1st HYCON PhD School on Hybrid Systems

www.unisi.it

www.ist-hycon.org

Model Predictive Control of Hybrid Systems

Alberto Bemporad

University of Siena, Italy

bemporad@dii.unisi.it

HYSCOM

IEEE CSS Technical Committee on Hybrid Systems

Information Society Technologies

Siena, July 19-22, 2005 - Rectorate of the University of Siena

Model Predictive Control of Hybrid Systems

Alberto Bemporad

Dept. Information Engineering - University of Siena, Italy

http://www.dii.unisi.it/~bemporad

University of Siena (founded in 1240)

EIHS – European Institute of Hybrid Systems

1st HYCON PhD School on Hybrid Systems (Siena, 20/7/05)

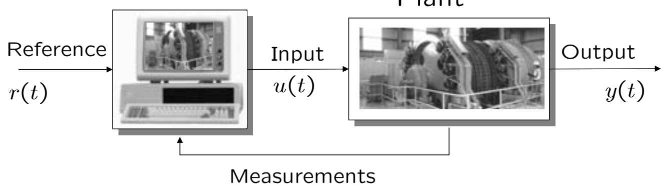
Outline

- Model Predictive Control (MPC) concepts
- Hybrid models for MPC
- MPC of hybrid systems
- Explicit MPC (multiparametric programming)
- Optimization-based reachability analysis
- Examples

Model Predictive Control

- MPC concepts
- Linear MPC
- Matlab tools for linear MPC

Model Predictive Control



- MODEL: a model of the plant is needed to predict the future behavior of the plant
- PREDICTIVE: optimization is based on the predicted future evolution of the plant
- CONTROL: control complex constrained multivariable plants

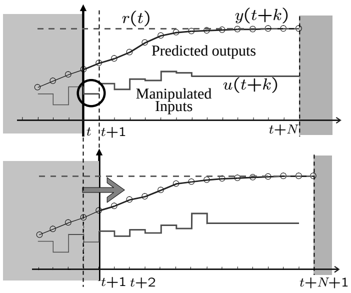
Receding Horizon Philosophy

- At time t:

  Solve an optimal control problem over a finite future horizon N:

  - minimize $f(|y-r|,|u|)$
  - subject to constraints

  $u_{\min} \leq u \leq u_{\max}$
  $y_{\min} \leq y \leq y_{\max}$

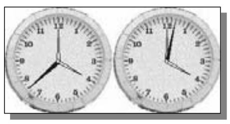


- Only apply the first optimal move $u^*(t)$
- Get new measurements, and repeat the optimization at time $t+1$

Advantage of on-line optimization: **FEEDBACK!**

Receding Horizon - Example

- MPC is like playing chess !



Constrained Optimal Control

- Linear Model: $\begin{cases} x(t+1) &= Ax(t)+Bu(t) \\ y(t) &= Cx(t) \end{cases}$ $\quad x \in \mathbb{R}^n,\ u \in \mathbb{R}^m$, $y \in \mathbb{R}^p$

- Constraints: $\begin{cases} u_{\min} \le u(t) \le u_{\max} \\ y_{\min} \le y(t) \le y_{\max} \end{cases}$

- Constrained optimal control problem (quadratic performance index):

$$\min_{u(0),\ldots,u(N-1)} \sum_{k=0}^{N-1} \left[x'(k)Qx(k) + u'(k)Ru(k)\right] + x'(N)Px(N)$$

$$\text{s.t. } u_{\min} \le u(k) \le u_{\max},\ k=0,\ldots,N-1$$
$$y_{\min} \le y(k) \le y_{\max},\ k=1,\ldots,N$$

$$Q = Q' \succeq 0,\ R = R' \succ 0,\ P \succeq 0$$

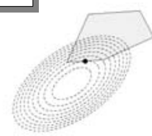
Constrained Optimal Control

- Optimization problem:

By substituting $x(k) = A^k x(0) + \sum_{j=0}^{k-1} A^{k-1-j}Bu(k)$, we get

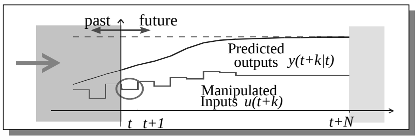
$$V(x(0)) = \tfrac{1}{2}x'(0)Yx(0) + \min_U \ \tfrac{1}{2}U'HU + x'(0)FU \quad \text{(quadratic)}$$
$$\text{s.t. } GU \le W + Sx(0) \quad \text{(linear)}$$

Convex QUADRATIC PROGRAM (QP)



- $U \triangleq [u'(0)\ \ldots\ u'(N-1)]' \in \mathbb{R}^s$, $s \triangleq Nm$, is the optimization vector
- $H = H' \succ 0$, and H, F, Y, G, W, S depend on weights Q, R, P, upper and lower bounds $u_{\min}$, $u_{\max}$, $y_{\min}$, $y_{\max}$, and model matrices A, B, C

MPC of Linear Systems



At time t:

- Get/estimate the current state $x(t)$
- Solve the QP problem $\min_U \ \tfrac{1}{2}U'HU + x'(t)FU$ s.t. $GU \le W + Sx(t)$

  and let U={$u^*(0)$,...,$u^*(N$-1)} be the solution (=finite-horizon constrained open-loop optimal control)
- Apply only $u(t) = u^*(0)$ and discard the remaining optimal inputs
- Go to time t+1

Model Predictive Control Toolbox

- **MPC Toolbox 2.0** (Bemporad, Ricker, Morari, 1998-2004):
  - Object-oriented implementation (MPC object)
  - MPC Simulink Library
  - MPC Graphical User Interface
  - RTW extension (code generation)

Only linear models are handled

http://www.mathworks.com

Example: AFTI-16

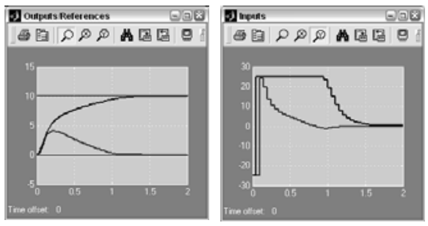
- Linearized model:

$$\begin{cases} \dot{x} = \begin{bmatrix} -.0151 & -60.5651 & 0 & -32.174 \\ -.0001 & -1.3411 & .9929 & 0 \\ .00018 & 43.2541 & -.86939 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} -2.516 & -13.136 \\ -.1689 & -.2514 \\ -17.251 & -1.5766 \\ 0 & 0 \end{bmatrix} u \\ y = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x, \end{cases}$$

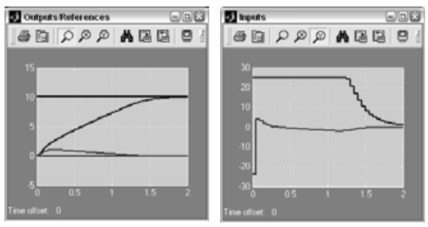
- Inputs: elevator and flaperon angle
- Outputs: attack and pitch angle
- Sampling time: $T_s = .05$ s (+ zero-order hold)
- Constraints: max 25° on both angles
- Open-loop response: unstable (open-loop poles: $-7.6636, -0.0075 \pm 0.0556j, 5.4530$)

see demo **afti16.m** (MPC-Tbx)

Example: AFTI-16



$N_y = 10,\ N_u = 3,$
$w_y = \{10, 10\},\ w_{\delta u} = \{.01, .01\},$
$u_{\min} = -25°,\ u_{\max} = 25°$



$N_y = 10,\ N_u = 3,$
$w_y = \{100, 10\},\ w_{\delta u} = \{.01, .01\},$
$u_{\min} = -25°,\ u_{\max} = 25°$

Convergence

Theorem 1 *Consider the linear system*

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

and the MPC control law based on

$$\min_U J(U,x(t)) = \sum_{k=0}^{N-1} \left\{ y'(t+k|t)Qy(t+k|t) + u'(t+k)Ru(t+k) \right\}$$

subj. to $y_{min} \le y(t+k) \le y_{max}$
$u_{min} \le u(t+k) \le u_{max}$

Assume that the optimization problem is feasible at time $t=0$. *Then, for either* $N \to \infty$ *or with the extra constraint* $x(t+N|t)=0$, *for all* $R>0$, $Q \ge 0$,

$$\lim_{t\to\infty} y(t) = 0, \quad \lim_{t\to\infty} u(t) = 0,$$

while fulfilling the constraints. Moreover, provided that $(Q^{\frac{1}{2}}C, A)$ *is a detectable pair,* $\lim_{t\to\infty} x(t) = 0$.

(Keerthi and Gilbert, 1988)(Bemporad *et al.*, 1994)

Proof: Use value function as Lyapunov function

Convergence Proof

- Assume we set the terminal constraint $x(t+N|t)=0$
- Let $\mathcal{U}_t^*$ denote the optimal control sequence @t $\{u_t^*(0),\ldots,u_t^*(N-1)\}$
- Let $V(t) \triangleq J(\mathcal{U}_t^*, x(t))$=value function ➡ Lyapunov function
- By construction, $\mathcal{U}_1 = \{u_t^*(1),\ldots,u_t^*(N-1),0\}$ is feasible @$t+1$, and hence

$$V(t+1) = J(\mathcal{U}^*{}_{t+1}, x(t+1)) \le J(\mathcal{U}_1, x(t+1)) = \\ = V(t) - y'(t)Qy(t) - u'(t)Ru(t)$$

- $V(t)$ is decreasing and lower-bounded by 0 $\Rightarrow \exists V_\infty = \lim_{t\to\infty} V(t) \Rightarrow V(t+1)-V(t) \to 0$, which implies $y'(t)Qy(t), u'(t)Ru(t) \to 0$
- Since $R>0$, $u(t) \to 0$
- Assume for simplicity that $C=I$ (i.e., $y(t)=x(t)$) and $Q>0$. Then also $x(t) \to 0$.

Global optimum is not needed to prove convergence !

Convergence Proof

If $Q \ge 0$ and/or $C \ne I$:

- For all $\forall k = 0,\ldots,n-1$, we have

$$\lim_{t\to\infty} y'(t+k)Qy(t+k) = \lim_{t\to\infty} \|Q^{\frac{1}{2}}C(A^k x(t) + \sum_{j=0}^{k-1} A^j Bu(t+k-1-j))\|^2 = 0$$

- As $u(t) \to 0$, also $Q^{\frac{1}{2}}CA^k x(t) \to 0$, and hence $\Theta x(t) \to 0$, where Θ is the observability matrix of $(Q^{\frac{1}{2}}C, A)$.
- If $(Q^{\frac{1}{2}}C, A)$ is observable, this also implies $x(t) \to 0$.
- If $(Q^{\frac{1}{2}}C, A)$ is only detectable, through a canonical decomposition one can observe that, as $u(t) \to 0$, unobservable modes go to zero spontaneously.

Convergence Proof

- Similar argument for infinite prediction horizon $N=\infty$:
  - Let $\mathcal{U}_t^*$ denote the infinite optimal control sequence @t $\{u_t^*(0), u_t^*(1)\ldots\}$
  - Let $V(t) \triangleq J(\mathcal{U}_t^*, x(t))$=value function ($\Rightarrow$ Lyapunov function)
  - Because constraints were checked up to $t+k=\infty$, $\mathcal{U}_1 = \{u_t^*(1), u_t^*(2),\ldots\}$ is feasible @$t+1$ by construction.
  - Hence

$$V(t+1) = J(\mathcal{U}^*{}_{t+1}, x(t+1)) \le J(\mathcal{U}_1, x(t+1)) = \\ = V(t) - y'(t)Qy(t) - u'(t)Ru(t)$$

  - Repeat same arguments as before

MPC and LQR

Jacopo Francesco Riccati (1676 - 1754)

- Consider the MPC control law:

$$\min_U J(U,t) = x'(t+T|t)Px(t+T|t) + \sum_{k=0}^{N-1} \left\{ x'(t+k|t)Qx(t+k|t) + u'(t+k)Ru(t+k) \right\}$$

subj. to $y_{min} \le y(t+k|t) \le y_{max},\ k=1,\ldots,N$
$u_{min} \le u(t+k) \le u_{max},\ k=0,\ldots,N-1$
$u(t+k) = Kx(t+k|t),\ k=N_u,\ldots,N-1$

$R=R'>0$, $Q=Q'\ge 0$, and P, K satisfy the Riccati equation

$$K = -(R+B'PB)^{-1}B'PA$$
$$P = (A+BK)'P(A+BK) + K'RK + Q$$

- In a polyhedral region around the origin the MPC control law is equivalent to the constrained LQR controller with weights Q,R. (Chmielewski, Manousiouthakis, 1996) (Scokaert and Rawlings, 1998)

MPC ≡ constrained LQR

- The larger the horizon, the larger the region where MPC=LQR

Outline

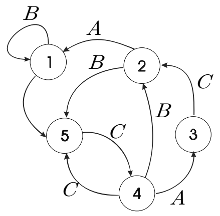
✓ Model Predictive Control (MPC) concepts
- Hybrid models for MPC
- MPC of hybrid systems
- Explicit MPC (multiparametric programming)
- Optimization-based reachability analysis
- Examples

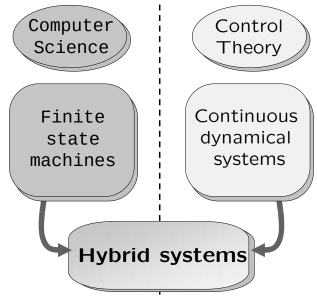
Hybrid Models for MPC

- Discrete Hybrid Automata (DHA)
- Mixed Logical Dynamical (MLD) Systems
- Piecewise Affine (PWA) Systems

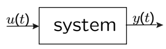
Hybrid Systems

$\mathbf{x} \in \{1,2,3,4,5\}$
$\mathbf{u} \in \{A,B,C\}$





$x \in \mathbb{R}^n$
$u \in \mathbb{R}^m$
$y \in \mathbb{R}^p$



$$\begin{cases} \frac{dx(t)}{dt} &= f(x(t),u(t)) \\ y(t) &= g(x(t),u(t)) \end{cases}$$

"Intrinsically Hybrid" Systems

Discrete input (1,N,2,3,4) + Continuous inputs (brakes, gas, clutch) + Continuous dynamical states (velocities, torques, air-flows, fuel level)

Cruise Control Problem

GOAL:

command gear ratio, gas pedal, and brakes to **track** a desired speed and minimize consumptions

CHALLENGES:

- continuous **and** discrete inputs
- dynamics depends on gear
- nonlinear torque/speed maps

Key Requirements for Hybrid Models

- **Descriptive** enough to capture the behavior of the system
  - continuous dynamics (physical laws)
  - logic components (switches, automata, software code)
  - interconnection between logic and dynamics
- **Simple** enough for solving *analysis* and *synthesis* problems

$$\begin{cases} x' &= Ax + Bu \\ y &= Cx + Du \end{cases}$$

linear systems

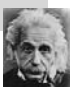
?

$$\begin{cases} x' &= f(x,u,t) \\ y &= g(x,u,t) \end{cases}$$

nonlinear systems

linear hybrid systems

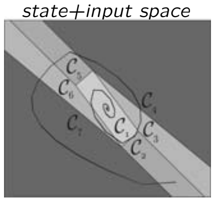
"Make everything as simple as possible, but not simpler."
— Albert Einstein



Piecewise Affine Systems

$$\begin{aligned} x(k+1) &= A_{i(k)}x(k) + B_{i(k)}u(k) + f_{i(k)} \\ y(k) &= C_{i(k)}x(k) + D_{i(k)}u(k) + g_{i(k)} \\ \\ i(k) \ \text{ s.t. } & \ H_{i(k)}x(k) + J_{i(k)}u(k) \leq K_{i(k)} \end{aligned}$$

state+input space



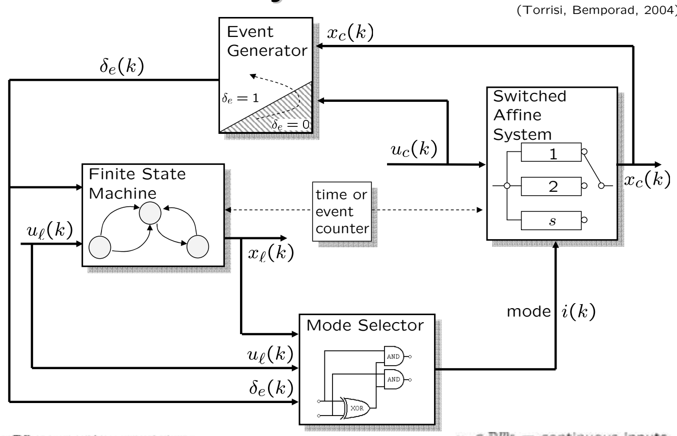
(Sontag 1981)

$x \in \mathcal{X} \subseteq \mathbb{R}^n$, $u \in \mathcal{U} \subseteq \mathbb{R}^m$, $y \in \mathcal{Y} \subseteq \mathbb{R}^p$
$i(k) \in \{1, \ldots, s\}$

- Can approximate nonlinear/discontinuous dynamics arbitrarily well

Discrete Hybrid Automaton

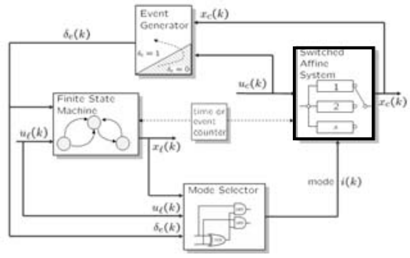
(Torrisi, Bemporad, 2004)



$x_c \in \mathbb{R}^{n_r}$ = continuous states
$x_\ell \in \{0,1\}^{n_b}$ = binary states
$i(k) \in \{1,\ldots,s\}$ = current mode

$u_c \in \mathbb{R}^{m_r}$ = continuous inputs
$u_\ell \in \{0,1\}^{m_b}$ = binary inputs
$\delta_e \in \{0,1\}^{n_e}$ = event conditions

