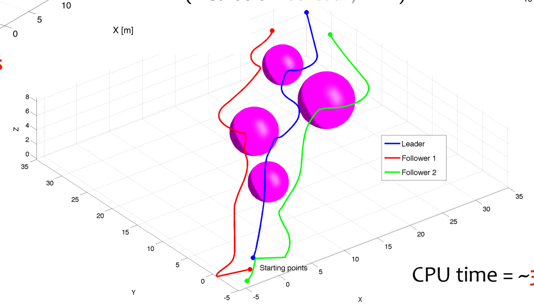
CPU time = ~**45 ms** (IBM CPLEX)

decentralized **LTV** MPC (this talk)



CPU time = ~**75 ms**

potential fields (method of Paul *et al.*, 2008)



CPU time = ~**3 ms**

Performance comparison

$J_{\mathrm{tt}} = \sum_{k=350}^{3080} \|p_L(k) - p_t\|_2^2$	*target tracking* Integral Square Error (ISE)
$J_{\mathrm{fpt}} = \sum_{k=350}^{3080} \|p_L(k) - p_{F1}(k) - p_{d1}\|_2^2 + \|p_L(k) - p_{F2}(k) - p_{d2}\|_2^2$	*formation pattern tracking* ISE
$J_{\mathrm{u}} = \sum_{k=350}^{3080} \|u(k) - u(k-1)\|_1$	*absolute derivative of input signals* (IADU)

	J_{tt}	J_{fpt}	J_{u}
centralized hybrid MPC	1	1	1
decentralized hybrid MPC	-0.09%	-0.51%	-0.03%
decentralized LTV MPC	-2.46%	-15.47%	+9.20%
potential fields	+210.32%	+294.30%	+212.75%

(indices normalized with respect to the centralized hybrid MPC performance)