Model Predictive Control

Manfred Morari

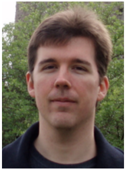
Institut für Automatik
ETH Zürich

Spring Semester 2014

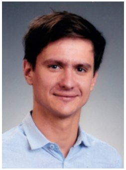
Lecturers

Prof. Dr. Manfred Morari
ETH Zurich
Institut für Automatik (IfA)

Prof. Dr. Francesco Borrelli
University of California, Berkeley
Model Predictive Control Lab

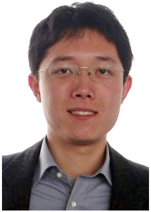


Dr. Paul J. Goulart
ETH Zurich
Institut für Automatik (IfA)



Dr. Alexander Domahidi
inspire-IfA

Head Teaching Assistants



Xiaojing (George) Zhang
xiaozhan@control.ee.ethz.ch

David Sturzenegger
sturzenegger@control.ee.ethz.ch

Lecture Material

Compilation:	Xiaojing (George) Zhang, David Sturzenegger Please email suggestions and typos to xiaozhan@control.ee.ethz.ch sturzenegger@control.ee.ethz.ch
Software:	*Beamer* for LaTeX by Till Tantau & Vedran Milentić
Printed material:	Available in 2-page layout Sold during lecture or later at ETL I23 for CHF 30
Recordings:	Entire lecture is video recorded Link will be provided on lecture homepage
Homepage:	http://control.ee.ethz.ch/index.cgi?page=lectures

About the Lecture

Duration: Monday, 17. Feb 2014 – Friday, 28. Feb 2014

Credits: 6 credits for passing the exam

Exercises: Computer excercises, ETZ D61.1/2

Exam: Fri, 14. March 2014 (written), Location: tba

Week 1:

Date	Topic	Lectures		Exercises
		Time	Location	
Mon, Feb 17	Linear Systems I	9.15 – 12	HG E3	13.15 – 17
Tue, Feb 18	Linear Systems II	9.15 – 12	HG E3	13.15 – 17
Wed, Feb 19	Optimization I	9.15 – 12	HG D16.2	13.15 – 17
Thu, Feb 20	Optimization II	9.15 – 12	HG D16.2	13.15 – 17
Fri, Feb 21	Introduction to MPC	9.15 – 12	HG E3	13.15 – 17

About the Lecture

Duration: Monday, 17. Feb 2014 – Friday, 28. Feb 2014

Credits: 6 credits for passing the exam

Exercises: Computer excercises, ETZ D61.1/2

Exam: Fri, 14. March 2014 (written), Location: tba

Week 2:

Date	Topic	Lectures		Exercises
		Time	Location	
Mon, Feb 24	Numerical Methods	9.15 – 12	HG E3	13.15 – 17
Tue, Feb 25	$\boxed{\text{Advanced Topics I}}$	9.15 – 12	HG E3	13.15 – 17
Wed, Feb 26	Invited Talks	9.15 – 17	HG D16.2	—
Thu, Feb 27	Design Exercise	—	—	10.15 – 17
Fri, Feb 28	$\boxed{\text{Advanced Topics II}}$	9.15 – 12	HG D16.2	—

Model Predictive Control
Part I – Introduction

C. Jones[†], F. Borrelli[*], M. Morari

Institut für Automatik
ETH Zürich

[*]UC Berkley

[†] EPFL

Spring Semester 2014

Literature

Model Predictive Control:

- Predictive Control for linear and hybrid systems, F. Borrelli, A. Bemporad, M. Morari, 2013 Cambridge University Press [http://www.mpc.berkeley.edu/mpc-course-material]
- Model Predictive Control: Theory and Design, James B. Rawlings and David Q. Mayne, 2009 Nob Hill Publishing
- Predictive Control with Constraints, Jan Maciejowski, 2000 Prentice Hall

Optimization:

- Convex Optimization, Stephen Boyd and Lieven Vandenberghe, 2004 Cambridge University Press
- Numerical Optimization, Jorge Nocedal and Stephen Wright, 2006 Springer

Table of Contents

Main Idea

Objective:

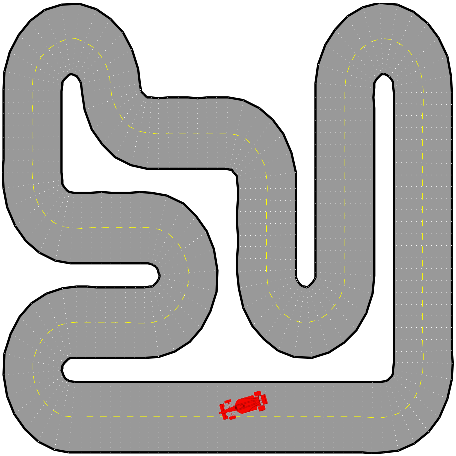
- Minimize lap time

Constraints:

- Avoid other cars
- Stay on road
- Don't skid
- Limited acceleration

Intuitive approach:

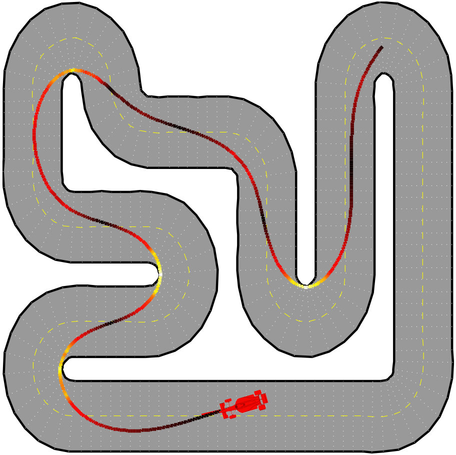
- Look forward and plan path based on
  - Road conditions
  - Upcoming corners
  - Abilities of car
  - etc...



Optimization-Based Control

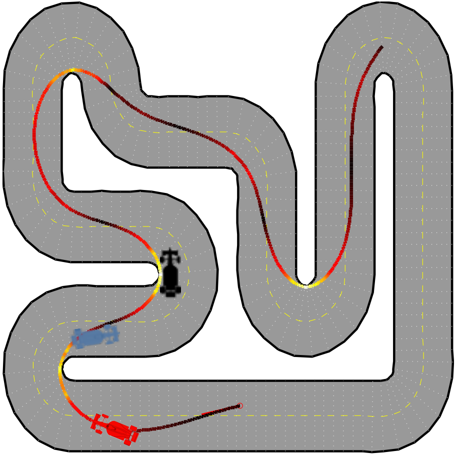
Minimize	(lap time)
while	avoid other cars
	stay on road
	…

- Solve **optimization problem** to compute minimum-time path



Optimization-Based Control

Minimize	(lap time)
while	avoid other cars
	stay on road
	...