Switched Affine System

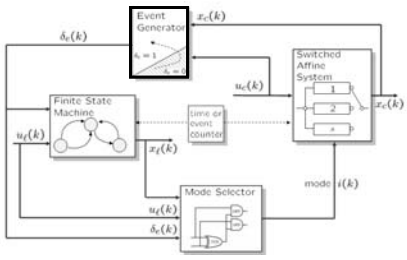


The affine dynamics depend on the current mode $i(k)$:

$$x_c(k+1) = A_{i(k)}x_c(k) + B_{i(k)}u_c(k) + f_{i(k)}$$

$x_c \in \mathbb{R}^{n_c},\ u_c \in \mathbb{R}^{m_c}$

Event Generator



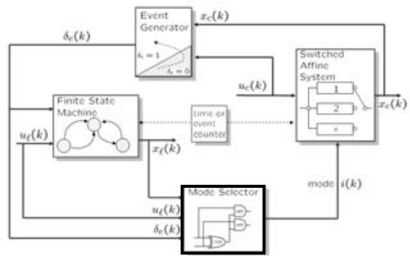
Event variables are generated by linear threshold conditions over continuous states, continuous inputs, and time:

$$[\delta_e^i(k) = 1] \leftrightarrow [H^i x_c(k) + K^i u_c(k) \leq W^i]$$

$x_c \in \mathbb{R}^{n_c},\ u_c \in \mathbb{R}^{m_c},\ \delta_e \in \{0,1\}^{n_e}$

Example: $[\delta{=}1] \leftrightarrow [x_c(k) \geq 0]$

Mode Selector



The active mode $i(k)$ is selected by a Boolean function of the current binary states, binary inputs, and event variables:

$$i(k) = f_{\mathsf{M}}(x_\ell(k), u_\ell(k), \delta_e(k))$$ $x_\ell \in \{0,1\}^{n_\ell},\ u_\ell \in \{0,1\}^{m_\ell},\ \delta_e\{0,1\}^{n_e}$

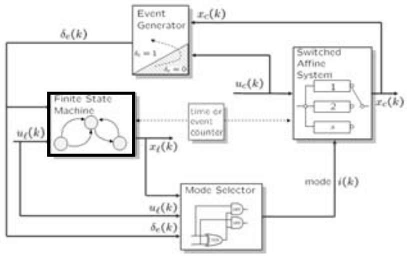
Example:

$$i(k) = \begin{bmatrix} \neg u_\ell(k) \vee x_\ell(k) \\ u_\ell(k) \wedge x_\ell(k) \end{bmatrix}$$

u_ℓ / x_ℓ	0	1
0	$i = \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$	$i = \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right]$
1	$i = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$	$i = \left[\begin{smallmatrix}1\\1\end{smallmatrix}\right]$

the system has 3 modes

Finite State Machine



The binary state of the finite state machine evolves according to a Boolean state update function:

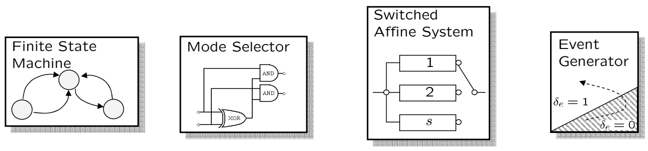
$$x_\ell(k+1) = f_{\mathsf{B}}(x_\ell(k), u_\ell(k), \delta_e(k))$$ $x_\ell \in \{0,1\}^{n_\ell},\ u_\ell \in \{0,1\}^{m_\ell},\ \delta_e \in \{0,1\}^{n_e}$

Example: $x_\ell(k+1) = \neg\delta_e(k) \vee (x_\ell(k) \wedge u_\ell(k))$

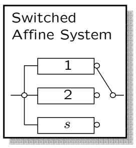
Logic and Inequalities

Glover 1975, Williams 1977

$X_1 \vee X_2$	$\delta_1 + \delta_2 \geq 1, \quad \delta_1, \delta_2 \in \{0,1\}$
Any logic statement $f(X)$ =TRUE $\bigwedge_{j=1}^{m} \left(\vee_{i\in P_j} X_i \vee_{i \in N_j} \neg X_i\right)$ (CNF) $N_j, P_j \subseteq \{1,\ldots,n\}$	$A\delta \leq B$ $\begin{cases} 1 \leq \sum_{i\in P_1} \delta_i + \sum_{i \in N_1} (1-\delta_i) \\ \vdots \\ 1 \leq \sum_{i\in P_m} \delta_i + \sum_{i \in N_m} (1-\delta_i) \end{cases}$
$[\delta_e^i(k) = 1] \leftrightarrow [H^i x_r(k) \leq W^i]$	$H^i x_r(k) - W^i \leq M^i(1-\delta_e^i)$ $H^i x_r(k) - W^i > m^i \delta_e^i$
IF $[\delta = 1]$ THEN $z = a_1^T x + b_1^T u + f_1$ ELSE $z = a_2^T x + b_2^T u + f_2$	$(m_2 - M_1)\delta + z \leq a_2 x + b_2 u + f_2$ $(m_1 - M_2)\delta - z \leq -a_2 x - b_2 u - f_2$ $(m_1 - M_2)(1-\delta) + z \leq a_1 x + b_1 u + f_1$ $(m_2 - M_1)(1-\delta) - z \leq -a_1 x - b_1 u - f_1$



Switched Affine Systems



The state-update equation can be rewritten as a combination of affine terms and *if-then-else* conditions:

$$z_1(t) = \begin{cases} A_1 x_c(t) + B_1 u_c(t) + f_1, & \text{if } (i(t) = 1) \\ 0, & \text{otherwise,} \end{cases}$$
$$\vdots$$
$$z_s(t) = \begin{cases} A_s x_c(t) + B_s u_c(t) + f_s, & \text{if } (i(t) = s), \\ 0, & \text{otherwise,} \end{cases}$$
$$x_c(t+1) = \sum_{i=1}^{s} z_i(t),$$

where $z_i(t) \in \mathbb{R}^{n_c}, i = 1, \ldots, s.$

If-then-else conditions are converted to inequalities.

Output equations $y_c(t)=C_i x_c(t)+D_i u_c(t)+g_i$ admit a similar transformation.

Logic → Inequalities: Symbolic Approach

0. Given a Boolean statement $F(X_1, X_2, \ldots, X_n) = \text{TRUE}$

1. Convert to Conjunctive Normal Form (CNF):

$$\bigwedge_{j=1}^{m} \left(\bigvee_{i \in P_j} X_i \bigvee_{i \in N_j} \bar{X}_i \right) = \text{TRUE}$$

2. Transform into inequalities:

$$\begin{array}{rcl} \sum_{i \in P_1} \delta_i + \sum_{i \in N_1} (1 - \delta_i) & \geq & 1 \\ \vdots & \vdots & \vdots \\ \sum_{i \in P_m} \delta_i + \sum_{i \in N_m} (1 - \delta_i) & \geq & 1 \end{array} \quad \Rightarrow \quad \text{polyhedron } A\delta \leq b, \quad \delta \in \{0,1\}^n$$

Any logic proposition can be translated into linear integer ineq.

Logic → Inequalities: Symbolic Approach

Example: $F(X_1, X_2, X_3) = [X_3 \leftrightarrow X_1 \wedge X_2]$

1. Convert to Conjunctive Normal Form (CNF):
   (see e.g.: **http://www.oursland.net/aima/propositionApplet.html**
   or just search [CNF + applet] on Google ...)

$$(X_3 \vee \neg X_1 \vee \neg X_2) \wedge (X_1 \vee \neg X_3) \wedge (X_2 \vee \neg X_3)$$

2. Transform into inequalities:

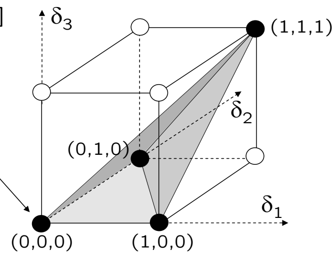
$$\begin{cases} \delta_3 + (1 - \delta_1) + (1 - \delta_2) \geq 1 \\ \delta_1 + (1 - \delta_3) \geq 1 \\ \delta_2 + (1 - \delta_3) \geq 1 \end{cases}$$

Logic → Inequalities: Geometric Approach

Example: logic "AND"
$F(X_1, X_2, X_3) = [X_3 \leftrightarrow X_1 \wedge X_2]$

T:

X_1	X_2	X_3
0	0	0
0	1	0
1	0	0
1	1	1



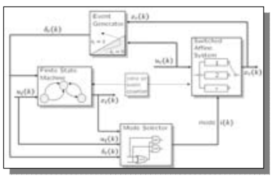
Key idea:
White points cannot be in the hull of black points

$$\text{conv}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \right) = \left\{ \delta : \begin{array}{rl} -\delta_1 + \delta_3 & \leq 0 \\ -\delta_2 + \delta_3 & \leq 0 \\ \delta_1 + \delta_2 - \delta_3 & \leq 1 \end{array} \right\}$$

Convex hull algorithms: **cdd, lrs, qhull, chD, Hull, Porto**

Mixed Logical Dynamical Systems

Discrete Hybrid Automaton



HYSDEL
(Torrisi, Bemporad, 2004)

Mixed Logical Dynamical (MLD) Systems (Bemporad, Morari 1999)

$$\begin{array}{rcl} x(t+1) & = & Ax(t) + B_1 u(t) + B_2 \delta(t) + B_3 z(t) + B_5 \\ y(t) & = & Cx(t) + D_1 u(t) + D_2 \delta(t) + D_3 z(t) + D_5 \\ E_2 \delta(t) + E_3 z(t) & \leq & E_4 x(t) + E_1 u(t) + E_5 \end{array}$$

Continuous and binary variables: $x \in \mathbb{R}^{n_r} \times \{0,1\}^{n_b}$, $u \in \mathbb{R}^{m_r} \times \{0,1\}^{m_b}$, $y \in \mathbb{R}^{p_r} \times \{0,1\}^{p_b}$, $\delta \in \{0,1\}^{r_b}$, $z \in \mathbb{R}^{r_r}$

- Computationally oriented (mixed-integer programming)
- Suitable for controller synthesis, verification, ...

A Simple Example

- System: $x(k+1) = \begin{cases} 0.8x(k) + u(k) & \text{if } x(k) \geq 0 \\ -0.8x(k) + u(k) & \text{if } x(k) < 0 \end{cases}$

  $-10 \leq x(k) \leq 10, \ -1 \leq u(k) \leq 1$

- Associate $[\delta(k) = 1] \leftrightarrow [x(k) \geq 0]$ and transform

$$\longrightarrow \quad \begin{array}{rcl} x(k) & \geq & m(1 - \delta(k)) \\ x(k) & \leq & -\epsilon + (M + \epsilon)\delta(k) \end{array} \qquad \begin{array}{l} M = -m = 10 \\ \epsilon > 0 \text{ "small"} \end{array}$$

- Then $x(k+1) = 1.6\boxed{\delta(k)x(k)} - 0.8x(k) + u(k)$

$$z(k) = \delta(k)x(k) \longrightarrow \begin{array}{rcl} z(k) & \leq & M\delta(k) \\ z(k) & \geq & m\delta(k) \\ z(k) & \leq & x(k) - m(1 - \delta(k)) \\ z(k) & \geq & x(k) - M(1 - \delta(k)) \end{array}$$

- Rewrite as a linear equation

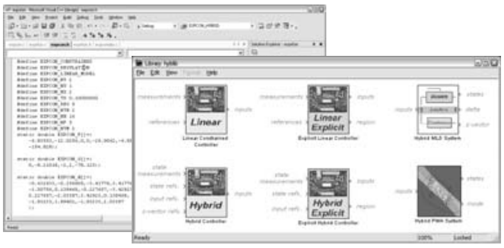
$$\longrightarrow \quad x(k+1) = 1.6z(k) - 0.8x(k) + u(k)$$

Hybrid Toolbox for Matlab

(Bemporad, 2003-2004)

Features:

- Hybrid model (MLD and PWA) design and simulation
- Control design for linear systems w/ constraints and hybrid systems (on-line optimization via QP/MILP/MIQP)
- Explicit control (via multiparametric programming)
- C-code generation
- Simulink



http://www.dii.unisi.it/hybrid/toolbox

HYSDEL

(HYbrid Systems DEscription Language)

- Describe *hybrid systems:*

  – Automata
  – Logic
  – Lin. Dynamics
  – Interfaces
  – Constraints

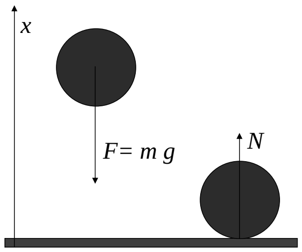
(Torrisi, Bemporad, 2004)

- Automatically generate MLD models in Matlab

Download: **http://www.dii.unisi.it/hybrid/toolbox**

Reference: **http://control.ethz.ch/~hybrid/hysdel**

Example: Bouncing Ball



$$\ddot{x} = -g$$

$$x \leq 0 \Rightarrow \dot{x}(t^+) = -(1-\alpha)\dot{x}(t^-)$$

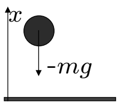
$$\alpha \in [0,1]$$

How to model this system in MLD form?

Bouncing Ball – Time Discretization

$x(k) > 0:$

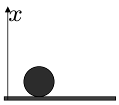
$$v(k) \approx \frac{x(k)-x(k-1)}{T_s} \qquad -g = \ddot{x}(k) \approx \frac{v(k)-v(k-1)}{T_s}$$



$$\left\{\begin{array}{rcl} v(k+1) &=& v(k) - T_s g \\ x(k+1) &=& x(k) + T_s v(k+1) \\ &=& x(k) + T_s v(k) - T_s^2 g \end{array}\right.$$

$x(k) \leq 0:$

$$v(k) = -(1-\alpha)v(k-1) \qquad \begin{array}{rcl} x(k+1) &=& x(k-1) \\ &=& x(k) - T_s v(k) \end{array}$$



$$\left\{\begin{array}{rcl} v(k+1) &=& -(1-\alpha)v(k) \\ x(k+1) &=& x(k) - T_s v(k) \end{array}\right.$$

HYSDEL - Bouncing Ball

```
SYSTEM bouncing_ball {
INTERFACE {
/* Description of variables and constants */
        STATE { REAL height [-10,10];
                REAL velocity [-100,100];   }

        PARAMETER {
                REAL g;
                REAL dissipation;  /* 0=elastic, 1=completely anelastic */
                REAL Ts; }
}
IMPLEMENTATION {
        AUX {   REAL z1;
                REAL z2;
                BOOL negative;   }

        AD {    negative = height <= 0; }

        DA {    z1 = {  IF negative THEN height-Ts*velocity
                        ELSE height+Ts*velocity-Ts*Ts*g};
                z2 = {  IF negative THEN -(1-dissipation)*velocity
                        ELSE velocity-Ts*g};  }
        CONTINUOUS {
                height = z1;
                velocity=z2;}
}}
```

go to demo **/demos/hybrid/bball.m**

Systems Theory for Discrete-Time Linear Hybrid Systems

- Analysis
  – Well-posedness
  – Realization & Transformation
  – Reachability (=Verification)
  – Observability
  – Stability
  – Identification

- Synthesis
  – Control
  – State estimation

Well-posedness

Are state and output trajectories defined ? Uniquely defined ? Persistently defined ?

- MLD well-posedness :

$$\begin{aligned} x(t+1) &= Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) \\ y(t) &= Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t) \\ E_2\delta(t) + E_3z(t) &\leq E_1u(t) + E_4x(t) + E_5 \end{aligned}$$

$\delta(t) = F(x(t), u(t))$

$z(t) = G(x(t), u(t))$

$\{x(t), u(t)\} \to \{x(t+1)\}$

$\{x(t), u(t)\} \to \{y(t)\}$

are single valued

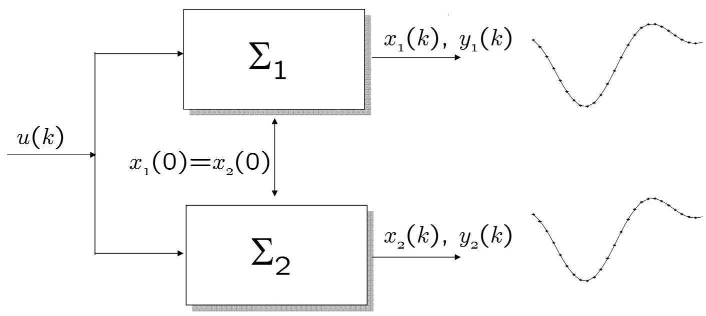
Definition 1 *Let $\Omega \subseteq \mathbb{R}^n \times \mathbb{R}^m$ be a set of input+state pairs. A hybrid MLD system is called* well-posed *on Ω, if for all pairs $(x(t), u(t)) \in \Omega$ there exists a solution $x(t+1), y(t), \delta(t), z(t)$ and moreover, $x(t+1), y(t)$ are uniquely determined.*

Numerical test based on mixed-integer programming available

(Bemporad, Morari, *Automatica*, 1999)

Equivalences of Hybrid Models

Definition 1 *Two hybrid systems Σ_1, Σ_2 are* equivalent *if for all initial conditions $x_1(0) = x_2(0)$ and input $\{u_1(k)\}_{k\in\mathbb{Z}_+} = \{u_2(k)\}_{k\in\mathbb{Z}_+}$ then $x_1(k) = x_2(k)$ and $y_1(k) = y_2(k)$, for all $k \in \mathbb{Z}_+$.*



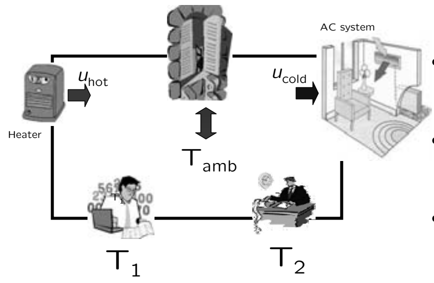
MLD and PWA Systems

Theorem MLD systems and PWA systems are equivalent

(Bemporad, Ferrari-Trecate, Morari, *IEEE TAC,2000*)