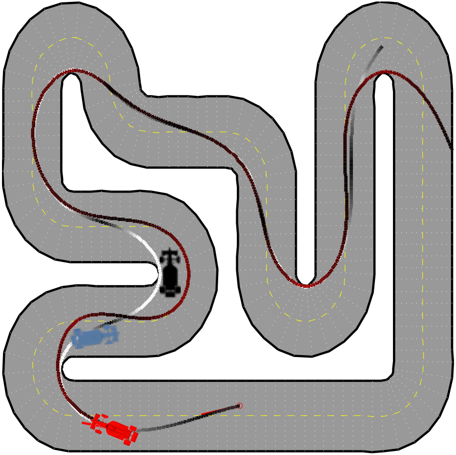


- Solve **optimization problem** to compute minimum-time path
- What to do if something unexpected happens?
  - We didn't see a car around the corner!
  - Must introduce *feedback*

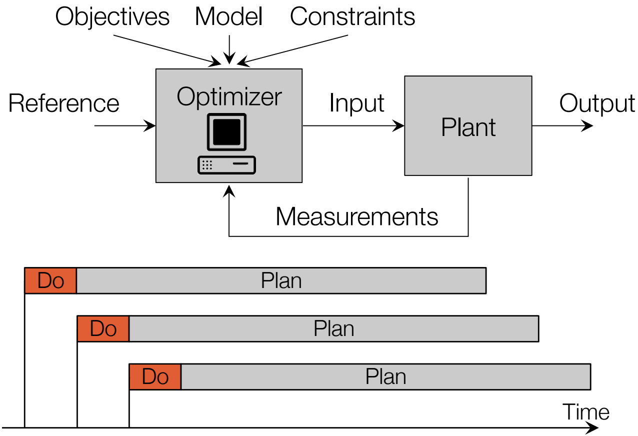
Optimization-Based Control

Minimize	(lap time)
while	avoid other cars
	stay on road
	...

- Solve **optimization problem** to compute minimum-time path
- Obtain series of planned control actions
- Apply *first* control action
- Repeat the planning procedure



Model Predictive Control

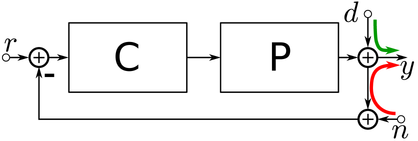


Receding horizon strategy introduces **feedback**.

Table of Contents

Two Different Perspectives

Classical design: design C

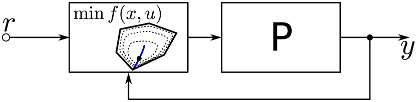


Dominant issues addressed

- Disturbance rejection $(d \to y)$
- Noise insensitivity $(n \to y)$
- Model uncertainty

(usually in *frequency domain*)

MPC: real-time, repeated optimization to choose $u(t)$



Dominant issues addressed

- Control constraints (limits)
- Process constraints (safety)

(usually in *time domain*)

Constraints in Control

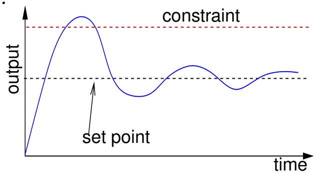
All physical systems have **constraints**:

- Physical constraints, e.g. actuator limits
- Performance constraints, e.g. overshoot
- Safety constraints, e.g. temperature/pressure limits

Optimal operating points are often near constraints.

Classical control methods:

- Ad hoc constraint management
- Set point sufficiently far from constraints
- Suboptimal plant operation



Predictive control:

- Constraints included in the design
- Set point optimal
- Optimal plant operation

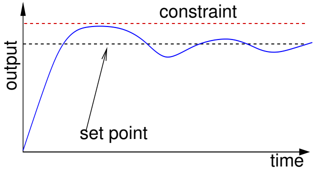


Table of Contents

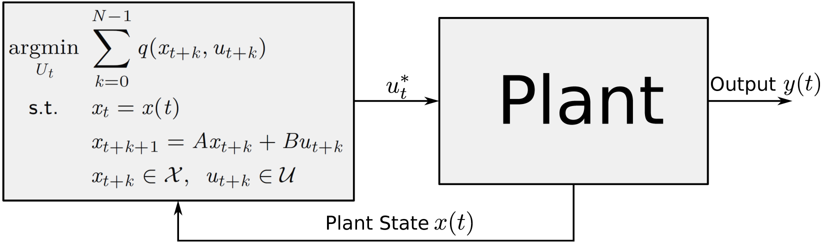
MPC: Mathematical Formulation

$$
\begin{aligned}
U_t^*(x(t)) := \operatorname*{argmin}_{U_t} \; & \sum_{k=0}^{N-1} q(x_{t+k}, u_{t+k}) && \\
\text{subj. to } \; & x_t = x(t) && \text{measurement} \\
& x_{t+k+1} = Ax_{t+k} + Bu_{t+k} && \text{system model} \\
& x_{t+k} \in \mathcal{X} && \text{state constraints} \\
& u_{t+k} \in \mathcal{U} && \text{input constraints} \\
& U_t = \{u_0, u_1, \ldots, u_{N-1}\} && \text{optimization variables}
\end{aligned}
$$

Problem is defined by

- **Objective** that is minimized,
  e.g., distance from origin, sum of squared/absolute errors, economic,...
- Internal **system model** to predict system behavior
  e.g., linear, nonlinear, single-/multi-variable, ...
- **Constraints** that have to be satisfied
  e.g., on inputs, outputs, states, linear, quadratic,...

MPC: Mathematical Formulation



At each sample time:

- Measure / estimate current state $x(t)$
- Find the optimal input sequence for the entire planning window N:
  $U_t^* = \{u_t^*, u_{t+1}^*, \ldots, u_{t+N-1}^*\}$
- Implement only the *first* control action u_t^*

Problem Formulation

Quadratic cost function

$$J_0(x(0), U_0) = x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k \tag{2}$$

with $P \succeq 0,\ Q \succeq 0,\ R \succ 0$.
Constrained Finite Time Optimal Control problem (CFTOC).

$$\boxed{\begin{array}{rll} J_0^*(x(0)) = & \min\limits_{U_0} & J_0(x(0), U_0) \\ & \text{subj. to} & x_{k+1} = Ax_k + Bu_k,\ k = 0, \ldots, N-1 \\ & & x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ k = 0, \ldots, N-1 \\ & & x_N \in \mathcal{X}_f \\ & & x_0 = x(0) \end{array}} \tag{3}$$

N is the time horizon and $\mathcal{X}$, $\mathcal{U}$, $\mathcal{X}_f$ are polyhedral regions.

Construction of the QP with substitution

- **Step 1**: Rewrite the cost as (see lectures on Day 1 & 2)

$$\begin{aligned} J_0(x(0), U_0) &= U_0' H U_0 + 2x(0)' F U_0 + x(0)' Y x(0) \\ &= [U_0' \; x(0)'] \left[\begin{smallmatrix} H & F' \\ F & Y \end{smallmatrix}\right] [U_0{}' \; x(0)']' \end{aligned}$$

  Note: $\left[\begin{smallmatrix} H & F' \\ F & Y \end{smallmatrix}\right] \succeq 0$ since $J_0(x(0), U_0) \geq 0$ by assumption.

- **Step 2**: Rewrite the constraints compactly as (details provided on the next slide)

$$G_0 U_0 \leq w_0 + E_0 x(0)$$

- **Step 3**: Rewrite the optimal control problem as

$$\boxed{\begin{aligned} J_0^*(x(0)) = \min_{U_0} \quad & [U_0' \; x(0)'] \left[\begin{smallmatrix} H & F' \\ F & Y \end{smallmatrix}\right] [U_0{}' \; x(0)']' \\ \text{subj. to} \quad & G_0 U_0 \leq w_0 + E_0 x(0) \end{aligned}}$$

Solution

$$
\begin{aligned}
J_0^*(x(0)) = \min_{U_0} \quad & [U_0' \; x(0)'] \left[\begin{smallmatrix} H & F' \\ F & Y \end{smallmatrix}\right] [U_0{}' \; x(0)']' \\
\text{subj. to} \quad & G_0 U_0 \leq w_0 + E_0 x(0)
\end{aligned}
$$

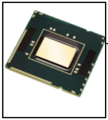
For a given x(0) U_0^* can be found via a QP solver.

Summary

- Obtain a model of the system
- Design a state observer
- Define optimal control problem
- Set up optimization problem in optimization software
- Solve optimization problem to get optimal control sequence
- Verify that closed-loop system performs as desired,
  e.g., check performance criteria, robustness, real-time aspects,...

Table of Contents

MPC: Applications



Computer control — ns

μs — Power systems

Traction control — ms

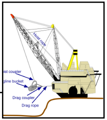
Seconds — Buildings

Refineries — Minutes

Hours — Nurse rostering

Train scheduling — Days

Weeks — Production planning



Important Aspects of Model Predictive Control

Main advantages:

- Systematic approach for handling *constraints*
- High *performance* controller

Main challenges:

- *Implementation*
  MPC problem has to be solved in real-time, i.e. within the sampling interval of the system, and with available hardware (storage, processor,...).
- *Stability*
  Closed-loop stability, i.e. convergence, is not automatically guaranteed
- *Robustness*
  The closed-loop system is not necessarily robust against uncertainties or disturbances
- *Feasibility*
  Optimization problem may become infeasible at some future time step, i.e. there may not exist a plan satisfying all constraints

Model Predictive Control
Part III – Feasibility and Stability

F. Borrelli*, C. Jones†, M. Morari

Institut für Automatik
ETH Zürich

*UC Berkley

† EPFL

Spring Semester 2014
revised 29.04.2014

Infinite Time Constrained Optimal Control (what we would like to solve)

$$
\begin{aligned}
J_0^*(x(0)) = \min \sum_{k=0}^{\infty} q(x_k, u_k) \\
\text{s.t. } & x_{k+1} = Ax_k + Bu_k, k = 0, \dots, N-1 \\
& x_k \in \mathcal{X}, u_k \in \mathcal{U}, k = 0, \dots, N-1 \\
& x_0 = x(0)
\end{aligned}
$$

- **Stage cost** $q(x, u)$ describes "cost" of being in state x and applying input u
- Optimizing over a trajectory provides a **tradeoff between short- and long-term benefits** of actions
- We'll see that such a control law has many beneficial properties...
  ... but we can't compute it: there are an **infinite number of variables**

Receding Horizon Control (what we can sometimes solve)

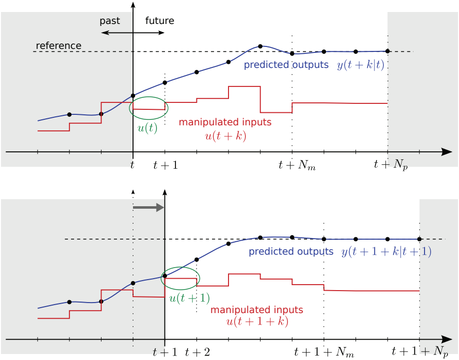
$$
\begin{array}{rll}
J_t^*(x(t)) = & \min\limits_{U_t} & p(x_{t+N}) + \sum\limits_{k=0}^{N-1} q(x_{t+k}, u_{t+k}) \\
& \text{subj. to} & x_{t+k+1} = Ax_{t+k} + Bu_{t+k},\ k = 0, \ldots, N-1 \\
& & x_{t+k} \in \mathcal{X},\ u_{t+k} \in \mathcal{U},\ k = 0, \ldots, N-1 \\
& & x_{t+N} \in \mathcal{X}_f \\
& & x_t = x(t)
\end{array} \quad (1)
$$

where $\mathcal{U}_t = \{u_t, \ldots, u_{t+N-1}\}$.

Truncate after a finite horizon:

- $p(x_{t+N})$: Approximates the 'tail' of the cost
- $\mathcal{X}_f$: Approximates the 'tail' of the constraints

On-line Receding Horizon Control



1. At each sampling time, solve a **CFTOC**.
2. Apply the optimal input **only during** $[t, t+1]$
3. At $t+1$ solve a CFTOC over a **shifted horizon** based on new state measurements
4. The resultant controller is referred to as **Receding Horizon Controller (RHC)** or **Model Predictive Controller (MPC)**.

On-line Receding Horizon Control

1) MEASURE the state $x(t)$ at time instance t
2) OBTAIN $U_t^*(x(t))$ by solving the optimization problem in (1)
3) IF $U_t^*(x(t)) = \emptyset$ THEN 'problem infeasible' STOP
4) APPLY the first element u_t^* of U_t^* to the system
5) WAIT for the new sampling time $t+1$, GOTO 1)

Note that, we need a constrained optimization solver for step 2).