- Proof is constructive: given an MLD system it returns its equivalent PWA form
- Drawback: it needs the enumeration of all possible combinations of binary states, binary inputs, and δ variables
- Most of such combinations lead to empty regions
- Efficient algorithms are available for converting MLD models into PWA models avoiding such an enumeration:
  - A. Bemporad, "*Efficient Algorithms for Converting Mixed Logical Dynamical Systems into an Equivalent Piecewise Affine Form*", IEEE Trans. Autom. Contr., 2004.
  - T. Geyer, F.D. Torrisi and M. Morari, "*Efficient Mode Enumeration of Compositional Hybrid Models*", HSCC'03

Example: Room Temperature



Hybrid Dynamics

- #1 turns the heater (air conditioning) on whenever he is cold (hot)
- If #2 is cold he turns the heater on, unless #1 is hot
- If #2 is hot he turns the air conditioning on, unless #2 is cold
- Otherwise, heater and air conditioning are off

- $\dot{T}_1 = -\alpha_1(T_1 - T_{\text{amb}}) + k_1(u_{\text{hot}} - u_{\text{cold}})$ (body temperature dynamics of #1)
- $\dot{T}_2 = -\alpha_2(T_2 - T_{\text{amb}}) + k_2(u_{\text{hot}} - u_{\text{cold}})$ (body temperature dynamics of #2)

go to demo **/demos/hybrid/heatcool.m**

HYSDEL Model

```
SYSTEM heatcool {

INTERFACE {
    STATE { REAL T1 [-10,50];
            REAL T2 [-10,50];
        }
    INPUT { REAL Tamb [-10,50];
        }
    PARAMETER {
        REAL Ts, alpha1, alpha2, k1, k2;
        REAL Thot1, Tcold1, Thot2, Tcold2, Uc, Uh;
        }
}
IMPLEMENTATION {
        AUX { REAL uhot, ucold;
              BOOL hot1, hot2, cold1, cold2;
        }
        AD  { hot1 = T1>=Thot1;
              hot2 = T2>=Thot2;
              cold1 = T1<=Tcold1;
              cold2 = T2<=Tcold2;
        }
        DA  { uhot = (IF cold1 | (cold2 & ~hot1) THEN Uh ELSE 0);
              ucold = (IF hot1 | (hot2 & ~cold1) THEN Uc ELSE 0);
        }
        CONTINUOUS { T1 = T1+Ts*(-alpha1*(T1-Tamb)+k1*(uhot-ucold));
                     T2 = T2+Ts*(-alpha2*(T2-Tamb)+k2*(uhot-ucold));
        }
    }
}
```

Hybrid Toolbox for Matlab

(Bemporad, 2003-2005)

http://www.dii.unisi.it/hybrid/toolbox

`>>S=mld('heatcoolmodel',Ts)` get the MLD model in Matlab

`>>[XX,TT]=sim(S,x0,U);` simulate the MLD model

Hybrid MLD Model

- MLD model

$$\begin{aligned} x(t+1) &= Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) \\ y(t) &= Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t) \\ E_2\delta(t) + E_3z(t) &\leq E_1u(t) + E_4x(t) + E_5 \end{aligned}$$

- 2 continuous states: (temperatures T_1,T_2)
- 1 continuous input: (room temperature T_{amb})
- 2 auxiliary continuous vars: (power flows u_{hot}, u_{cold})
- 6 auxiliary binary vars: (4 thresholds + 2 for OR condition)
- 20 mixed-integer inequalities

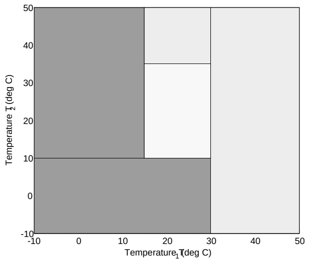
Possible combination of integer variables: $2^6 = 64$

Hybrid PWA Model

- PWA model

$$
\begin{aligned}
x(k+1) &= A_{i(k)}x(k) + B_{i(k)}u(k) + f_{i(k)} \\
y(k) &= C_{i(k)}x(k) + D_{i(k)}u(k) + g_{i(k)} \\
i(k) \ \text{s.t.} \ & H_{i(k)}x(k) + J_{i(k)}u(k) \leq K_{i(k)}
\end{aligned}
$$

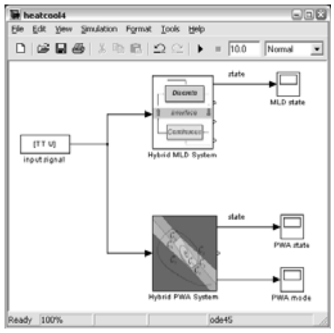
- 2 continuous states: (temperatures T_1, T_2)
- 1 continuous input: (room temperature T_{amb})
- 5 polyhedral regions *(partition does not depend on input)*

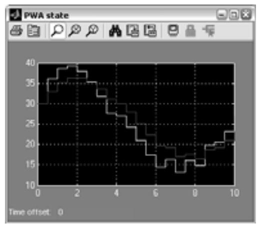


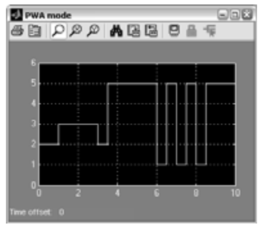
$u_{hot} = 0$, $u_{cold} = 0$ | $u_{hot} = 0$, $u_{cold} = \bar{U}_C$ | $u_{hot} = \bar{U}_H$, $u_{cold} = 0$

```
>>P=pwa(S);
```


Simulation in Simulink







Other Existing Hybrid Models

- Linear complementarity (LC) systems (Heemels, 1999)

$$
\begin{aligned}
x(t+1) &= Ax(t) + B_1u(t) + B_2w(t) \\
y(t) &= Cx(t) + D_1u(t) + D_2w(t) \\
v(t) &= E_1x(t) + E_2u(t) + E_3w(t) + e_4 \\
0 &\leq v(t) \perp w(t) \geq 0
\end{aligned}
$$

Ex: mechanical systems, circuits with diodes etc.

- Extended linear complementarity (ELC) systems
  Generalization of LC systems (De Schutter, De Moor, 2000)

- Min-max-plus-scaling (MMPS) systems (De Schutter, Van den Boom, 2000)

$$
\begin{aligned}
x(t+1) &= M_x(x(t), u(t), d(t)) \\
y(t) &= M_y(x(t), u(t), d(t)) \\
0 &\geq M_c(x(t), u(t), d(t))
\end{aligned}
$$

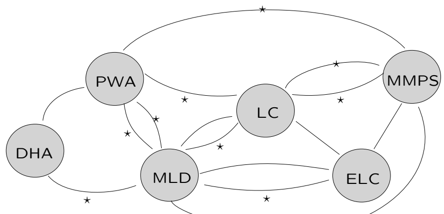
MMPS function: defined by the grammar

$M := x_i | \alpha | \max(M_1, M_2) | \min(M_1, M_2) | M_1 + M_2 | \beta M_1$

Example: $x(t+1) = 2\max(x(t), 0) + \min(-\frac{1}{2}u(t), 1)$

Used for modeling discrete-event systems (t=event counter)

Equivalence Results



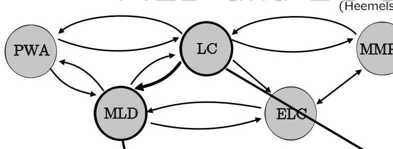
Theorem *All the above six classes of discrete-time hybrid models are equivalent (possibly under some additional assumptions, such as boundedness of input and state variables)*

(Heemels, De Schutter, Bemporad, *Automatica*, 2001)
(Torrisi, Bemporad, *IEEE CST*, 2003)
(Bemporad and Morari, *Automatica*, 1999)
(Bemporad, Ferrari-T.,Morari, *IEEETAC*, 2000)

Theoretical properties and analysis/synthesis tools can be transferred from one class to another

MLD and LC Systems

(Heemels, De Schutter, Bemporad, *Automatica*, 2001)



Theorem 1 *Every LC system can be written as an MLD system, provided that the variables $w(k)$ and $v(k)$ are (componentwise) bounded.*

Proof:

$$
\begin{aligned}
x(t+1) &= Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) \\
y(t) &= Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t) \\
E_2\delta(t) + E_3z(t) &\leq E_4x(t) + E_1u(t) + E_5
\end{aligned}
$$

$$
\begin{aligned}
x(t+1) &= Ax(t) + B_1u(t) + B_2w(t) \\
y(t) &= Cx(t) + D_1u(t) + D_2w(t) \\
v(t) &= E_1x(t) + E_2u(t) + E_3w(t) + e_4 \\
0 &\leq v(t) \perp w(t) \geq 0
\end{aligned}
$$

For each complementarity pair $v_i(t), w_i(t)$ introduce a binary variable $\delta_i(t) \in \{0,1\}$

$[\delta_i(t) = 1] \rightarrow [v_i(t) = 0, \ w_i(t) \geq 0]$
$[\delta_i(t) = 0] \rightarrow [v_i(t) \geq 0, \ w_i(t) = 0]$

⇒

$$
\begin{aligned}
w_i(t) &\leq M\delta_i(t) \\
v_i(t) &\leq M(1 - \delta_i(t)) \\
w_i(t) &\geq 0 \\
v_i(t) &\geq 0
\end{aligned}
$$

Set $z_i(t) = w_i(t)$ and substitute $v(t) = E_1x(t) + E_2u(t) + E_3w(t) + e_4$ ■

Hybrid System Identification

- Sometimes a *hybrid model* of the process (or of a part of it) cannot be derived manually from available knowledge.
- Therefore, a model must be either
  - Estimated from data (model unknown)
  - or *hybridized* before it can be used for control/analysis (model known but nonlinear)
- If a linear model is enough, no problem: several algorithms are available (e.g.: use Ljung's ID TBX)
- If switching modes are known and data can be generated for each mode, no problem: we identify one linear model per mode (e.g.: use Ljung's ID TBX)
- If modes & dynamics must be identified together, we need

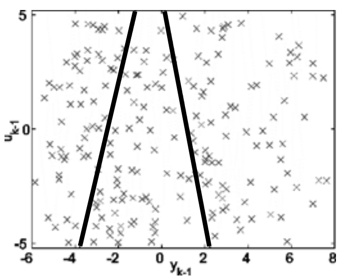
hybrid system identification

PWA Identification Problem

> Estimate from data **both** the parameters of the affine submodels **and** the partition of the PWA map

Example Let the data be generated by the PWARX system

$$y_k = \begin{cases} \begin{bmatrix} -0.4 & 1 & 1.5 \end{bmatrix} \varphi_k + \varepsilon_k \\ \quad \text{if } \begin{bmatrix} 4 & -1 & 10 \end{bmatrix} \varphi_k < 0 \\ \begin{bmatrix} 0.5 & -1 & -0.5 \end{bmatrix} \varphi_k + \varepsilon_k \\ \quad \text{if } \begin{bmatrix} -4 & 1 & -10 \\ 5 & 1 & -6 \end{bmatrix} \varphi_k \leq 0 \\ \begin{bmatrix} -0.3 & 0.5 & -1.7 \end{bmatrix} \varphi_k + \varepsilon_k \\ \quad \text{if } \begin{bmatrix} -5 & -1 & 6 \end{bmatrix} \varphi_k < 0 \end{cases}$$



with $\varphi_k = [y_{k-1}\ u_{k-1}\ 1]'$, $|u_k| \leq 5$ and $|\varepsilon_k| \leq 0.1$

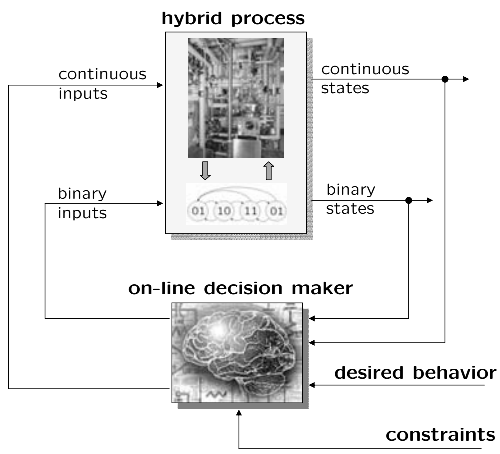
Outline

- ✓ Model Predictive Control (MPC) concepts
- ✓ Hybrid models for MPC
- MPC of hybrid systems
- Explicit MPC (multiparametric programming)
- Optimization-based reachability analysis
- Examples

MPC of Hybrid Systems

- Problem setup
- Convergence properties
- Computational aspects
- Matlab tools for hybrid MPC

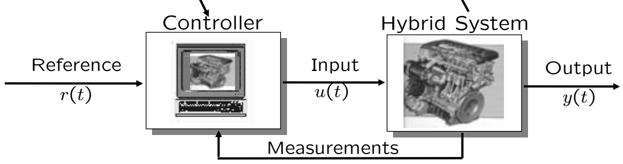
Hybrid Control Problem



Model Predictive Control of Hybrid Systems

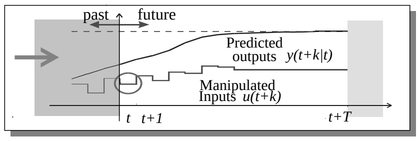
MLD model

$$\begin{aligned} x(t+1) &= Ax(t) + B_1 u(t) + B_2 \delta(t) + B_3 z(t) \\ y(t) &= Cx(t) + D_1 u(t) + D_2 \delta(t) + D_3 z(t) \\ E_2 \delta(t) + E_3 z(t) &\leq E_4 x(t) + E_1 u(t) + E_5 \end{aligned}$$



- MODEL: use an MLD or PWA model of the plant to predict the future behavior of the hybrid system
- PREDICTIVE: optimization is still based on the predicted future evolution of the hybrid system
- CONTROL: the goal is to control the hybrid system

MPC for Hybrid Systems



Model Predictive (MPC) Control

- At time t solve with respect to $U \triangleq \{u(t), u(t+1), \ldots, u(t+T-1)\}$ the finite-horizon open-loop, optimal control problem:

$$\min_{u(t),\ldots,u(t+T-1)} \sum_{k=0}^{T-1} \|y(t+k|t) - r(t)\| + \rho \|u(t+k)\| + \sigma(\|\delta(t+k) - \delta_r\| + \|z(t+k) - z_r\| + \|x(t+k|t) - x_r\|)$$

subject to MLD model

$$x(t|t) = x(t)$$

$$x(t+T|t) = x_r$$

- Apply only $u(t) = u^*(t)$ (discard the remaining optimal inputs)
- Repeat the whole optimization at time $t+1$

Closed-Loop Convergence

Theorem 1 *Let $(x_r, u_r, \delta_r, z_r)$ be the equilibrium values corresponding to the set point r, and assume $x(0)$ is such that the MPC problem is feasible at time $t = 0$. Then* $\forall Q, R \succ 0$, $\forall \sigma > 0$

$$\lim_{t\to\infty} y(t) = r$$
$$\lim_{t\to\infty} u(t) = u_r$$

$\lim_{t\to\infty} x(t) = x_r$, $\lim_{t\to\infty} \delta(t) = \delta_r$, $\lim_{t\to\infty} z(t) = z_r$, *and all constraints are fulfilled.*

(Bemporad, Morari 1999)

Proof: Easily follows from standard Lyapunov arguments

More stability results: see (Lazar, Heemels, Weiland, Bemporad, 2005)

Convergence Proof

- Assume we set the terminal constraint $x(t+T|t) = x_r$ in the optimal control problem
- Let $\mathcal{U}_t^*$ denote the optimal control sequence $\{u_t^*(0), \ldots, u_t^*(T-1)\}$
- Let $V(t) \triangleq J(\mathcal{U}_t^*, x(t))$=value function ⟹ Lyapunov function
- By construction, $\mathcal{U}_1 = \{u_t^*(1), \ldots, u_t^*(T-1), u_r\}$ is feasible @$t+1$
- Hence,

$$V(t+1) \leq J(\mathcal{U}_1, x(t+1)) = V(t) - \|y(t) - r\|_Q - \|u(t) - u_r\|_R - \sigma(\|\delta(t) - \delta_r\| - \|z(t) - z_r\| - \|x(t) - x_r\|)$$

- Hence $V(t)$ is decreasing and lower-bounded by 0 $\Rightarrow \exists V_\infty = \lim_{t\to\infty} V(t)$ $\Rightarrow V(t+1) - V(t) \to 0$
- Hence, $\|y(t) - r\|_Q \to 0$, $\|u(t) - u_r\|_R \to 0$, . . . , $\|x(t) - x_r\| \to 0$

Note: Global optimum not needed for convergence !