History of MPC

- **A. I. Propoi, 1963**, “Use of linear programming methods for synthesizing sampled-data automatic systems”, *Automation and Remote Control*.
- **J. Richalet et al., 1978** “Model predictive heuristic control- application to industrial processes”. *Automatica*, 14:413-428.
  - known as **IDCOM (Identification and Command)**
  - impulse response model for the plant, linear in inputs or internal variables (**only stable plants**)
  - quadratic performance objective over a finite prediction horizon
  - future plant output behavior specified by a reference trajectory
  - **ad hoc** input and output constraints
  - optimal inputs computed using a heuristic iterative algorithm, interpreted as the dual of identification
  - controller was not a transfer function, hence called **heuristic**

History of MPC

- 1970s: Cutler suggested MPC in his PhD proposal at the University of Houston in 1969 and introduced it later at Shell under the name Dynamic Matrix Control. **C. R. Cutler, B. L. Ramaker, 1979** "Dynamic matrix control – a computer control algorithm". *AICHE National Meeting*, Houston, TX.
  - successful in the petro-chemical industry
  - linear step response model for the plant
  - quadratic performance objective over a finite prediction horizon
  - future plant output behavior specified by trying to follow the set-point as closely as possible
  - input and output constraints included in the formulation
  - optimal inputs computed as the solution to a least–squares problem
  - **ad hoc** input and output constraints. Additional equation added online to account for constraints. Hence a **dynamic matrix** in the least squares problem.
- **C. Cutler, A. Morshedi, J. Haydel, 1983**. "An industrial perspective on advanced control". *AICHE Annual Meeting*, Washington, DC.
  - Standard QP problem formulated in order to systematically account for constraints.

History of MPC

- Mid 1990s: extensive theoretical effort devoted to provide conditions for guaranteeing feasibility and closed-loop stability
- 2000s: development of tractable robust MPC approaches; nonlinear and hybrid MPC; MPC for very fast systems
- 2010s: stochastic MPC; distributed large-scale MPC; economic MPC

Table of Contents

MPC Features

Pros

- Any model
  - linear
  - nonlinear
  - single/multivariable
  - time delays
  - constraints
- Any objective:
  - sum of squared errors
  - sum of absolute errors (i.e., integral)
  - worst error over time
  - economic objective

Cons

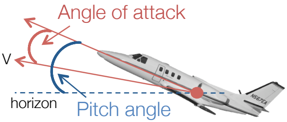
- Computationally demanding in the general case
- May or may not be stable
- May or may not be feasible

Example: Cessna Citation Aircraft

Linearized continuous-time model:
(at altitude of 5000m and a speed of 128.2 m/sec)

$$\dot{x} = \begin{bmatrix} -1.2822 & 0 & 0.98 & 0 \\ 0 & 0 & 1 & 0 \\ -5.4293 & 0 & -1.8366 & 0 \\ -128.2 & 128.2 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} -0.3 \\ 0 \\ -17 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x$$



- Input: elevator angle
- States: x_1: angle of attack, x_2: pitch angle, x_3: pitch rate, x_4: altitude
- Outputs: pitch angle and altitude
- Constraints: elevator angle ± 0.262rad ($\pm 15°$), elevator rate ± 0.524rad ($\pm 60°$), pitch angle ± 0.349 ($\pm 39°$)

Open-loop response is unstable (open-loop poles: $0,\ 0,\ -1.5594 \pm 2.29i$)

LQR and Linear MPC with Quadratic Cost

- Quadratic cost
- Linear system dynamics
- Linear constraints on inputs and states

LQR

$$J_\infty(x(t)) = \min \sum_{k=0}^{\infty} x_t^T Q x_t + u_k^T R u_k$$
$$\text{s.t. } x_{k+1} = A x_k + B u_k$$
$$x_0 = x(t)$$

MPC

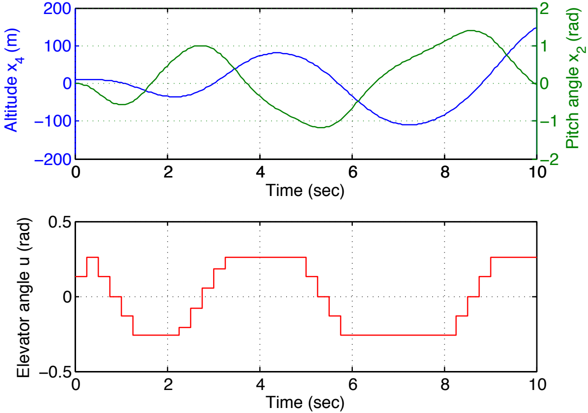
$$J_0^*(x(t)) = \min_{U_0} \sum_{k=0}^{N-1} {x_k}^T Q x_k + {u_k}^T R u_k$$
$$\text{s.t. } x_{k+1} = A x_k + B u_k$$
$$x_k \in \mathcal{X},\ u_k \in \mathcal{U}$$
$$x_0 = x(t)$$

Assume: $Q = Q^T \succeq 0,\ R = R^T \succ 0$

Example: LQR with saturation

Linear quadratic regulator with saturated inputs.

At time $t = 0$ the plane is flying with a deviation of $10m$ of the desired altitude, i.e. $x_0 = [0; 0; 0; 10]$



Problem parameters:

Sampling time 0.25sec, $Q = I$, $R = 10$

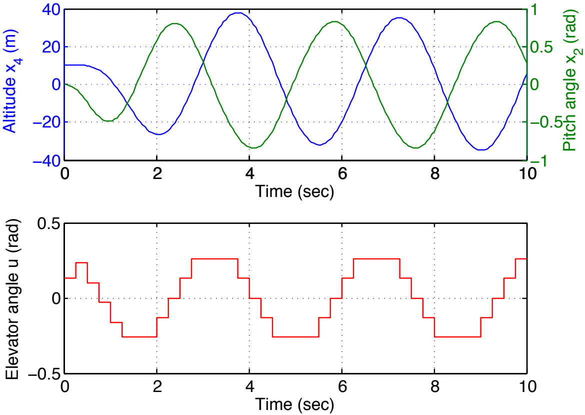
- Closed-loop system is unstable
- Applying LQR control and saturating the controller can lead to instability!

Example: MPC with Bound Constraints on Inputs

MPC controller with input constraints $|u_i| \leq 0.262$

Problem parameters:

$$\boxed{\begin{array}{l}\text{Sampling time } 0.25\text{sec},\\ Q = I,\ R = 10,\ N = 10\end{array}}$$



The MPC controller uses the knowledge that the elevator will saturate, but it does not consider the rate constraints.

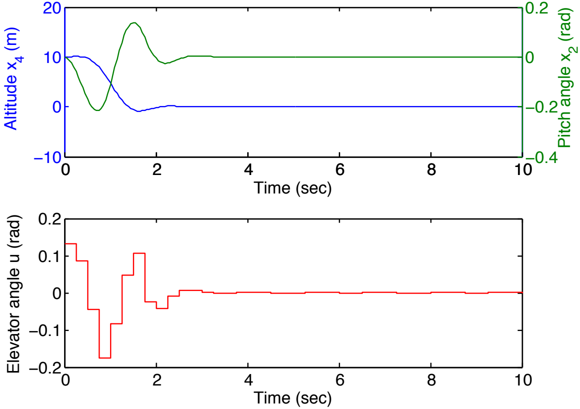
$\Rightarrow$ System does not converge to desired steady-state but to a limit cycle

Example: MPC with all Input Constraints

MPC controller with input constraints $|u_i| \leq 0.262$ and rate constraints $|\dot{u}_i| \leq 0.349$ approximated by $|u_k - u_{k-1}| \leq 0.349\, T_s$

Problem parameters:

$$\boxed{\begin{array}{l}\text{Sampling time } 0.25\text{sec},\\ Q = I,\ R = 10,\ N = 10\end{array}}$$



The MPC controller considers all constraints on the actuator

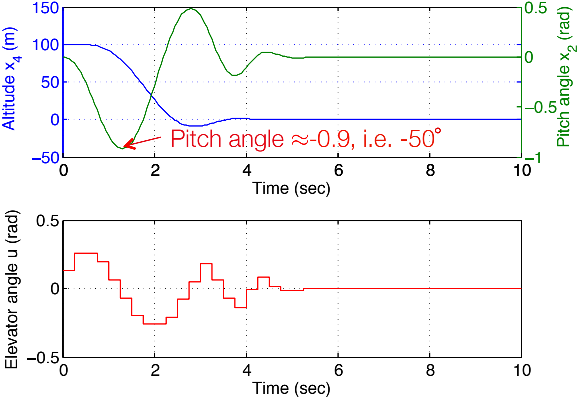
- Closed-loop system is stable
- Efficient use of the control authority

Example: Inclusion of state constraints

MPC controller with input constraints $|u_i| \leq 0.262$ and rate constraints $|\dot{u}_i| \leq 0.349$ approximated by $|u_k - u_{k-1}| \leq 0.349\,T_s$

Problem parameters:

Sampling time 0.25sec, $Q = I,\ R = 10,\ N = 10$



Increase step:
At time $t = 0$ the plane is flying with a deviation of 100m of the desired altitude, i.e. $x_0 = [0; 0; 0; 100]$

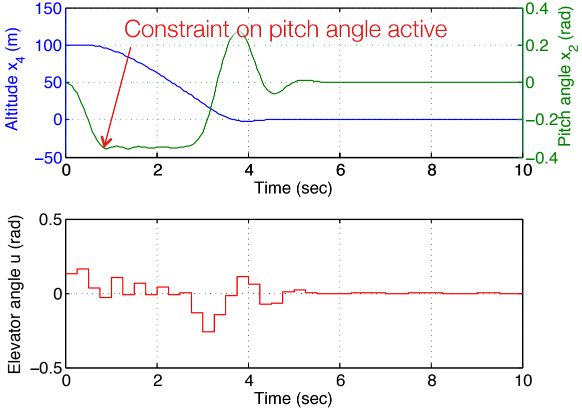
- Pitch angle too large during transient

Example: Inclusion of state constraints

MPC controller with input constraints $|u_i| \leq 0.262$ and rate constraints $|\dot{u}_i| \leq 0.349$ approximated by $|u_k - u_{k-1}| \leq 0.349\, T_s$

Problem parameters:

Sampling time 0.25sec, $Q = I$, $R = 10$, $N = 10$



Add state constraints for passenger comfort:

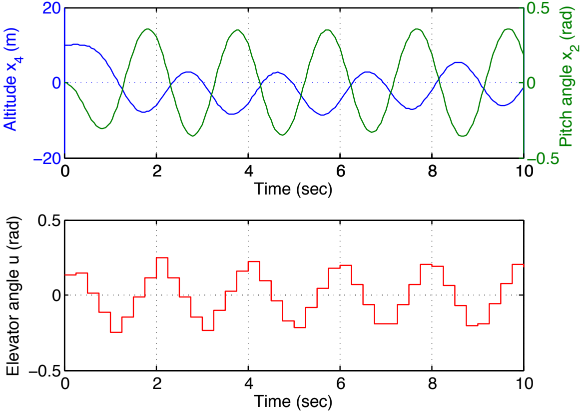
$$|x_2| \leq 0.349$$

Example: Short horizon

MPC controller with input constraints $|u_i| \leq 0.262$ and rate constraints $|\dot{u}_i| \leq 0.349$ approximated by $|u_k - u_{k-1}| \leq 0.349\,T_s$

Problem parameters:

Sampling time 0.25sec, $Q = I$, $R = 10$, $N = 4$



Decrease in the prediction horizon causes loss of the stability properties

Loss of Feasibility and Stability

What can go wrong with "standard" MPC?

- No feasibility guarantee, i.e., the MPC problem may not have a solution
- No stability guarantee, i.e., trajectories may not converge to the origin

Summary: Feasibility and Stability

- Infinite-Horizon
  If we solve the RHC problem for $N = \infty$ (as done for LQR), then the open loop trajectories are the same as the closed loop trajectories. Hence
  - If problem is feasible, the closed loop trajectories will be always feasible
  - If the cost is finite, then states and inputs will converge asymptotically to the origin

- Finite-Horizon
  RHC is "short-sighted" strategy approximating infinite horizon controller. But

  - **Feasibility**. After some steps the finite horizon optimal control problem may become infeasible. (Infeasibility occurs without disturbances and model mismatch!)
  - **Stability**. The generated control inputs may not lead to trajectories that converge to the origin.

Feasibility and stability in MPC - Solution

Main idea: Introduce terminal cost and constraints to explicitly ensure feasibility and stability:

$$
\begin{aligned}
J_0^*(x_0) = \quad & \min_{U_0} & & p(x_N) + \sum_{k=0}^{N-1} q(x_k, u_k) & & \text{Terminal Cost} \\
& \text{subj. to} & & & & \\
& & & x_{k+1} = Ax_k + Bu_k,\ k = 0, \ldots, N-1 & & \\
& & & x_k \in \mathcal{X},\ u_k \in \mathcal{U},\ k = 0, \ldots, N-1 & & \\
& & & x_N \in \mathcal{X}_f & & \text{Terminal Constraint} \\
& & & x_0 = x(t) & &
\end{aligned}
$$

$p(\cdot)$ and $\mathcal{X}_f$ are chosen to mimic an infinite horizon.