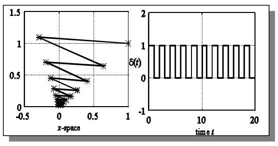
Hybrid MPC - Example

PWA system:

$$x(t+1) = 0.8 \begin{bmatrix} \cos\alpha(t) & -\sin\alpha(t) \\ \sin\alpha(t) & \cos\alpha(t) \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$
$$y(t) = x_2(t)$$
$$\alpha(t) = \begin{cases} \frac{\pi}{3} & \text{if } x_1(t) \geq 0 \\ -\frac{\pi}{3} & \text{if } x_1(t) < 0 \end{cases}$$

Constraint: $-1 \leq u(t) \leq 1$

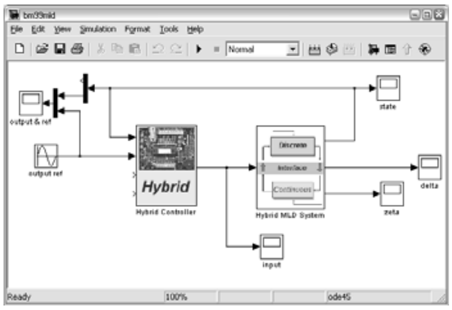
Open loop behavior

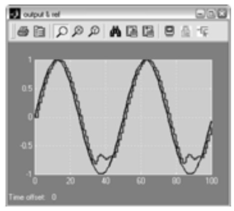


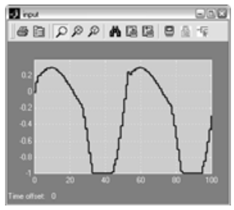
go to demo **/demos/hybrid/bm99sim.m**

Hybrid MPC - Example

Closed loop:







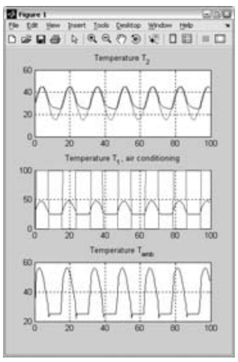
Hybrid MPC – Temperature Control

```
>>refs.x=2;   % just weight state #2
>>Q.x=1;
>>Q.rho=Inf;  % hard constraints
>>Q.norm=2;   % quadratic costs
>>N=2;        % optimization horizon
>>limits.xmin=[25;-Inf];
```

$$\min \sum_{k=1}^{2} (x_2(k) - r)^2$$
s.t. $x_1(k) \geq 25$ $k = 1, 2$
MLD model

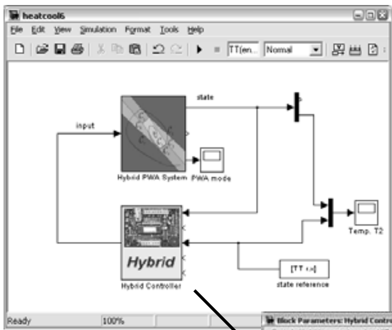
```
>>C=hybcon(S,Q,N,limits,refs);
```

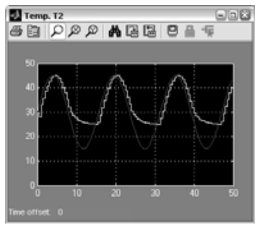
```
>> C

Hybrid controller based on MLD model S <heatcoolmodel.hys>

  2 state measurement(s)
  0 output reference(s)
  0 input reference(s)
  1 state reference(s)
  0 reference(s) on auxiliary continuous z-variables

 20 optimization variable(s) (8 continuous, 12 binary)
 46 mixed-integer linear inequalities
sampling time = 0.5, MILP solver = 'glpk'

Type "struct(C)" for more details.
>>
```

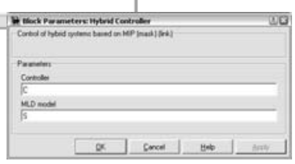



```
>>[XX,UU,DD,ZZ,TT]=sim(C,S,r,x0,Tstop);
```


Hybrid MPC – Temperature Control







Optimal Control of Hybrid Systems: Computational Aspects

MIQP Formulation of MPC

(Bemporad, Morari, 1999)

$$\min_\xi J(\xi, x(0)) = \sum_{t=0}^{T-1} y'(t)Qy(t) + u'(t)Ru(t)$$

$$\text{subject to} \begin{cases} x(t+1) = Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) + B_5 \\ y(t) = Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t) + D_5 \\ E_2\delta(t) + E_3z(t) \leq E_4x(t) + E_1u(t) + E_5 \end{cases}$$

$$\xi = [u(0), \dots, u(T-1), \delta(0), \dots, \delta(T-1), z(0), \dots z(T-1)]'$$

$$\min_\xi \frac{1}{2}\xi' H\xi + x(0)'F\xi + \frac{1}{2}x'(0)Yx(0)$$
$$\text{subj. to} \quad G\xi \leq W + Sx(t)$$

Mixed Integer Quadratic Program (MIQP)

$u \in \mathbb{R}^{n_u}$, $\delta \in \{0,1\}^{n_\delta}$, $z \in \mathbb{R}^{n_z}$ ⇒ $\xi \in \mathbb{R}^{(n_u+n_z)T} \times \{0,1\}^{n_\delta T}$

MILP Formulation of MPC

(Bemporad, Borrelli, Morari, 2000)

$$\min_\xi \; J(\xi, x(0)) = \sum_{t=0}^{T-1} \|Qy(t)\|_\infty + \|Ru(t)\|_\infty$$
subject to MLD model

- Introduce slack variables: $\min |x|$ ⇒ $\min \epsilon$ s.t. $\epsilon \geq x$, $\epsilon \geq -x$

$$\begin{aligned} \epsilon_k^x &\geq \|Qy(t+k|t)\|_\infty \\ \epsilon_k^u &\geq \|Ru(t+k)\|_\infty \end{aligned} \Rightarrow \begin{aligned} \epsilon_k^x &\geq [Qy(t+k|t)]_i \quad i=1,\dots,p,\ k=1,\dots,T-1 \\ \epsilon_k^x &\geq -[Qy(t+k|t)]_i \quad i=1,\dots,p,\ k=1,\dots,T-1 \\ \epsilon_k^u &\geq [Ru(t+k))]_i \quad i=1,\dots,m,\ k=0,\dots,T-1 \\ \epsilon_k^u &\geq -[Ru(t+k))]_i \quad i=1,\dots,m,\ k=0,\dots,T-1 \end{aligned}$$

- Set $\xi \triangleq [\epsilon_1^x, \dots, \epsilon_{N_y}^x, \epsilon_1^u, \dots, \epsilon_{T-1}, U, \delta, z]$

Mixed Integer Linear Program (MILP)

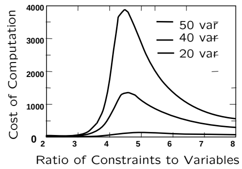
$$\min_\xi J(\xi, x(t)) = \sum_{k=0}^{T-1} \epsilon_i^x + \epsilon_i^u$$
$$\text{s.t.} G\xi \leq W + Sx(t)$$

Mixed-Integer Program Solvers

- Mixed-Integer Programming is *NP*-hard

Phase transitions have been found in computationally hard problems.

BUT



(Monasson et al., *Nature*, 1999)

- General purpose Branch & Bound/Branch & Cut solvers available for MILP and MIQP (CPLEX, Xpress-MP, BARON, GLPK, ...)
  More solvers and benchmarks: **http://plato.la.asu.edu/bench.html**

- No need to reach global optimum (see proof of the theorem), although performance deteriorates

Mixed-Integer Programming Solvers

Main drawbacks:

- Loss of the original discrete structure (Boolean formulas)
- On-line combinatorial optimization

Good for large sampling times (e.g., 1 h) / expensive hardware ...

... but not for fast sampling (e.g. 10 ms) / cheap hardware !

Mixed-Integer Programming Solvers

Main drawbacks:

- **Loss of the original discrete structure (Boolean formulas)**
- On-line combinatorial optimization

Good for large sampling times (e.g., 1 h) / expensive hardware ...

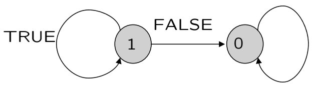
... but not for fast sampling (e.g. 10 ms) / cheap hardware !

Exploiting Logic Structures

Main drawbacks when using Mixed-Integer Programming for implementing hybrid MPC control laws:

1. Loss of the original Boolean structure

$$\begin{cases} x_\ell(k+1) + (1 - x_\ell(k)) + (1 - u_\ell(k)) \geq 1 \\ x_\ell(k) + (1 - x_\ell(k+1)) \geq 1 \\ u_\ell(k) + (1 - x_\ell(k+1)) \geq 1 \end{cases}$$



$x_\ell(k+1) = x_\ell(k) \wedge u_\ell(k)$

Efficiency of MIP solver usually not good when continuous LP/QP relaxations are not tight

"Hybrid" Solvers

Combine MIP and Constraint Satisfaction (CSP) techniques to exploit the discrete structure of the problem

Why CSP ?

- More flexible modeling than MIP (e.g.: constraint logic programming (CLP) and Satisfiability of Boolean formulas (SAT))
- Structure is kept and exploited to direct the search.

Why MIP ?

- Specialized techniques for highly structured problems (e.g. LP problems); Better for handling continuous vars
- A wide range of tight relaxations are available

Why a combined approach ?

Performance increase already shown in other application domains

(Harjunkoski, Jain, Grossmann, 2000)

SAT-Based Branch&Bound

The basic modeling framework has the following form:

(Bockmayr, Kasper,1998)

$\min_z f(z)$ ← Convex cost function

s.t. $Gz \leq d,\ \overline{G}z = \overline{d}$ ← Continuous constraints

$G'z + D'\mu \leq E',\ \overline{G'}z + \overline{D'}\mu = \overline{E'}$ ← Mixed constraints

$g(\nu, \mu)$ ← Purely logic constraints

$z \in \mathcal{R}^{n_z},\ \mu \in \{0,1\}^{n_\mu},\ \nu \in \{0,1\}^{n_\nu}$ ← optimization variables

SAT-based B&B "ingredients"

(Bemporad, Giorgetti, 2003)

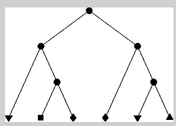
- A relaxed convex problem (pure logic constraints dropped) — Convex solver (e.g.: LP/QP)
- A SAT feasibility problem (for logic constraints only) — SAT solver

Computations: MILP vs. LB-B&B

T	Bool. Vars	SATbB&B (s)	SATbB&B LPs	SATbB&B SATs	Cplex 9.0 (s)	Cplex 9.0 LPs	Pure B&B (s)	Pure B&B LPs
5	82	0.09	5	6	0.03	18	0.48	23
10	157	0.18	5	6	0.13	79	3.7150	119
15	232	0.33	5	6	0.42	199	83.69	943
20	307	0.5110	6	8	0.5410	243	109.0870	2181
25	382	0.7620	8	10	0.8210	286	503.0030	3833
30	457	1.0520	9	12	1.0110	333	1072.3	6227
35	532	1.4420	10	13	1.7170	341	> 1200	–
40	607	1.8630	13	16	2.5030	374	> 1200	–
45	682	2.7740	15	20	3.8320	475	> 1200	–

Computation time and # LP solved for finding an optimal control sequence

SAT-based B&B



#nodes=11

Pure B&B

#nodes=6227

Pentium IV 1.8GHz SAT solver: zCHAFF 2003.07.22

SAT vs. MILP

Uniform Random 3CNF benchmarks (from http://www.satlib.org)
All 3SAT instances are in the phase transition region.

N. Vars	N. Cons	Sat instances zCHAFF	Sat instances CPLEX	Unsat instances zCHAFF	Unsat instances CPLEX
20	91	0	0.036	-	-
50	218	0	0.343	0	0.453
75	325	0	0.203	0	3.671
100	430	0	23.328	0	33.921
125	538	0.016	15.171	0.031	209.766
150	645	0.031	20.625	0.281	4949.58
175	753	0.031	> 1500	0.891	> 5000

All clauses are passed to CPLEX new MILP solver as logic constraints (we exploit the new feature of release 9.0).

PC: P4 2.8GHz + 1GB RAM

Solvers:
CPLEX 9.0
zCHAFF 2003.12.04

Mixed-Integer Programming Solvers

Main drawbacks:

- Loss of the original discrete structure (Boolean formulas)
- **On-line combinatorial optimization**

Good for large sampling times (e.g., 1 h) / expensive hardware ...

... but not for fast sampling (e.g. 10 ms) / cheap hardware !

Outline

✓ Model Predictive Control (MPC) concepts
✓ Hybrid models for MPC
✓ MPC of hybrid systems
- Explicit MPC (multiparametric programming)
- Optimization-based reachability analysis
- Examples

Explicit Model Predictive Control

- Multiparametric quadratic programming
- Explicit linear MPC
- Explicit hybrid MPC
- Matlab tools for explicit MPC

MPC: Pros and Cons

- PRO: systematic design approach
  - Multivariable systems
  - State and Input Constraints
  - Stability guarantees
  - Reference preview
- CON: computation complexity !
  - Large sampling time/fast hardware
  - Software reliability

On-Line vs. Off-Line Optimization

$$\begin{array}{ll}\min_U & \frac{1}{2}U'HU + x'(t)F'U + \frac{1}{2}x'(t)Yx(t) \\ \text{subj. to} & GU \leq W + Sx(t),\end{array}$$

- **On-line** optimization: given $x(t)$ solve the problem at each time step t (the control law $u{=}u(x)$ is implicitly defined by the QP solver)

  ➡ Quadratic Program (QP)

- **Off-line** optimization: solve the QP **for all $x(t)$** to find the control law $u{=}u(x)$ explicitly

  ➡ multi-parametric Quadratic Program (mp-QP)

Multiparametric Quadratic Programming

(Bemporad et al., 2002)

$$\begin{array}{ll}\min_U & \frac{1}{2}U'HU + x'F'U + \frac{1}{2}x'Yx \\ \text{subj. to} & GU \leq W + Sx,\end{array}$$

(the term $\frac{1}{2}x'Yx$ is struck through)

$U \triangleq [u_0' \ \dots \ u_{N-1}']'$

$U \in \mathbb{R}^r$, $r \triangleq mN_u$

$x \in \mathbb{R}^n$

- Objective: solve the QP **for all** $x \in X \subseteq \mathbb{R}^n$
- Assumptions: $\begin{bmatrix} H & F \\ F' & Y \end{bmatrix} \succeq 0$ (always satisfied if QP problem originates from optimal control problem)

  $H \succ 0$ (can be easily satisfied, e.g. by choosing positive input weights)

Linearity of the Solution

$x_0 \in X$ ⇨ solve QP to find $U^*(x_0)$, $\lambda^*(x_0)$

⇨ identify active constraints at $U^*(x_0)$

⇨ form matrices $\tilde{G}$, $\tilde{W}$, $\tilde{S}$ by collecting active constraints: $\tilde{G}U^*(x_0) - \tilde{W} - \tilde{S}x_0 = 0$

KKT optimality conditions:

(1) $HU + Fx + G'\lambda = 0$, (2) $\tilde{G}U - \tilde{W} - \tilde{S}x = 0$
(3) $\lambda_i(G^iU - W^i - S^ix) = 0$, (4) $\hat{G}U \leq \hat{W} + \hat{S}x$
(5) $\tilde{\lambda}_i \geq 0$, $\hat{\lambda}_i = 0$

$\hat{G}$=rows of G not in $\tilde{G}$ (inactive constraints)

From (1) : $U = -H^{-1}(Fx + \tilde{G}'\tilde{\lambda})$

From (2) :

$$\tilde{\lambda}(x) = -(\tilde{G}H^{-1}\tilde{G}')^{-1}(\tilde{W} + (\tilde{S} + \tilde{G}H^{-1}F)x).$$
$$U(x) = H^{-1}[\tilde{G}'(\tilde{G}H^{-1}\tilde{G}')^{-1}(\tilde{W}+(\tilde{S}+\tilde{G}H^{-1}F)x)-Fx]$$

➡ In some neighborhood of x_0, λ and U are explicit affine functions of x !

Determining a Critical Region

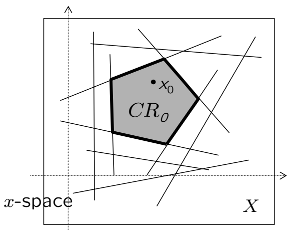
- Impose primal and dual feasibility: $\hat{G}U(x) \leq \hat{W} + \hat{S}x$, $\tilde{\lambda}(x) \geq 0$

  ➡ linear inequalities in x !

- Remove redundant constraints: (this requires solving LP's)

  ➡ critical region CR_0

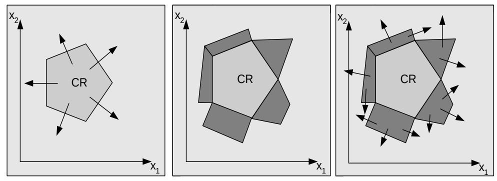
$$CR_0 = \{x \in X : \ \mathcal{A}x \leq \mathcal{B}\}$$



- CR_0 is the set of all and only parameters x for which $\tilde{G}, \tilde{W}, \tilde{S}$ is the optimal combination of active constraints at the optimizer

Neighboring Regions

(Tøndel, Johansen, Bemporad, 2003)



The active set of a neighboring region is found by using the active set of the current region + knowledge of the type of hyperplane we are crossing:

$\hat{G}^iU(x) \leq \hat{W}^i + \hat{S}^ix$ ⇒ The corresponding constraint is **added** to the active set

$\tilde{\lambda}_j(x) \geq 0$ ⇒ The corresponding constraint is **withdrawn** from the active set

Mp-QP Properties

Theorem 1 *Consider a multi-parametric quadratic program with* $H \succ 0$, $\begin{bmatrix} H & F \\ F' & Y \end{bmatrix} \succeq 0$. *The set* X^* *of parameters* x *for which the problem is feasible is a polyhedral set, the value function* $J^* : X^* \mapsto \mathbb{R}$ *is piecewise quadratic, convex and continuous and the optimizer* $U^* : X^* \mapsto \mathbb{R}^r$ *is piecewise affine and continuous.*

$$U^*(x) = \arg\min_U \ \tfrac{1}{2}U'HU + x'F'U \quad \text{subj. to} \quad GU \leq W + Sx$$

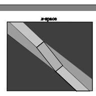
continuous, piecewise affine

$$V^*(x) = \tfrac{1}{2}x'Yx + \min_U \ \tfrac{1}{2}U'HU + x'F'U \quad \text{subj. to} \quad GU \leq W + Sx$$

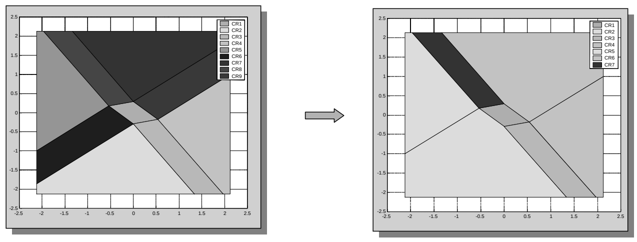
convex, continuous, piecewise quadratic, C^1 (if no degeneracy)

Corollary: The linear MPC controller is a continuous piecewise affine function of the state

$$u(x) = \begin{cases} F_1x + G_1 & \text{if } H_1x \leq K_1 \\ \vdots & \vdots \\ F_Nx + G_N & \text{if } H_Nx \leq K_N \end{cases}$$



Complexity Reduction



$$U(x) \triangleq [u'_0(x) \ u'_1(x) \ \ldots \ u'_{N-1}(x)]'$$

Regions where the first component of the solution is the same can be joined (when their union is convex).