Choice of Terminal Set and Cost: Summary

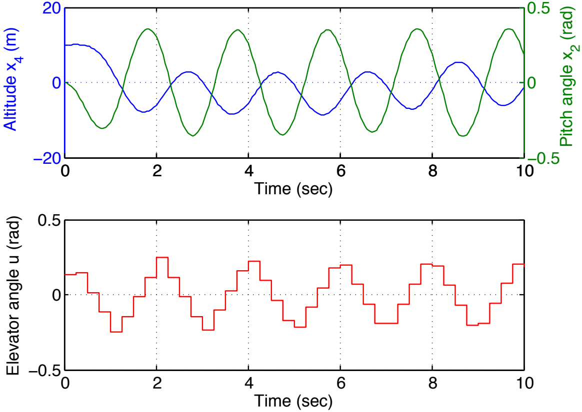
- Terminal constraint provides a sufficient condition for stability
- Region of attraction without terminal constraint may be larger than for MPC with terminal constraint but characterization of region of attraction extremely difficult
- $\mathcal{X}_f = 0$ simplest choice but small region of attaction for small N
- Solution for linear systems with quadratic cost
- In practice: Enlarge horizon and check stability by sampling
- With larger horizon length N, region of attraction approaches maximum control invariant set

Example: Short horizon

MPC controller with input constraints $|u_i| \leq 0.262$ and rate constraints $|\dot{u}_i| \leq 0.349$ approximated by $|u_k - u_{k-1}| \leq 0.349\,T_s$

Problem parameters:

Sampling time 0.25sec, $Q = I$, $R = 10$, $N = 4$



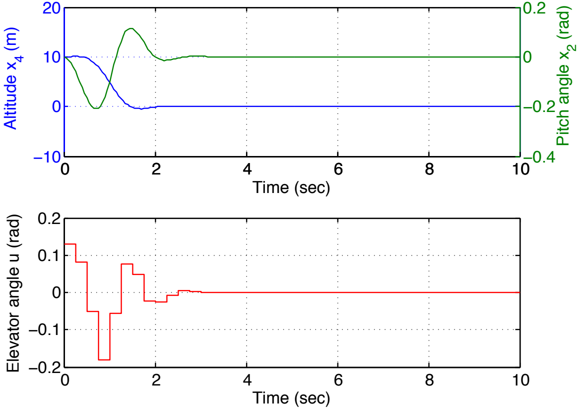
Decrease in the prediction horizon causes loss of the stability properties

Example: Short horizon

MPC controller with input constraints $|u_i| \leq 0.262$
and rate constraints $|\dot{u}_i| \leq 0.349$
approximated by $|u_k - u_{k-1}| \leq 0.349\,T_s$

Problem parameters:

Sampling time 0.25sec,
$Q = I$, $R = 10$, $N = 4$



Inclusion of terminal cost and constraint provides stability

Summary

Finite-horizon MPC may not be stable!

Finite-horizon MPC may not satisfy constraints for all time!

- An infinite-horizon provides stability and invariance.
- We 'fake' infinite-horizon by forcing the final state to be in an invariant set for which there exists an invariance-inducing controller, whose infinite-horizon cost can be expressed in closed-form.
- These ideas extend to non-linear systems, but the sets are difficult to compute.

Extension to Nonlinear MPC

Consider the nonlinear system dynamics: $x(t+1) = g(x(t), u(t))$

$$
\begin{array}{rll}
J_0^*(x(t)) = & \min\limits_{U_0} & p(x_N) + \sum\limits_{k=0}^{N-1} q(x_k, u_k) \\
& \text{subj. to} & x_{k+1} = g(x_k, u_k),\ \ k = 0, \ldots, N-1 \\
& & x_k \in \mathcal{X},\ \ u_k \in \mathcal{U},\ \ k = 0, \ldots, N-1 \\
& & x_N \in \mathcal{X}_f \\
& & x_0 = x(t)
\end{array}
$$

- Presented assumptions on the terminal set and cost did not rely on linearity
- Lyapunov stability is a general framework to analyze stability of nonlinear dynamic systems

$\rightarrow$ Results can be directly extended to nonlinear systems.

However, computing the sets $\mathcal{X}_f$ and function p can be very difficult!

MPC: Tracking, Soft Constraints, Move-Blocking

M. Morari, F. Borrelli*, C. Jones†

Institut für Automatik
ETH Zürich

*UC Berkeley

† EPFL

Spring Semester 2014
revised 29.04.2014

Table of Contents

Table of Contents

Tracking problem

Consider the linear system model

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{aligned}$$

Goal: Track given reference r such that $y_k \to r$ as $k \to \infty$.

Determine the steady state target condition x_s, u_s:

$$\begin{aligned} x_s &= Ax_s + Bu_s \\ Cx_s &= r \end{aligned} \qquad \Longleftrightarrow \qquad \begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix}$$

Table of Contents

Steady-state target problem

- In the presence of constraints: (x_s, u_s) has to satisfy state and input constraints.
- In case of multiple feasible u_s, compute 'cheapest' steady-state (x_s, u_s) corresponding to reference r:

$$\begin{aligned} \min \ & u_s^T R_s u_s \\ \text{s.t.} \ & \begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix} \\ & x_s \in \mathcal{X}, \quad u_s \in \mathcal{U}. \end{aligned}$$

- In general, we assume that the target problem is feasible
- If no solution exists: compute reachable set point that is 'closest' to r:

$$\begin{aligned} \min \ & (Cx_s - r)^T Q_s (Cx_s - r) \\ \text{s.t.} \ & x_s = Ax_s + Bu_s \\ & x_s \in \mathcal{X}, \quad u_s \in \mathcal{U}. \end{aligned}$$

RHC Reference Tracking

We now use control (MPC) to bring the system to a desired steady-state condition (x_s, u_s) yielding the desired output $y_k \to r$.

The MPC is designed as follows

$$\begin{array}{ll} \min\limits_{u_0,\dots,u_{N-1}} & \|y_N - Cx_s\|_P^2 + \sum\limits_{k=0}^{N-1} \|y_k - Cx_s\|_Q^2 + \|u_k - u_s\|_R^2 \\ \text{subj. to} & \text{model} \\ & \text{constraints} \\ & x_0 = x(t). \end{array}$$

Drawback: controller will show **offset** in case of unknown model error or disturbances.