(Bemporad, Fukuda, Torrisi, *Computational Geometry*, 2001)

Double Integrator Example

- System: $y(t) = \dfrac{1}{s^2}u(t)$ ➡ sampling + ZOH T_s=1 s

$$x(t+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$
$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t)$$

- Constraints: $-1 \leq u(t) \leq 1$
- Control objective: minimize $\sum_{t=0}^{\infty} y^2(t) + \frac{1}{100}u^2(t)$

$$u(t+k) = K_{LQ}x(t+k|t), \ \forall k \geq N_u$$

- Optimization problem: for N_u=2

$$H = \begin{bmatrix} 0.8365 & 0.3603 \\ 0.3603 & 0.2059 \end{bmatrix}, \ F = \begin{bmatrix} 0.4624 & 1.2852 \\ 0.1682 & 0.5285 \end{bmatrix}$$

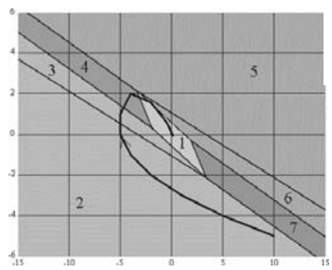
(cost function is normalized by max svd(H))

$$G = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, \ W = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \ S = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

mp-QP solution

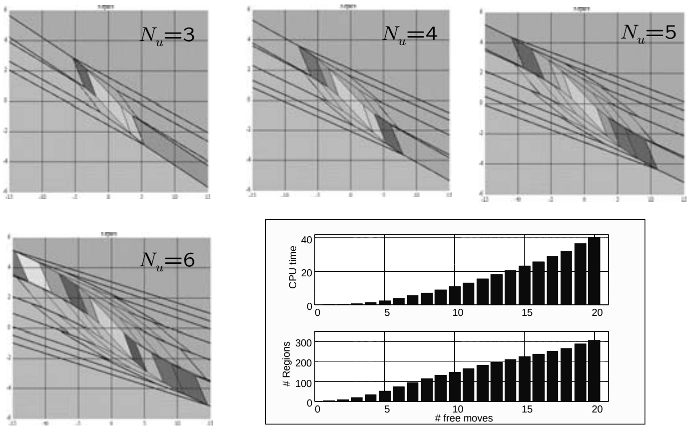
N_u=2

$$u(x) = \begin{cases} \begin{bmatrix} -0.8166 & -1.7499 \end{bmatrix} x & \text{if } \begin{bmatrix} -0.8166 & -1.7499 \\ 0.8166 & 1.7499 \\ 0.6124 & 0.4957 \\ -0.6124 & -0.4957 \end{bmatrix} x \leq \begin{bmatrix} 1.0000 \\ 1.0000 \\ 1.0000 \\ 1.0000 \end{bmatrix} & \text{(Region \#1)} \\ 1.0000 & \text{if } \begin{bmatrix} 0.3864 & 1.0738 \\ 0.2970 & 0.9333 \end{bmatrix} x \leq \begin{bmatrix} -1.0000 \\ -1.0000 \end{bmatrix} & \text{(Region \#2)} \\ 1.0000 & \text{if } \begin{bmatrix} 0.9712 & 2.6991 \\ -0.2970 & -0.9333 \\ 0.8166 & 1.7499 \end{bmatrix} x \leq \begin{bmatrix} -1.0000 \\ 1.0000 \\ -1.0000 \end{bmatrix} & \text{(Region \#3)} \\ \begin{bmatrix} -0.5528 & -1.5364 \end{bmatrix} x + 0.4308 & \text{if } \begin{bmatrix} -0.9712 & -2.6991 \\ 0.3864 & 1.0738 \\ 0.6124 & 0.4957 \end{bmatrix} x \leq \begin{bmatrix} 1.0000 \\ 1.0000 \\ -1.0000 \end{bmatrix} & \text{(Region \#4)} \\ -1.0000 & \text{if } \begin{bmatrix} -0.3864 & -1.0738 \\ -0.2970 & -0.9333 \end{bmatrix} x \leq \begin{bmatrix} -1.0000 \\ -1.0000 \end{bmatrix} & \text{(Region \#5)} \\ -1.0000 & \text{if } \begin{bmatrix} -0.9712 & -2.6991 \\ 0.2970 & 0.9333 \\ -0.8166 & -1.7499 \end{bmatrix} x \leq \begin{bmatrix} -1.0000 \\ 1.0000 \\ -1.0000 \end{bmatrix} & \text{(Region \#6)} \\ \begin{bmatrix} -0.5528 & -1.5364 \end{bmatrix} x - 0.4308 & \text{if } \begin{bmatrix} -0.3864 & -1.0738 \\ 0.9712 & 2.6991 \\ -0.6124 & -0.4957 \end{bmatrix} x \leq \begin{bmatrix} 1.0000 \\ 1.0000 \\ -1.0000 \end{bmatrix} & \text{(Region \#7)} \end{cases}$$



go to demo **/demos/linear/doubleintexp.m** (Hyb-Tbx)

Complexity



(is the number of regions finite for $N_u \to \infty$?)

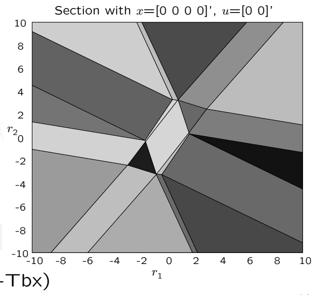
Example: AFTI-16

- Linearized model:

$$\begin{cases} \dot{x} = \begin{bmatrix} -.0151 & -60.5651 & 0 & -32.174 \\ -.0001 & -1.3411 & .9929 & 0 \\ .00018 & 43.2541 & -.86939 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} -2.516 & -13.136 \\ -.1689 & -.2514 \\ -17.251 & -1.5766 \\ 0 & 0 \end{bmatrix} u \\ y = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x, \end{cases}$$

- Inputs: elevator and flaperon angle
- Outputs: attack and pitch angle
- Sampling time: $T_s = .05$ s (+ zero-order hold)
- Constraints: max 25° on both angles
- Open-loop response: unstable
  (open-loop poles: $-7.6636, -0.0075 \pm 0.0556j, 5.4530$)

Explicit controller: 8 parameters, 51 regions



go to demo **linear/afti16.m** (Hyb-Tbx)

Explicit Hybrid MPC (MLD)

$$\min_\xi J(\xi, x(t)) = \sum_{k=0}^{T-1} \|Q(y_k - r(t))\|_\infty + \|Ru_k\|_\infty$$

$$\text{subject to} \begin{cases} x_{k+1} = Ax_k + B_1u_k + B_2\delta_k + B_3z_k + B_5 \\ y_k = Cx_k + D_1u_k + D_2\delta_k + D_3z_k + D_5 \\ E_2\delta_k + E_3z_k \leq E_4x_k + E_1u_k + E_5 \\ x_0 = x(t) \end{cases}$$

- On-line optimization: solve the problem *for each* given $x(t)$

  Mixed-Integer Linear Program (MILP)

- Off-line optimization: solve the MILP **for all** $x(t)$ in advance

$$\min_\xi \sum_{k=0}^{T-1} \epsilon_i^x + \epsilon_i^u \quad \text{s.t. } G\xi \leq W + S \begin{bmatrix} x(t) \\ r(t) \end{bmatrix}$$

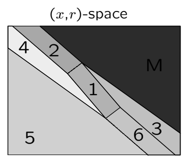
multi-parametric Mixed Integer Linear Program (mp-MILP)

Multiparametric MILP

$$\min_{\xi=\{\xi_c,\xi_d\}} f'\xi_c + d'\xi_d \quad \text{s.t. } G\xi_c + E\xi_d \leq W + Fx \qquad \xi_c \in \mathbf{R}^n \quad \xi_d \in \{0,1\}^m$$

- mp-MILP can be solved (by alternating MILPs and mp-LPs)
  (Dua, Pistikopoulos, 1999)
- **Theorem:** The multiparametric solution $\xi^*(x)$ is piecewise affine
- **The MPC controller is piecewise affine in x,r**

$$u(x,r) = \begin{cases} F_1x + E_1r + g_1 & \text{if } H_1 \left[\begin{smallmatrix} x \\ r \end{smallmatrix}\right] \leq K_1 \\ \vdots & \vdots \\ F_Mx + E_Mr + g_M & \text{if } H_M \left[\begin{smallmatrix} x \\ r \end{smallmatrix}\right] \leq K_M \end{cases}$$



Explicit Hybrid MPC (PWA)

$$\min_U J(U,x,r) = \sum_{k=0}^{T-1} \|R(y(k) - r)\|_p + \|Qu(k)\|_p \quad \text{subject to} \begin{cases} \text{PWA model} \\ x(0) = x \end{cases}$$

$p = 1, 2, \infty$

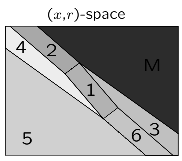
$\|v\|_2 = v'v$

$\|v\|_\infty = \max |v_i|$

$\|v\|_1 = \sum v_i$

- Solution $u(x,r)$ found via a combination of
  - Dynamic programming or enumeration of feasible mode sequences, multiparametric linear or quadratic programming, and polyhedral computation. (Borrelli, Baotic, Bemporad, Morari, 2003) (Mayne, ECC 2001)
- **The MPC controller is piecewise affine in x,r**

$$u(x,r) = \begin{cases} F_1x + E_1r + g_1 & \text{if } H_1 \left[\begin{smallmatrix} x \\ r \end{smallmatrix}\right] \leq K_1 \\ \vdots & \vdots \\ F_Mx + E_Mr + g_M & \text{if } H_M \left[\begin{smallmatrix} x \\ r \end{smallmatrix}\right] \leq K_M \end{cases}$$



<u>Note</u>: in the 2-norm case the partition may not be fully polyhedral

Explicit Hybrid MPC (PWA)

<u>Method A</u>: (Borrelli, Baotic, Bemporad, Morari, *Automatica*, 2005)

Use a combination of DP (dynamic programming) and mpLP (1-norm, ∞-norm), or mpQP (quadratic forms)

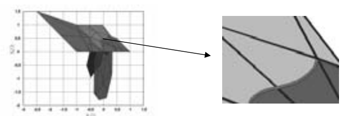
<u>Method B</u>: (Bemporad, *Hybrid Toolbox*, 2003) (Alessio, Bemporad, 2005) (Mayne, ECC 2001)

1 - Use backwards (=DP) reachability analysis for enumerating all feasible mode sequences $I=\{i(0),i(1),\ldots,i(T\text{-}1)\}$;

2 - For each fixed sequence I, solve the explicit finite-time optimal control problem for the corresponding linear time-varying system (mpQP or mpLP);

3 - Case $1/\infty$-norm: Compare value functions and split regions. Quadratic case: keep overlapping regions (possibly eliminate overlaps that are never optimal) and compare on-line (if needed).

<u>Note</u>: in the 2-norm case, the fully explicit partition may not be polyhedral



Hybrid Control Example (Revisited)

Hybrid Control - Example

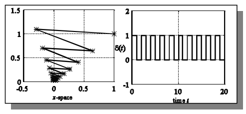
PWA system:

$$
\begin{aligned}
x(t+1) &= 0.8\begin{bmatrix}\cos\alpha(t) & -\sin\alpha(t)\\ \sin\alpha(t) & \cos\alpha(t)\end{bmatrix}x(t)+\begin{bmatrix}0\\1\end{bmatrix}u(t)\\
y(t) &= x_2(t)\\
\alpha(t) &= \begin{cases}\frac{\pi}{3} & \text{if } x_1(t)\geq 0\\ -\frac{\pi}{3} & \text{if } x_1(t)<0\end{cases}
\end{aligned}
$$

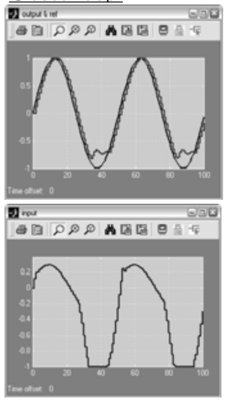
Constraints: $-1 \leq u(t) \leq 1$

Objective: $\min \sum_{k=1}^{2} |y(t+k|t)-r(t)|$

Open loop behavior:



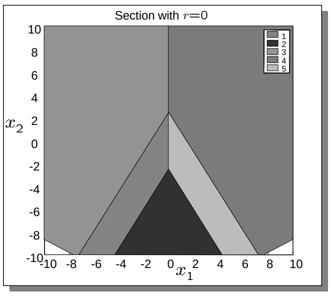
Closed loop:



HybTbx: **/demos/hybrid/bm99sim.m**

Explicit PWA Controller

$$
u(x,r)=\begin{cases}
\begin{bmatrix}0.6928 & -0.4 & 1\end{bmatrix}\begin{bmatrix}x\\r\end{bmatrix} & \text{if } \begin{bmatrix}0.6928 & -0.4 & 1\\ -0.4 & -0.6928 & 0\\ 0 & -1 & 0\\ 0 & 0 & -1\\ -0.6928 & 0.4 & -1\\ 0 & 0 & 1\\ 1 & 0 & 0\end{bmatrix}\begin{bmatrix}x\\r\end{bmatrix}\leq\begin{bmatrix}1\\10\\10\\1\\1\\1\\1e{-}006\end{bmatrix} \text{ (Region \#1)}\\
1 & \text{if } \begin{bmatrix}-0.6928 & 0.4 & -1\\ 0.6928 & 0.4 & -1\\ 0 & 0 & 1\\ 0 & 0 & -1\\ 0 & -1 & 0\end{bmatrix}\begin{bmatrix}x\\r\end{bmatrix}\leq\begin{bmatrix}-1\\-1\\1\\1\\10\end{bmatrix} \text{ (Region \#2)}\\
-1 & \text{if } \begin{bmatrix}-0.4 & -0.6928 & 0\\ 0 & -1 & 0\\ 1 & 0 & 0\\ 0 & 1 & 0\\ 0.6928 & -0.4 & 1\\ 0 & 0 & 1\\ 0 & 0 & -1\\ -1 & 0 & 0\end{bmatrix}\begin{bmatrix}x\\r\end{bmatrix}\leq\begin{bmatrix}10\\10\\1e{-}006\\10\\-1\\1\\1\\10\end{bmatrix} \text{ (Region \#3)}\\
-1 & \text{if } \begin{bmatrix}-1 & 0 & 0\\ 0 & 1 & 0\\ 0.4 & -0.6928 & 0\\ 0 & -1 & 0\\ -0.6928 & -0.4 & 1\\ 0 & 0 & 1\\ 0 & 0 & -1\\ 1 & 0 & 0\end{bmatrix}\begin{bmatrix}x\\r\end{bmatrix}\leq\begin{bmatrix}0\\10\\10\\10\\-1\\1\\1\\10\end{bmatrix} \text{ (Region \#4)}\\
\begin{bmatrix}-0.6928 & -0.4 & 1\end{bmatrix}\begin{bmatrix}x\\r\end{bmatrix} & \text{if } \begin{bmatrix}-0.6928 & -0.4 & 1\\ 0.4 & -0.6928 & 0\\ 0 & -1 & 0\\ 0 & 0 & -1\\ 0.6928 & 0.4 & -1\\ 0 & 0 & 1\\ -1 & 0 & 0\end{bmatrix}\begin{bmatrix}x\\r\end{bmatrix}\leq\begin{bmatrix}1\\10\\10\\1\\1\\1\\0\end{bmatrix} \text{ (Region \#5)}
\end{cases}
$$



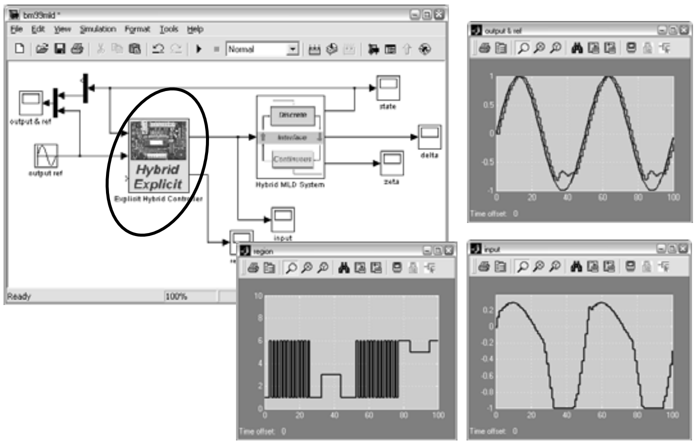
PWA law $\equiv$ MPC law

HybTbx: **/demos/hybrid/bm99sim.m**

(CPU time: 1.51 s, Pentium M 1.4GHz)

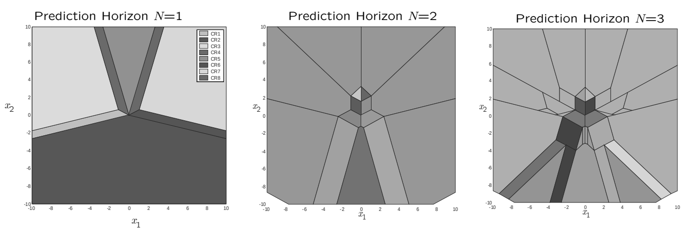
Hybrid MPC - Example

Closed loop:



Explicit PWA Regulator

Objective: $\min \sum_{k=1}^{N} \|x(t+k|t)\|_\infty$



HybTbx: **/demos/hybrid/bm99benchmark.m**

Explicit MPC – Temperature Control

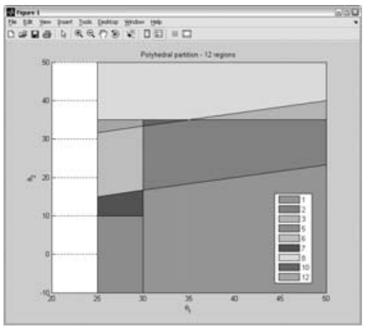
```
>>E=expcon(C,range,options);
```

```
>> E

Explicit controller (based on hybrid controller C)
  3 parameter(s)
  1 input(s)
 12 partition(s)
sampling time = 0.5

The controller is for hybrid systems (tracking)
This is a state-feedback controller.

Type "struct(E)" for more details.
>>
```

$\min \sum_{k=1}^{2} x_2^2(k)$

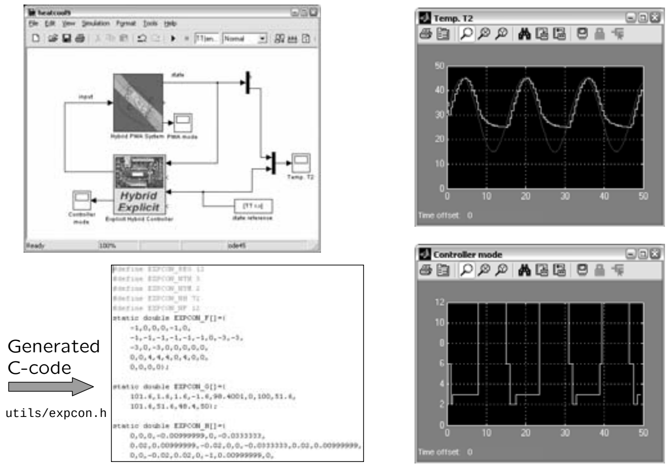
s.t. $x_1(k) \geq 25$ $k=1,2$

PWA model



Section in the (T_1,T_2)-space for $T_{ref} = 30$

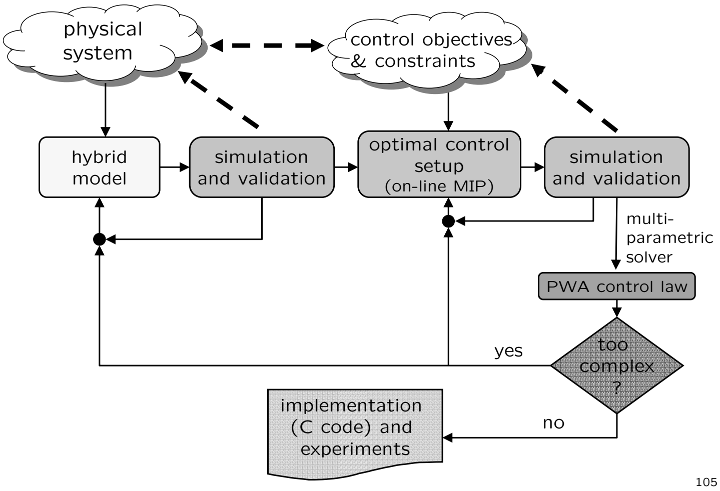
Explicit MPC – Temperature Control



Comments on Explicit Solutions

- Alternatives: either (1) solve an MIP on-line
  or (2) evaluate a PWA function
- For problems with many variables and/or long horizons: MIP may be preferable
- For simple problems (short horizon/few constraints):
  - time to evaluate the control law is shorter than MIP
  - control code is simpler (no complex solver must be included in the control software !)
  - more insight in controller's behavior

Hybrid Control Design Flow

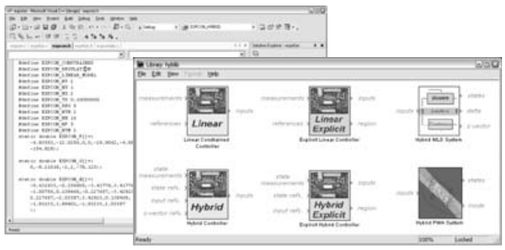


Hybrid Toolbox for Matlab

(Bemporad, 2003-2004)

Features:

- Hybrid model (MLD and PWA) design and simulation
- Control design for linear systems w/ constraints and hybrid systems (on-line optimization via QP/MILP/MIQP)
- Explicit control (via multiparametric programming)
- C-code generation
- Simulink

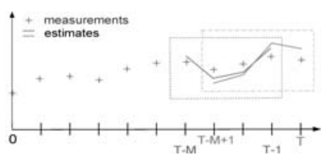


http://www.dii.unisi.it/hybrid/toolbox

State Estimation / Fault Detection

State Estimation / Fault Detection

- Problem: given past output measurements and inputs, estimate the current state/faults
- Solution: Use Moving Horizon Estimation for MLD systems (dual of MPC)



Augment the MLD model with:

- Input disturbances $\xi \in \mathbf{R}^n$
- Output disturbances $\zeta \in \mathbf{R}^p$

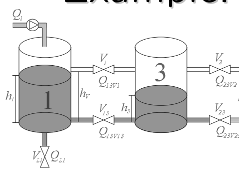
At each time t solve the optimization: $\min \sum_{k=1}^{T} \|\hat{y}(t-k|t) - y(t-k)\|^2 + \ldots$ and get estimates $\hat{x}(t)$

MHE optimization = MIQP (Bemporad, Mignone, Morari, ACC 1999)

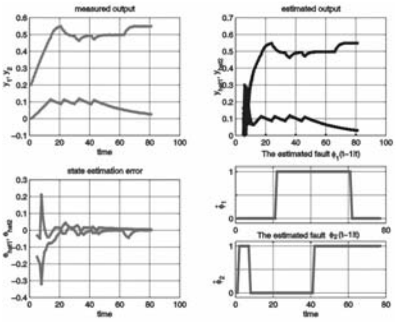
Convergence can be guaranteed (Ferrari-T., Mignone, Morari, 2002)

Fault detection: augment MLD with unknown **binary** distubances $\phi \in \{0,1\}^p$

Example: Three Tank System

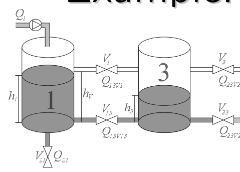


COSY Benchmark problem, ESF

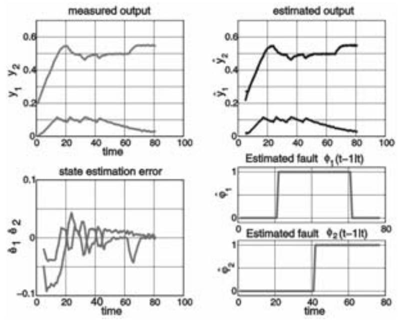


- ϕ_1 : leak in tank 1 for $20s \le t \le 60s$
- ϕ_2 : valve V_1 blocked for $t \ge 40s$

Example: Three Tank System



COSY Benchmark problem, ESF



- ϕ_1 : leak in tank 1 for $20s \le t \le 60s$
- ϕ_2 : valve V_1 blocked for $t \ge 40s$
- Add logic constraint $[h_1 \le h_v] \Rightarrow \phi_2 = 0$

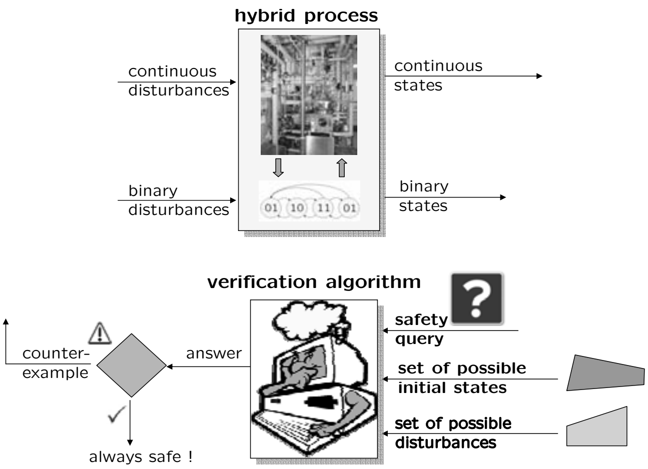
Outline

- ✓ Model Predictive Control (MPC) concepts
- ✓ Hybrid models for MPC
- ✓ MPC of hybrid systems
- ✓ Explicit MPC (multiparametric programming)
- Optimization-based reachability analysis
- Examples

Optimization-based Reachability Analysis

(Verification of Safety Properties via Mixed-integer Programming)

Hybrid Verification Problem



Verification Algorithm

- **QUERY**: Is the target set X_f reachable after N steps from some initial state $x_0 \in X_0$ for some input profile $u \in U$?
- Computation: Solve the mixed-integer linear program (MILP)

$$\min \ 0 \quad \text{s.t.} \left\{ \begin{array}{rcl} x(k+1) & = & Ax(k) + B_1 u(k) + B_2 \delta(k) + B_3 z(k) \\ E_2 \delta(k) & + & E_3 z(k) \le E_1 u(k) + E_4 x(k) + E_5 \\ S_u u(k) & \le & T_u \quad (u(k) \in U) \\ & & k = 0, 1, \ldots, N-1 \\ S_0 x(0) & \le & T_0 \quad (x(0) \in X_0) \\ S_f x(N) & \le & T_f \quad (x(N) \in X_f) \end{array} \right.$$

with respect to u(0),δ(0),z(0),..., u(N-1),δ(N-1),z(N-1),x(0)

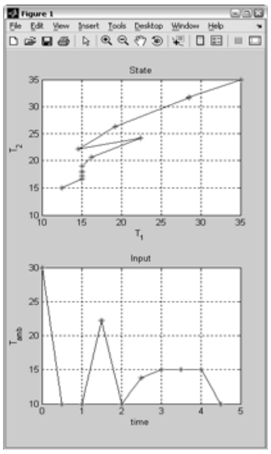
- **Alternative solutions**:
  - Exploit the special structure of the problem and use polyhedral computation. (Torrisi, 2003)
  - Use abstractions (LPs) + SAT solvers (Giorgetti, Pappas, 2005)

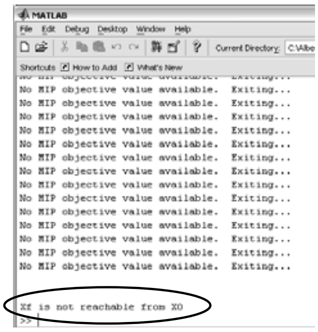
Verification Example

- MLD model: room temperature system
- $X_f = \{\left[\begin{smallmatrix} T_1 \\ T_2 \end{smallmatrix}\right] : 10 \leq T_1, T_2 \leq 15\}$ (set of unsafe states)
- $X_0 = \{\left[\begin{smallmatrix} T_1 \\ T_2 \end{smallmatrix}\right] : 35 \leq T_1, T_2 \leq 40\}$ (set of initial states)
- $U = \{T_{\text{amb}} : 10 \leq T_{\text{amb}} \leq 30\}$ (set of possible inputs)
- N=10 (time horizon)

```
>>[flag,x0,U]=reach(S,N,Xf,X0,umin,umax);
```


Verification Example





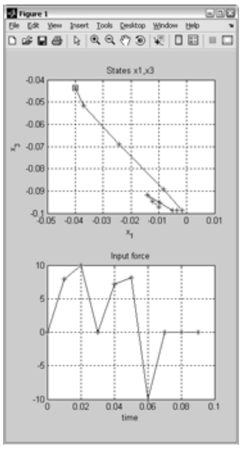
$U = \{T_{\text{amb}} : 10 \leq T_{\text{amb}} \leq 30\}$ $U = \{T_{\text{amb}} : 20 \leq T_{\text{amb}} \leq 30\}$

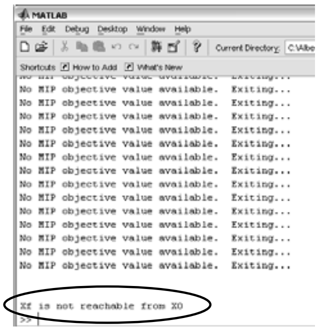
Verification Example

- MLD model: semiactive suspension system
- $X_f = \{\left[\begin{smallmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{smallmatrix}\right] : -0.05 \leq x_1 \leq -0.04\}$ (set of unsafe states)
- $X_0 = \{\left[\begin{smallmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{smallmatrix}\right] : -0.01 \leq x_1, x_2, x_4 \leq 0.01, \; -0.01 \leq x_3 \leq 0.01\}$ (set of initial states)
- $U = \{\bar{f} : -10 \leq \bar{f} \leq 10\}$ (set of possible inputs)
- N=10 (time horizon)

```
>>[flag,x0,U]=reach(S,N,Xf,X0,umin,umax);
```


Verification Example



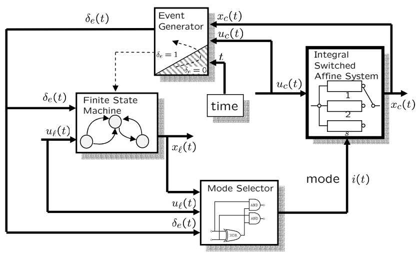


$U = \{\bar{f} : -10 \leq \bar{f} \leq 10\}$ $U = \{\bar{f} : -10 \leq \bar{f} \leq 5\}$

Event-Driven / Continuous-Time Hybrid Automata

Integral Switched Affine System

(Bemporad, Di Cairano, Julvez, 2005)



Integral dynamics:

$$\frac{dx_c(t)}{dt} = B_{i(t)} u_c(t) + f_{i(t)}$$

$i(t) \equiv i(t_k) \quad \forall t_k \leq t < t_k + 1$

$u_c(t) \equiv u(t_k) \quad \forall t_k \leq t < t_k + 1$

k = event index

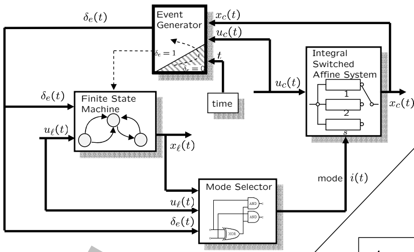
t_k = event instant

$$\begin{aligned} x_c(t_{k+1}) &= x_c(t_k) + B_{i(t_k)} q(t_k) u_c(t_k) + f_{i(t_k)} q(t_k) \\ t_{k+1} &= t_k + q(t_k) \end{aligned}$$

Change of variables: $v(t_k) = q(t_k) u_c(t_k)$

Same as discrete-time dynamics !

Event Generator

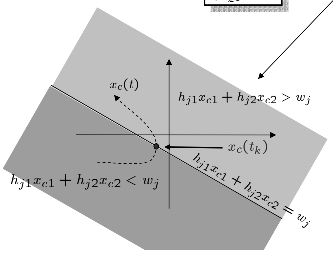


Events occur at time t if:

(1) $\delta_e(t) \neq \delta_e(t_{k-1})$ or

(2) $u_c(t) \neq u_c(t_{k-1})$ or

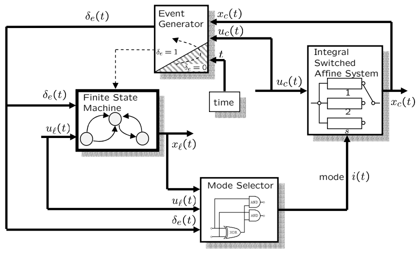
(3) $u_\ell(t) \neq u_\ell(t_{k-1})$



$$t_k = \inf \; t \quad \text{s.t. } t > t_{k-1}, \; (1) \text{ or } (2) \text{ or } (3)$$

Example:
sign($H^j x_c(t) - W^j$) $\neq$
sign($H^j x_c(t_{k-1}) - W^j$)
for some threshold index j

Asynchronous Finite State Machine



Asynchronous FSM dynamics: $x_\ell(t_{k+1}) = f_{\text{aFSM}}(x_\ell(t_k), u_\ell(t_k), \delta_e(t_k))$

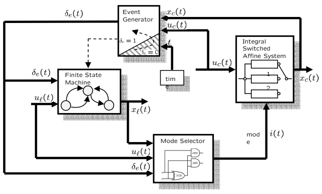
$x_\ell(t) \equiv x_\ell(t_{k+1}),\ \forall t_k < t \leq t_{k+1}$

Discrete state transitions, events, and mode switches might occur at **any (continuous) time** t !

Hybrid Automaton with Integral Continuous(-time) Dynamics (icHA)

Wrapping up:

event-driven (asynchronous) FSM
+
integral dynamics
=
same as discrete hybrid automaton



$$\begin{aligned} x_c(t_{k+1}) &= x_c(t_k) + B_{i(t_k)} v(t_k) + f_{i(t_k)} q(t_k) \\ t_{k+1} &= t_k + q(t_k) \\ x_\ell(t_{k+1}) &= f_{\text{aFSM}}(x_\ell(t_k), u_\ell(t_k), \delta_e(t_k)) \end{aligned}$$

$u_c(t_k) = v(t_k)/q(t_k)$

states inputs

All numerical control/verification techniques developed for DHA can be immediately applied to icHA !