Table of Contents

Soft Constraints: Motivation

- Input constraints are dictated by physical constraints on the actuators and are usually “hard”
- State/output constraints arise from practical restrictions on the allowed operating range and are **rarely hard**
- Hard state/output constraints always lead to *complications in the controller implementation*
  - Feasible operating regime is constrained even for stable systems
  - Controller patches must be implemented to generate reasonable control action when measured/estimated states move outside feasible range because of disturbances or noise
- In industrial implementations, typically, state constraints are **softened**

Table of Contents

Mathematical Formulation

- Original problem:

$$\begin{array}{ll} \min\limits_{z} & f(z) \\ \text{subj. to} & g(z) \leq 0 \end{array}$$

  Assume for now $g(z)$ is scalar valued.

- "Softened" problem:

$$\begin{array}{ll} \min\limits_{z,\epsilon} & f(z) + l(\epsilon) \\ \text{subj. to} & g(z) \leq \epsilon \\ & \epsilon \geq 0 \end{array}$$

Requirement on $l(\epsilon)$

If the original problem has a feasible solution z^*, then the softened problem should have the same solution z^*, and $\epsilon = 0$.

Note: $l(\epsilon) = v \cdot \epsilon^2$ does not meet this requirement for any $v > 0$ as demonstrated next.

Main Result

Theorem (Exact Penalty Function)

$l(\epsilon) = u \cdot \epsilon$ satisfies the requirement for any $u > u^* \geq 0$, where u^* is the optimal Lagrange multiplier for the original problem.

- **Disadvantage:** $l(\epsilon) = u \cdot \epsilon$ renders the cost non-smooth.
- Therefore in practice, to get a smooth penalty, we use

$$l(\epsilon) = u \cdot \epsilon + v \cdot \epsilon^2$$

  with $u > u^*$ and $v > 0$.
- Extension to multiple constraints $g_j(z) \leq 0,\ j = 1, \ldots, r$:

$$l(\epsilon) = \sum_{j=1}^{r} u_j \cdot \epsilon_j + v_j \cdot \epsilon_j^2 \qquad (1)$$

  where $u_j > u_j^*$ and $v_j > 0$ can be used to weight violations (if necessary) differently.

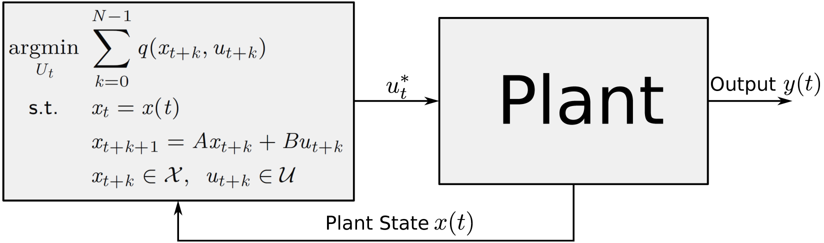
Explicit Model Predictive Control

F. Borrelli, C. Jones, M. Morari

UC Berkeley, EPFL, ETHZ

Spring Semester 2014
revised 29.04.2014

Introduction



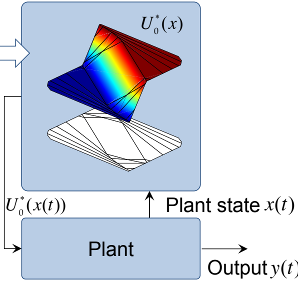
- Requires at each time step on-line solution of an optimization problem

Introduction

OFFLINE

$$U_0^*(x(t)) = \text{argmin} \;\; x_N^T P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k$$
$$\text{subj. to} \;\; x_0 = x(t)$$
$$x_{k+1} = A x_k + B u_k, \;\; k = 0, \ldots, N-1$$
$$x_k \in \mathcal{X}, \;\; u_k \in \mathcal{U}, \;\; k = 0, \ldots, N-1$$
$$x_N \in \mathcal{X}_f$$

ONLINE



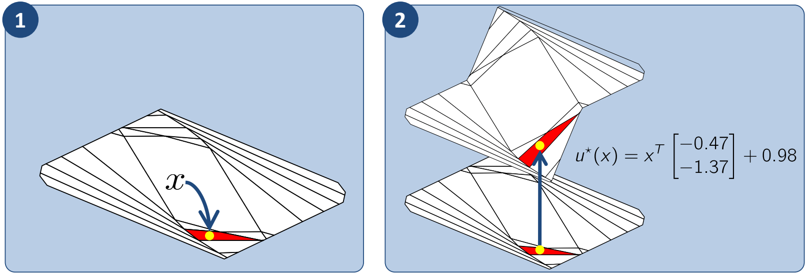
- Optimization problem is parameterized by state
- Pre-compute control law as function of state x
- Control law is piecewise affine for linear system/constraints

Result: Online computation dramatically reduced and *real-time*
Tool: *Parametric programming*

Online evaluation: Point location

Calculation of piecewise affine function:

1. Point location
2. Evaluation of affine function



Real-time MPC Software Toolbox

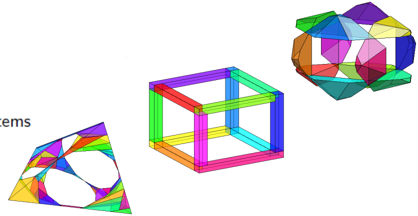
Formal specification
- YALMIP
- HYSDEL
- Linear + Hybrid models

Control law
- Explicit MPC
- Fixed-complexity solutions

Software synthesis
- Real-time workshop
- Bounded-time solvers
- Verifiable code generation

Verified controller

Multi-Parametric Toolbox (MPT)
- Computational geometry
- Multi-parametric programming
- Control of linear and hybrid systems



Hybrid Model Predictive Control

F. Borrelli*, M. Morari, C. Jones†

Institut für Automatik
ETH Zürich

*UC Berkley

† EPFL

Spring Semester 2014

Introduction

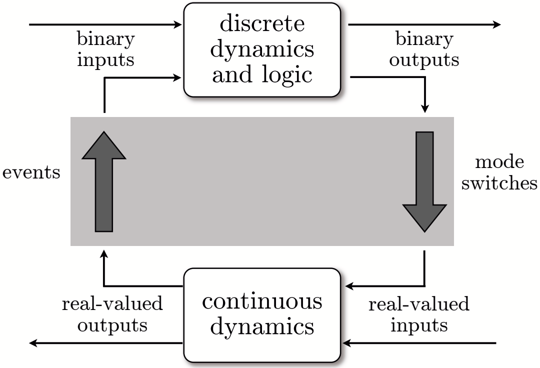
Up to this point: Discrete-time linear systems with linear constraints.