Outline

- ✓ Model Predictive Control (MPC) concepts
- ✓ Hybrid models for MPC
- ✓ MPC of hybrid systems
- ✓ Explicit MPC (multiparametric programming)
- ✓ Optimization-based reachability analysis
- Examples

Hybrid Control Example: Cruise Control System

Hybrid Control Problem

GOAL:

command gear ratio, gas pedal, and brakes to **track** a desired speed and minimize consumption

Hybrid Model

- Vehicle dynamics

$$\boxed{m\ddot{x} = F_e - F_b - \beta\dot{x}}$$

$\dot{x}$ = vehicle speed
F_e = traction force
F_b = brake force

⟹ discretized with sampling time $T_s = 0.5$ s

- Transmission kinematics

$$\boxed{\omega = \frac{R_g(i)}{k_s}\dot{x}}$$

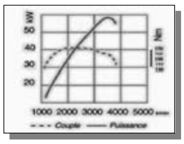
$$\boxed{F_e = \frac{R_g(i)}{k_s}M}$$

ω = engine speed
M = engine torque
i = gear

127

Hybrid Model

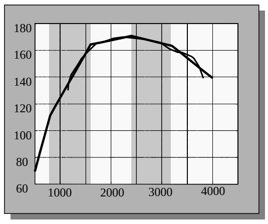
- Engine torque $\boxed{-C_e^-(\omega) \leq M \leq C_e^+(\omega)}$
- Max engine torque $\boxed{C_e^+(\omega)}$



Piecewise-linearization:
(PWL Toolbox, Julián, 1999)

requires: 4 binary aux variables
4 continuous aux variables

⟹



- Min engine torque $\boxed{C_e^-(\omega) = \alpha_1 + \beta_1\omega}$

128

Hybrid Model

- Gear selection: for each gear #i,

define a binary input $\boxed{g_i \in \{0,1\}}$

- Gear selection (traction force):

$\boxed{F_e = \frac{R_g(i)}{k_s}M}$ depends on gear #i

define auxiliary continuous variables:

$$\boxed{\text{IF } g_i = 1 \text{ THEN } F_{ei} = \frac{R_g(i)}{k_s}M \text{ ELSE } 0}$$

$$\Longrightarrow \boxed{F_e = F_{eR} + F_{e1} + F_{e2} + F_{e3} + F_{e4} + F_{e5}}$$

- Gear selection (engine/vehicle speed):

$\boxed{\omega = \frac{R_g(i)}{k_s}\dot{x}}$ similarly, also requires 6 auxiliary continuous variables

129

Hysdel Model

```
SYSTEM car {

INTERFACE {
        STATE { REAL position, speed; }
        INPUT { REAL torque, F_brake;
                BOOL gear1, gear2, gear3, gear4, gear5, gearR;  }

        PARAMETER{
                REAL mass = 1020; /* kg */
                REAL beta_friction = 25; /* W/m*s */
                REAL Rgear1 = 3.7271; REAL Rgear2 = 2.048;
                REAL Rgear3 = 1.321; REAL Rgear4 = 0.971;
                REAL Rgear5 = 0.756; REAL RgearR = -3.545;
                REAL wheel_rim = 14;        /* in */
                ...

                }  }
IMPLEMENTATION {
        AUX {REAL Fe1, Fe2, Fe3, Fe4, Fe5, FeR;
             REAL w1, w2, w3, w4, w5, wR;
             BOOL dPWL1,dPWL2,dPWL3,dPWL4;
             REAL DCe1,DCe2,DCe3,DCe4; }

        AD { dPWL1 = wPWL1-(w1+w2+w3+w4+w5+wR)<=0;
             dPWL2 = wPWL2-(w1+w2+w3+w4+w5+wR)<=0;
             dPWL3 = wPWL3-(w1+w2+w3+w4+w5+wR)<=0;
             dPWL4 = wPWL4-(w1+w2+w3+w4+w5+wR)<=0; }

        DA { Fe1 = {IF gear1  THEN torque/speed_factor*Rgear1};
             Fe2 = {IF gear2  THEN torque/speed_factor*Rgear2};
             Fe3 = {IF gear3  THEN torque/speed_factor*Rgear3};
             Fe4 = {IF gear4  THEN torque/speed_factor*Rgear4};
             Fe5 = {IF gear5  THEN torque/speed_factor*Rgear5};
             FeR = {IF gearR  THEN torque/speed_factor*RgearR};

             w1 = {IF gear1  THEN speed/speed_factor*Rgear1};
             w2 = {IF gear2  THEN speed/speed_factor*Rgear2};
             w3 = {IF gear3  THEN speed/speed_factor*Rgear3};
             w4 = {IF gear4  THEN speed/speed_factor*Rgear4};
             w5 = {IF gear5  THEN speed/speed_factor*Rgear5};
             wR = {IF gearR  THEN speed/speed_factor*RgearR};

             DCe1 = {IF dPWL1 THEN (aPWL2-aPWL1)+(bPWL2-bPWL1)*(w1+w2+w3+w4+w5+wR)};
             DCe2 = {IF dPWL2 THEN (aPWL3-aPWL2)+(bPWL3-bPWL2)*(w1+w2+w3+w4+w5+wR)};
             DCe3 = {IF dPWL3 THEN (aPWL4-aPWL3)+(bPWL4-bPWL3)*(w1+w2+w3+w4+w5+wR)};
             DCe4 = {IF dPWL4 THEN (aPWL5-aPWL4)+(bPWL5-bPWL4)*(w1+w2+w3+w4+w5+wR)};
             }

        CONTINUOUS { position = position+Ts*speed;
                     speed = speed+Ts/mass*(Fe1+Fe2+Fe3+Fe4+Fe5+FeR-
                                     F_brake-beta_friction*speed); }

        MUST  {  wemin <= w1+w2+w3+w4+w5+wR;
                 w1+w2+w3+w4+w5+wR <= wemax;
                 -F_brake <=0; /* brakes cannot accelerate ! */
                 F_brake <= max_brake_force;

                 -torque-(alpha1+beta1*(w1+w2+w3+w4+w5+wR)) <=0;
                 torque-(aPWL1+bPWL1*(w1+w2+w3+w4+w5+wR)+DCe1+DCe2+DCe3+DCe4)-1<=0;

                 -(gear1+gear2+gear3+gear4+gear5+gearR)<=-1;
                 (gear1+gear2+gear3+gear4+gear5+gearR)<=1;
                 Fe1+Fe2+Fe3+Fe4+Fe5+FeR <= max_force;
                 -Fe1-Fe2-Fe3-Fe4-Fe5-FeR <= -max_force;

                 dPWL4 -> dPWL3; dPWL4 -> dPWL2;
                 dPWL4 -> dPWL1; dPWL3 -> dPWL2;
                 dPWL3 -> dPWL1; dPWL2 -> dPWL1;  }
```

go to demo **/demos/cruise/init.m**

130

Hybrid Model

- MLD model

$$\boxed{\begin{aligned} x(t+1) &= Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) \\ y(t) &= Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t) \\ E_2\delta(t) + E_3z(t) &\leq E_4x(t) + E_1u(t) + E_5 \end{aligned}}$$

- 2 continuous states: x, v (vehicle position and speed)
- 2 continuous inputs: M, F_b (engine torque, brake force)
- 6 binary inputs: $g_R, g_1, g_2, g_3, g_4, g_5$ (gears)
- 1 continuous output: v (vehicle speed)
- 16 auxiliary continuous vars: (6 traction force, 6 engine speed, 4 PWL max engine torque)
- 4 auxiliary binary vars: (PWL max engine torque breakpoints)
- 96 mixed-integer inequalities

131

Hybrid Controller

- Max-speed controller

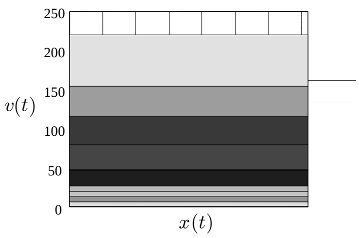
$$\boxed{\begin{array}{l} \max_{u_t} J(u_t, x(t)) \triangleq v(t+1|t) \\ \text{subj. to} \begin{cases} \text{MLD model} \\ x(t|t) = x(t) \end{cases} \end{array}}$$

Objective: maximize speed

(to reproduce max acceleration plots)

MILP optimization problem

Linear constraints	96
Continuous variables	18
Binary variables	10
Parameters	1
Time to solve mp-MILP (Sun Ultra 10)	45 s
Number of regions	**11**



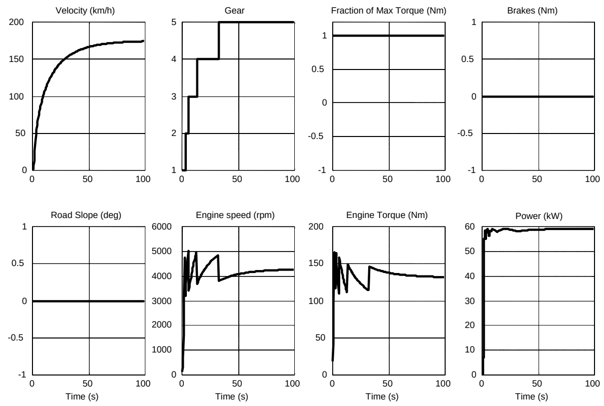
($x(t)$ is irrelevant)

(Parameters: Renault Clio 1.9 DTI RXE)

132

Hybrid Controller

• Max-speed controller



Hybrid Controller

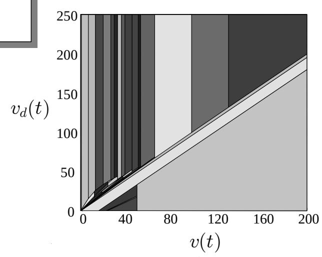
• Tracking controller

$$\min_{u_t} J(u_t, x(t)) \triangleq |v(t+1|t) - v_d(t)| + \rho|\omega|$$

$$\text{subj. to} \begin{cases} \text{MLD model} \\ x(t|t) = x(t) \end{cases}$$

MILP optimization problem

Linear constraints	98
Continuous variables	19
Binary variables	10
Parameters	2
Time to solve mp-MILP (PC 850Mhz)	43 s
Number of regions	**49**

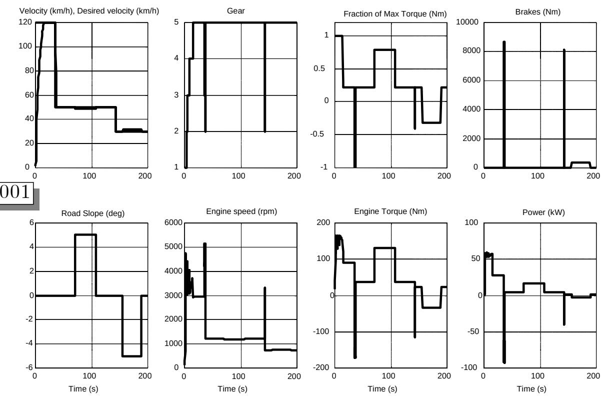


go to demo **/demos/cruise/init_exp.m**

Hybrid Controller

• Tracking controller $\min_{u_t} J(u_t, x(t)) \triangleq |v(t+1|t) - v_d(t)| + \rho|\omega|$

$\rho = 0.001$



Hybrid Controller

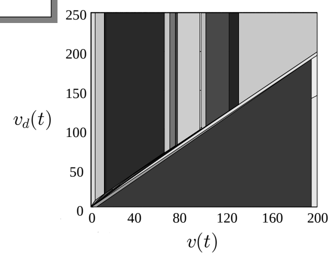
• Smoother tracking controller

$$\min_{u_t} J(u_t, x(t)) \triangleq |v(t+1|t) - v_d(t)| + \rho|\omega|$$

$$\text{subj. to} \begin{cases} |v(t+1|t) - v(t)| < T_s a_{\max} \\ \text{MLD model} \\ x(t|t) = x(t) \end{cases}$$

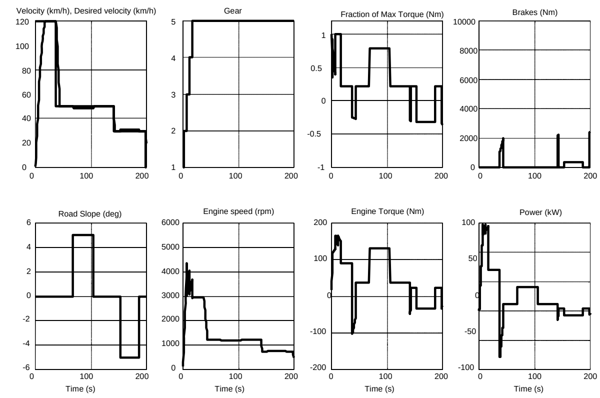
MILP optimization problem

Linear constraints	100
Continuous variables	19
Binary variables	10
Parameters	2
Time to solve mp-MILP (PC 850Mhz)	47 s
Number of regions	**54**



Hybrid Controller

• Smoother tracking controller



Traction Control System

(joint work with F. Borrelli, M. Fodor, D. Hrovat)

Vehicle Traction Control

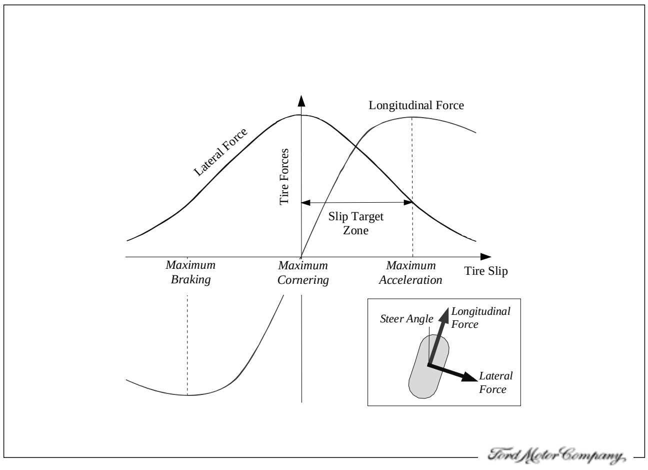
Improve driver's ability to control a vehicle
under adverse external conditions (wet or icy roads)

Model
nonlinear, uncertain, constraints

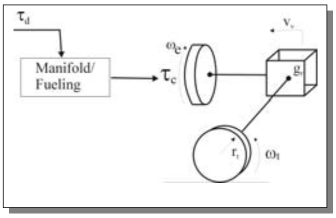
Controller
suitable for real-time implementation

MLD hybrid framework + optimization-based control strategy

Tire Force Characteristics



Simple Traction Model

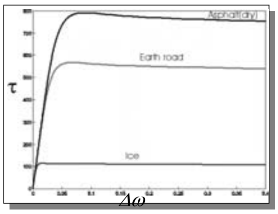


- Mechanical system

$$\dot{\omega}_e = \frac{1}{J_e}\left(\tau_c - b_e\omega_e - \frac{\tau_t}{g_r}\right)$$
$$\dot{v}_v = \frac{\tau_t}{m_v r_t}$$

- Manifold/fueling dynamics

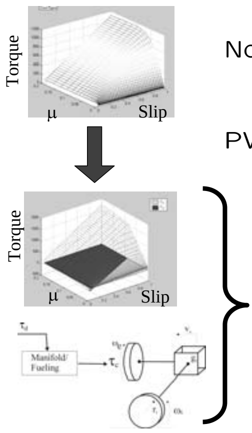
$$\tau_c = b_i\tau_d(t-\tau_f)$$



- Tire torque τ_t is a function of slip $\Delta\omega$ and road surface adhesion coefficient μ

$$\Delta\omega = \frac{\omega_e}{g_r} - \frac{v_v}{r_t}$$ Wheel slip

Hybrid Model



Nonlinear tire torque $\tau_t = f(\Delta\omega, \mu)$

PWA Approximation
(PWL Toolbox, Julian, 1999)

HYSDEL
(Hybrid Systems Description Language)

→ Mixed-Logical Dynamical (MLD) Hybrid Model (discrete time)

MLD Model

$$\begin{aligned} x(t+1) &= Ax(t) + B_1u(t) + B_2\delta(t) + B_3z(t) + B_5 \\ y(t) &= Cx(t) + D_1u(t) + D_2\delta(t) + D_3z(t) + D_5 \\ E_2\delta(t) + E_3z(t) &\leq E_4x(t) + E_1u(t) + E_5 \end{aligned}$$

State x(t)	4 variables
Input u(t)	1 variable
Aux. Binary vars δ(t)	1 variable
Aux. Continuous vars z(t)	3 variables
Mixed-integer inequalities	14

➡ The MLD matrices are automatically generated in Matlab format by HYSDEL

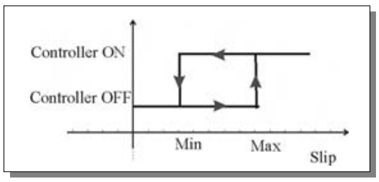
go to demo **`/demos/traction/init.m`**

Performance and Constraints

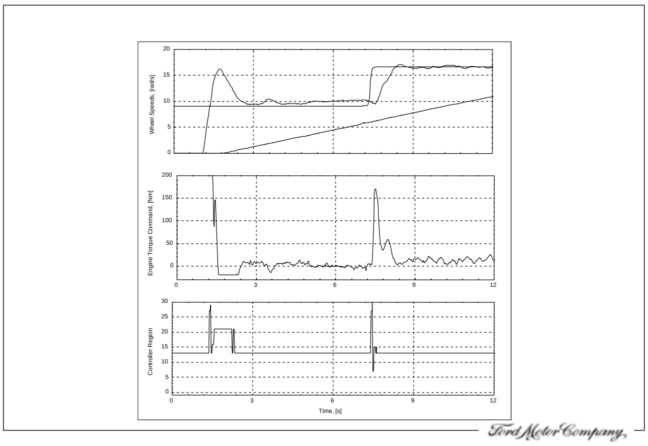
- Control objective:

$$\min \sum_{k=0}^{N} |\Delta\omega(k|t) - \Delta\omega_{des}|$$
subj. to. MLD Dynamics

- Constraints:
  - Limits on the engine torque: $-20Nm \leq \tau_d \leq 176Nm$
- Logic Constraint:
  - Hysteresis

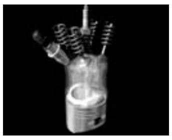


Experimental Results



Experiment

- ≈500 regions
- 20ms sampling time
- Pentium 266Mhz + Labview



Hybrid Control of a DISC Engine

(joint work with N. Giorgetti, I. Kolmanovsky and D. Hrovat)

DISC Engine Control Problem

Goal: Develop a controller for a Direct-Injection Stratified Charge (DISC) engine.

Main features of DISC engines:

- Direct injection of fuel (as Diesel engines);
- Two **operating modes** (homogeneous/stratified);
- Reduction of consumptions up to 15%;
- Highly complex post-treatment system of exhaust gas;
- Different control objectives and **constraints** that depend on the operating mode

(Photo: Courtesy Mitsubishi)

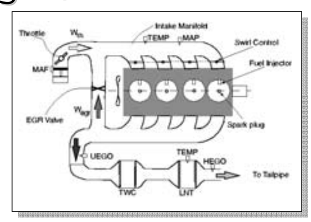
Strategy:

- Develop a hybrid model that captures the two operating modes;
- Design a constrained MPC controller based on the hybrid model

DISC Engine

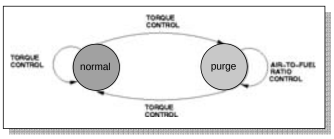
Two distinct regimes:

Regime	fuel injection	air-to-fuel ratio
Homogeneous combustion	intake stroke	λ=14.64
Stratified combustion	compression stroke	λ [illegible] 14.64



- Mode is switched by changing fuel injection timing (late / early)
- Better fuel economy during stratified mode

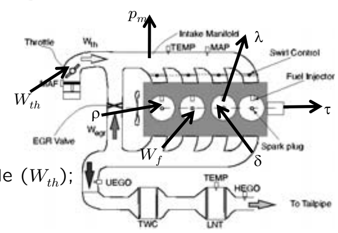
Periodical cleaning of the aftertreatment system needed (λ=14.00, homogeneous regime)



the stratified operation can only be sustained in a restricted part of the engine operating range

DISC Engine

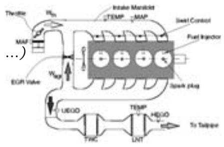
- States/Controlled outputs:
  - Intake manifold pressure (p_m);
  - Air-to-fuel ratio (λ);
  - Engine brake torque (τ);
- Inputs (continuous):
  - Air mass flow rate through throttle (W_{th});
  - Mass flow rate of fuel (W_f);
  - Spark timing (δ);
- Inputs (binary):
  - ρ = regime of combustion (homogeneous/stratified);
- Constraints on:
  - Air-to-fuel ratio (due to engine roughness, misfiring, smoke emiss.)
  - Spark timing (to avoid excessive engine roughness)
  - Mass flow rate on intake manifold (constraints on throttle)



- Dynamic equations are nonlinear
- Dynamics and constraints depend on regime ρ

DISC Dynamics

Nonlinear model of the engine developed and validated at Ford Research *(Kolmanovsky, Sun, ...)*



Assumptions:
- no EGR (exhaust gas recirculation) rate,
- engine speed=2000 rpm.

- Intake manifold pressure: $\dot{p_m} = c_m \left(W_{th} - W_{cyl}\right) = c_m \left(W_{th} - k_{cyl} p_m\right)$
- In-cylinder Air-to-Fuel ratio: $\lambda = \frac{W_{cyl}}{W_f} = \frac{k_{cyl} p_m}{W_f}$
- Engine torque: $\tau = \tau_{mfr} + \tau_{pump} + \tau_{ind}$ with τ_{mfr}, τ_{pump} functions of p_m

$\tau_{ind} = (\theta_a + \theta_b(\delta - \delta_{mbt})^2) W_f$ where θ_a, θ_b, δ_{mbt} are functions of λ, δ and ρ.

✓ Good for simulation
× Not suitable for controller synthesis

Hybridization of DISC Model

DYNAMICS (intake pressure, air-to-fuel ratio, torque):

- Definition of two operating points;
- Linearization of nonlinear dynamics;
- Time discretization of the linear models.

→ ρ-dependent dynamic equations

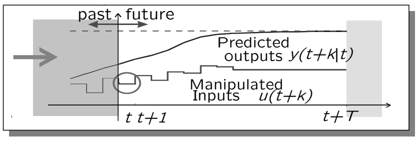
CONSTRAINTS on:

- Air-to-Fuel Ratio: $\lambda_{min}(\rho) \leq \lambda(t) \leq \lambda_{max}(\rho)$;
- Mass of air through the throttle: $0 \leq W_{th} \leq K$;
- Spark timing: $0 \leq \delta(t) \leq \delta_{mbt}(\lambda, \rho)$

→ ρ–dependent constraints

Hybrid system with 2 modes (switching affine system)

MPC for Hybrid Systems



Model Predictive (MPC) Control

- At time t solve with respect to $U \triangleq \{u_t, u_{t+1}, \ldots, u_{t+T-1}\}$ the finite-horizon open-loop, optimal control problem:

$$\min_U J(U, x(t)) = \sum_{k=0}^{N-1} u'_{t+k} R u_{t+k} + [y'_{t+k} - r(t)] Q [y_{t+k} - r(t)]$$

$$\text{subj. to} \begin{cases} x_t = x(t), \\ \text{hybrid model} \end{cases}$$

- Apply only $u(t) = u_t^*$ (discard the remaining optimal inputs);
- Repeat the whole optimization at time $t+1$

Integral Action

Integrators on torque error and air-to-fuel ratio error are added to obtain zero offsets in steady-state:

$$\epsilon_\tau(t+1) = \epsilon_\tau(t) + T \cdot (\tau_{ref} - \tau)$$
$$\epsilon_\lambda(t+1) = \epsilon_\lambda(t) + T \cdot (\lambda_{ref} - \lambda)$$

T = sampling time

τ_{ref}, λ_{ref} brake torque and air-to-fuel references

Simulation based on nonlinear model confirms zero offsets in steady-state

(despite the model mismatch)

MPC of DISC Engine

$$\min_\xi J(\xi, x(t)) = \sum_{k=0}^{N-1} u'_k R u_k + y'_k Q y_k + x'_{k+1} S x_{k+1}$$

$$\text{subj. to} \begin{cases} x_0 = x(t), \\ \text{hybrid model} \end{cases}$$

N = control horizon

$x(t)$ = current state

$\xi = [u'_0, \gamma'_0, z'_0, \ldots, u'_{N-1}, \gamma'_{N-1}, z'_{N-1}]'$,

where:
$$\begin{aligned} u_k &= [W_{th,k} - W_{th,ref},\ W_{f,k} - W_{f,ref}, \delta_k - \delta_{ref},\ \rho_k - \rho_{ref}]' \\ y_k &= [\tau_k - \tau_{ref},\ \lambda_k - \lambda_{ref}]' \\ x_k &= [p_{m,k} - p_{m,ref},\ \epsilon_{\tau,k},\ \epsilon_{\lambda,k}]' \end{aligned}$$

and:
$$R = \begin{pmatrix} r_{W_{th}} & 0 & 0 & 0 \\ 0 & r_{W_f} & 0 & 0 \\ 0 & 0 & r_\delta & 0 \\ 0 & 0 & 0 & r_\rho \end{pmatrix} \quad Q = \begin{pmatrix} q_\tau & 0 \\ 0 & q_\lambda \end{pmatrix} \quad S = \begin{pmatrix} s_{p_m} & 0 & 0 \\ 0 & s_{\epsilon_\tau} & 0 \\ 0 & 0 & s_{\epsilon_\lambda} \end{pmatrix}$$

DISC Engine - HYSDEL List

```
SYSTEM hysdisc{
   INTERFACE{
      STATE{
         REAL pm      [1, 101.325];
         REAL xtau    [-1e3, 1e3];
         REAL xlam    [-1e3, 1e3];
         REAL taud    [0,    100];
         REAL lamd    [10,    60];
      }
      OUTPUT {
         REAL lambda;  /* [10,     50]; */
         REAL tau;     /* [0,     100]; */
      }
      INPUT{
         REAL Wth     [0,38.5218];
         REAL Wf      [0,      2];
         REAL delta   [0,     40];
         BOOL rho;
      }
      PARAMETER{
         REAL Ts, pm1, pm2;
         REAL l01, l02, l0c;
         REAL l11,l12,l1c;
         REAL t01,t02,t03,t04,t05;
         REAL t11,t12,t13,t14,t15;
      }
   }
IMPLEMENTATION{
      AUX{
          REAL lam, taul;
          REAL lmin,lmax;
          REAL dmbt;  }
DA{
  lam={ IF rho THEN l11*pm+l12*Wf+l1c
                ELSE   l01*pm+l02*Wf+l0c};
  taul={IF rho THEN
            t11*pm+t12*Wf+t13*delta+t14*lam+t15
     ELSE   t01*pm+t02*Wf+t03*delta+t04*lam+t05 };
  lmin={IF rho THEN 13 ELSE 19};
  lmax ={IF rho THEN 21 ELSE 38};
  dmbt ={IF rho THEN -28.74+3.1845*lam
                ELSE 14.0877+0.2810*lam};
}
CONTINUOUS{
   pm=pm1*pm+pm2*Wth;
   xtau=xtau+Ts*(taud-taul);
   xlam=xlam+Ts*(lamd-lam);
   taud=taud;
   lamd=lamd;
}
OUTPUT {
   lambda=lam;
   tau=taul;
}
MUST{
   lmin-lam     <=0;
   lam-lmax     <=0;
   delta-dmbt   <=0;
}
}}
```

MPC – Torque Control Mode

$$\min_{\xi} J(\xi, x(t)) = \sum_{k=0}^{N-1} u_k' R u_k + y_k' Q y_k + x_{k+1}' S x_{k+1}$$

subj. to $\begin{cases} x_0 = x(t), \\ \text{hybrid model} \end{cases}$

$u(t) = [W_{th}(t),\ W_f(t),\ \delta(t),\ \rho(t)]$

Solve **MIQP problem** (mixed-integer quadratic program) to compute $u(t)$

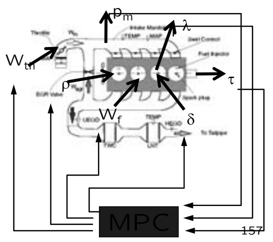
Weights:

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.1 \end{pmatrix} \quad Q = \begin{pmatrix} 10 & 0 \\ 0 & 0.01 \end{pmatrix} \quad S = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 10^4 & 0 \\ 0 & 0 & 10 \end{pmatrix}$$

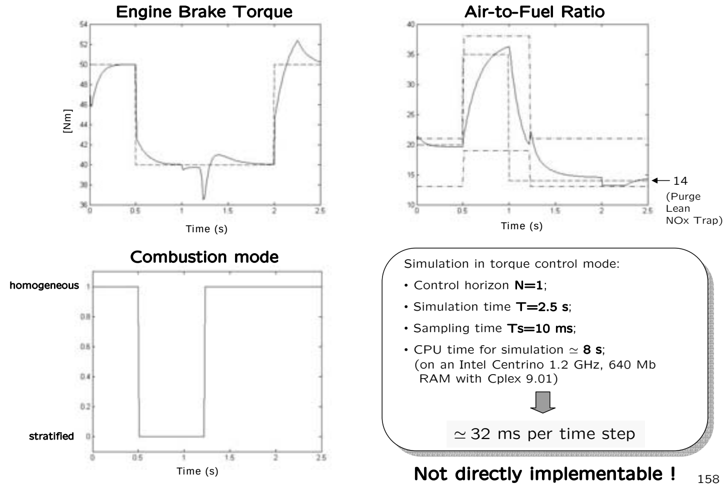
r_ρ (prevents unneeded chattering)

q_τ q_λ s_{ε_τ} s_{ε_λ}

main emphasis on torque



Simulation Results



Simulation in torque control mode:

- Control horizon **N=1**;
- Simulation time **T=2.5 s**;
- Sampling time **Ts=10 ms**;
- CPU time for simulation ≃ **8 s**; (on an Intel Centrino 1.2 GHz, 640 Mb RAM with Cplex 9.01)

≃ 32 ms per time step

Not directly implementable !

Multiparametric (Explicit) Solution

$$\min_{\xi} J(\xi, x(t)) = \sum_{k=0}^{N-1} u_k' R u_k + y_k' Q y_k + x_{k+1}' S x_{k+1}$$

subj. to $\begin{cases} x_0 = x(t) \\ \text{hybrid model} \end{cases}$

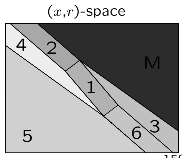
Mixed Integer Quadratic Program

- On-line optimization: given $x(t)$, solve an MIQP at each t.
- Off-line optimization: solve the MIQP **for all** $x(t)$ off line
  - find feasible switching sequences using reachability analysis
  - solve sequence of **multi-parametric quadratic programs** (using the Hybrid Toolbox)

The MPC controller is piecewise affine in x,r

$$u(x,r) = \begin{cases} F_1 x + E_1 r + g_1 & \text{if } H_1 \left[\begin{smallmatrix} x \\ r \end{smallmatrix}\right] \le K_1 \\ \vdots & \vdots \\ F_M x + E_M r + g_M & \text{if } H_M \left[\begin{smallmatrix} x \\ r \end{smallmatrix}\right] \le K_M \end{cases}$$

(x,r)-space



Explicit MPC Controller

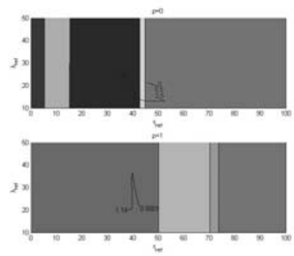
Explicit control law: $u(t) = f(\theta(t))$

where: $u = [W_{th}\ W_f\ \delta\ \rho]'$
$\theta = [p_m\ \epsilon_\tau\ \epsilon_\lambda\ \tau_{\text{ref}}\ \lambda_{\text{ref}}$
$p_{m,\text{ref}}\ W_{th,\text{ref}}\ W_{f,\text{ref}}\ \delta_{\text{ref}}]'$

N=1 (control horizon)

75 partitions

Cross-section by the τ_{ref}-λ_{ref} plane

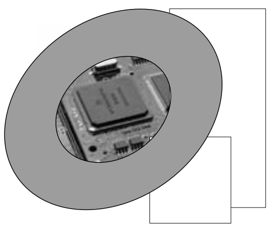


- Time to compute the explicit MPC: 3.4750 s;
- Simulation time T=2.5 s;
- Sampling time Ts=10 ms;
- CPU time for simulation: **0.52 s**

≃ **10 μs** per time step

Microcontroller Implementation

- C-code **automatically** generated by the Hybrid Toolbox
- Microcontroller Motorola MPC 555 (custom made for Ford)
- 43 Kb memory available
- Floating point arithmetic



≈ 3ms execution time

sampling period = 10ms → **Implementable !**

- Further reduction of number of partitions possible *(Alessio, Bemporad, 2005)*
- C-code can be further optimized

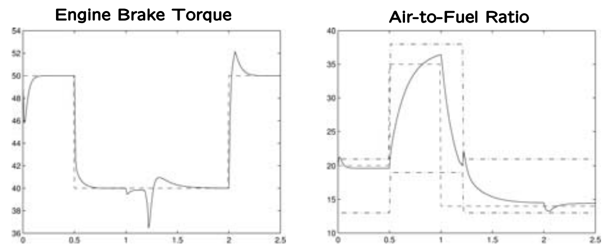
Explicit MPC Controller (N=2)

Explicit control law: $u(t) = f(\theta(t))$

where: $u = [W_{th}\ W_f\ \delta\ \rho]'$
$\theta = [p_m\ \epsilon_\tau\ \epsilon_\lambda\ \tau_{\text{ref}}\ \lambda_{\text{ref}}$
$p_{m,\text{ref}}\ W_{th,\text{ref}}\ W_{f,\text{ref}}\ \delta_{\text{ref}}]'$

N=2 (control horizon)

5637 partitions



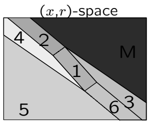
Closed-loop N=2 ⟺ Closed-loop N=1 ⇐ Adequate!

Conclusions

- **Hybrid systems** are a modeling framework for automotive control problems where continuous switched dynamics and logic are relevant (and linear models are not enough!)
- **MPC control design** handle all performance specs and constraints in a natural and direct way. Quite complex systems can be controlled using on-line optimization
- **Piecewise affine MPC controllers** can be synthesized, off-line, and implemented as look-up tables of linear gains

$$u(x,r) = \begin{cases} F_1 x + E_1 r + g_1 & \text{if } H_1 \left[\begin{smallmatrix} x \\ r \end{smallmatrix}\right] \leq K_1 \\ \vdots & \vdots \\ F_M x + E_M r + g_M & \text{if } H_M \left[\begin{smallmatrix} x \\ r \end{smallmatrix}\right] \leq K_M \end{cases}$$

(x,r)-space



- **Matlab tools** available to assist the whole design process (models, simulation, MPC design, code generation):
  MPC Toolbox (linear), Hybrid Toolbox (hybrid, explicit), Multi-Parametric Toolbox (PWA, explicit)

Key References

[1] A. Bemporad, ''Modeling, control, and reachability analysis of discrete-time hybrid systems,'' Lecture Notes - DISC School on Hybrid Systems, Mar. 2003.

[2] F.D. Torrisi and A. Bemporad, ''HYSDEL - A tool for generating computational hybrid models,'' *IEEE Trans. on Control Systems Technology*, vol. 12, no. 2, pp. 235-249, Mar. 2004

[3] A. Bemporad and M. Morari, ''Control of systems integrating logic, dynamics, and constraints,'' *Automatica*, vol. 35, no. 3, pp. 407-427, Mar. 1999.

[4] A. Bemporad, ''Efficient conversion of mixed logical dynamical systems into an equivalent piecewise affine form,'' *IEEE Trans. Automatic Control*, vol. 49, no. 5, pp. 832-838, 2004.

[5] W.P.H.M Heemels, B. de Schutter, and A. Bemporad, ''Equivalence of hybrid dynamical models,'' *Automatica*, vol. 37, no. 7, pp. 1085-1091, July 2001

[6] A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos, ''The explicit linear quadratic regulator for constrained systems,'' *Automatica*, vol. 38, no. 1, pp. 3-20, 2002.

more on **http://www.dii.unisi.it/~bemporad/publications**

Key References

[7] A. Bemporad, F. Borrelli, and M. Morari, ''Piecewise linear optimal controllers for hybrid systems,'' In *Proc. American Control Conference*, 2000, pp. 1190-1194.

[8] F. Borrelli, M. Baotic, A. Bemporad, and M. Morari, ''Dynamic programming for constrained optimal control of discrete-time linear hybrid systems,'' *Automatica*, vol. 41, no. 10, Oct. 2005.

[9] A. Bemporad and N. Giorgetti, ''Logic-based methods for optimal control of hybrid systems,'' *IEEE Trans. Autom. Control*, 2005, to appear

[10] A. Bemporad, D. Mignone, and M. Morari, ''Moving horizon estimation for hybrid systems and fault detection,'' in *Proc. American Control Conf.*, 1999, Chicago, IL, pp. 2471-2475.

[11] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino, ''A bounded-error approach to piecewise affine system identification,'' *IEEE Trans. Automatic Control*, 2005, to appear.

[12] A. Bemporad, ''Hybrid Toolbox – User's Guide,'' Dec. 2003, http://www.dii.unisi.it/hybrid/toolbox.

[13] M. Lazar, M. Heemels, S. Weiland, A. Bemporad, ''Stability of Hybrid Model Predictive Control,'' *IEEE Trans. Automatic Control*, 2005, submitted.

more on **http://www.dii.unisi.it/~bemporad/publications**