We now consider MPC for systems with

1. **Continuous dynamics**: described by one or more difference (or differential) equations; states are continuous-valued.
2. **Discrete events**: state variables assume *discrete* values, e.g.
   - binary digits $\{0, 1\}$,
   - $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\ldots$
   - finite set of symbols

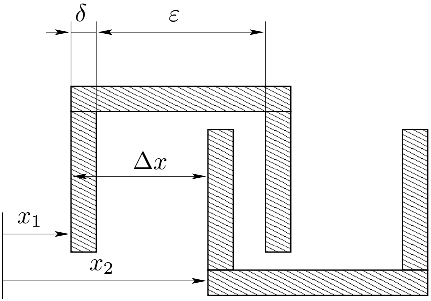
Hybrid systems: Dynamical systems whose state evolution depends on an interaction between continuous dynamics and discrete events.

Introduction



Hybrid systems: Logic-based discrete dynamics and continuous dynamics interact through events and mode switches

Mechanical System with Backlash

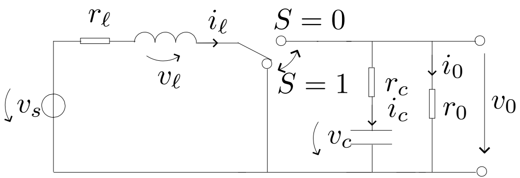


- **Continuous dynamics**: states x_1, x_2, $\dot{x}_1$, $\dot{x}_2$.
- **Discrete events**:
  a) *"contact mode"* $\Rightarrow$ mechanical parts are in contact and the force is transmitted. Condition:

$$[(\Delta x = \delta) \wedge (\dot{x}_1 > \dot{x}_2)] \bigvee [(\Delta x = \varepsilon) \wedge (\dot{x}_2 > \dot{x}_1)]$$

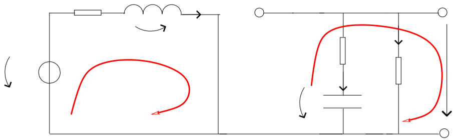
  b) *"backlash mode"* $\Rightarrow$ mechanical parts are not in contact

DCDC Converter

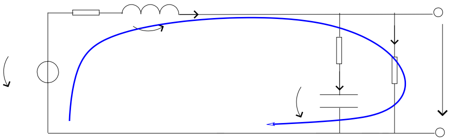


- **Continuous dynamics**: states v_ℓ, i_ℓ, v_c, i_c, v_0, i_0
- **Discrete events**: $S = 0$, $S = 1$

Mode 1 ($S = 1$)



Mode 2 ($S = 0$)



Mixed Logical Dynamical Systems

Goal: Describe hybrid system in form compatible with optimization software:

- continuous and boolean variables
- linear equalities and inequalities

Idea: associate to each Boolean variable p_i a binary integer variable δ_i:

$$p_i \Leftrightarrow \{\delta_i = 1\}, \quad \neg p_i \Leftrightarrow \{\delta_i = 0\}$$

and embed them into a set of constraints as **linear integer inequalities**.

Two main steps:

1. Translation of Logic Rules into Linear Integer Inequalities
2. Translation continuous and logical components into Linear Mixed-Integer Relations

Final result: a compact model with linear equalities and inequalities involving real and binary variables

MLD Hybrid Model

A DHA can be converted into the following MLD model

$$\begin{aligned} x_{t+1} &= Ax_t + B_1 u_t + B_2 \delta_t + B_3 z_t \\ y_t &= Cx_t + D_1 u_t + D_2 \delta_t + D_3 z_t \\ E_2 \delta_t + E_3 z_t &\leq E_4 x_t + E_1 u_t + E_5 \end{aligned}$$

where $x \in \mathbb{R}^{n_c} \times \{0,1\}^{n_\ell}$, $u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_\ell}$ $y \in \mathbb{R}^{p_c} \times \{0,1\}^{p_\ell}$, $\delta \in \{0,1\}^{r_\ell}$ and $z \in \mathbb{R}^{r_c}$.

Physical constraints on continuous variables:

$$\mathcal{C} = \left\{ \begin{bmatrix} x_c \\ u_c \end{bmatrix} \in \mathbb{R}^{n_c+m_c} \;\middle|\; Fx_c + Gu_c \leq H \right\}$$

HYbrid System DEscription Language

HYSDEL

- based on DHA
- enables description of discrete-time hybrid systems in a compact way:
  - automata and propositional logic
  - continuous dynamics
  - A/D and D/A conversion
  - definition of constraints
- automatically **generates MLD models** for MATLAB
- freely available from:

`http://control.ee.ethz.ch/~hybrid/hysdel/`

Optimal Control for Hybrid Systems: General Formulation

Consider the CFTOC problem:

$$J^*(x(t)) = \min_{U_0} \; p(x_N) + \sum_{k=0}^{N-1} q(x_k, u_k, \delta_k, z_k),$$

$$\text{s.t.} \left\{ \begin{array}{l} x_{k+1} = Ax_k + B_1 u_k + B_2 \delta_k + B_3 z_k \\ \\ E_2 \delta_k + E_3 z_k \leq E_4 x_k + E_1 u_k + E_5 \\ \\ x_N \in \mathcal{X}_f \\ x_0 = x(t) \end{array} \right.$$

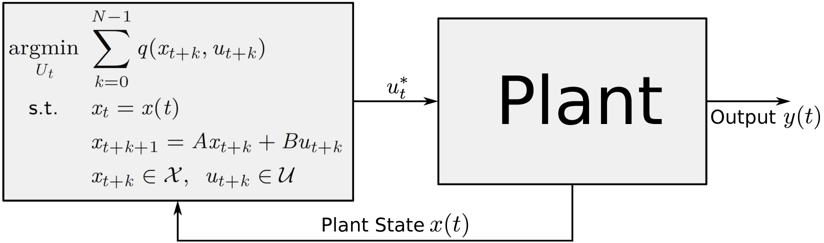
where $x \in \mathbb{R}^{n_c} \times \{0,1\}^{n_b}$, $u \in \mathbb{R}^{m_c} \times \{0,1\}^{m_b}$, $y \in \mathbb{R}^{p_c} \times \{0,1\}^{p_b}$, $\delta \in \{0,1\}^{r_b}$ and $z \in \mathbb{R}^{r_c}$ and

$$U_0 = \{u_0, u_1, \ldots, u_{N-1}\}$$

Mixed Integer Optimization

Model Predictive Control of Hybrid Systems

MPC solution: Optimization in the loop



As for linear MPC, at each sample time:

- Measure / estimate current state $x(t)$
- Find the optimal input sequence for the entire planning window N:
  $U_t^* = \{u_t^*, u_{t+1}^*, \ldots, u_{t+N-1}^*\}$
- Implement only the *first* control action u_t^*
- **Key difference: Requires online solution of an MILP or MIQP**