Introduction to Model Predictive Control

Amazing tour around MPC (in just 45 minutes!)

Carlos Bordons
University of Seville, Spain

Industry Workshop on HD-MPC. 24th June 2011. Leuven

Outline

1. **Introduction**
2. Basic ideas of MPC
3. Some algorithms
4. MPC and Constraints
5. MPC for “fast” systems
6. Other formulations of MPC
7. Stability
8. Some applications
9. Conclusions and open issues

What is Model Predictive Control?

MPC is a form of control in which the current control action is obtained by ***solving on-line,*** at each sampling instant, a ***finite-horizon open-loop optimal*** control problem, using the current state of the plant as the initial state; the optimization yields an optimal control sequence ***and the first control in this sequence*** is applied to the plant.

Optimization over a future receding horizon using a dynamic model of the plant

1. Introduction

- Current trends in automatic control
- Past: stable operation. Replace the human operator.
- Present: operate processes according to the market. High efficiency and flexibility. New challenges.
- Objective: act on the manipulated variables in order to satisfy changing operating criteria:
  - Profit
  - Yield
  - Safety
  - Energy saving
  - Environment
  - Quality

MPC successful in industry.

- Many and very diverse and successful applications:
  - Refining, petrochemical, polymers, semiconductor production scheduling, air traffic control, clinical anesthesia, power converters, etc.
- Many MPC vendors.
- Most general and intuitive way of posing the control problem in the **time domain**. Integrates:
  - Optimal control
  - Stochastic control
  - Mutivariable control
  - Constraints
  - Measurable disturbances
  - Nonlinear processes, etc.

Successful in industry

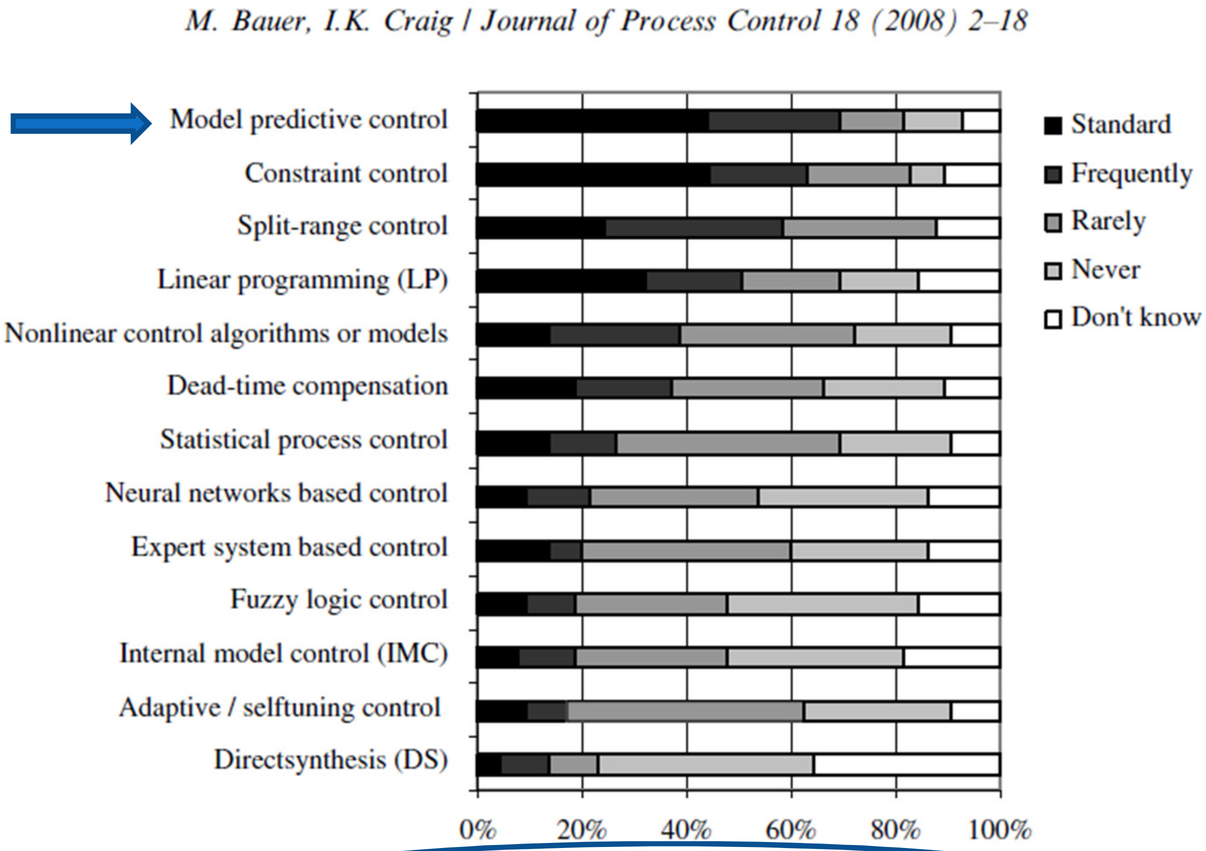


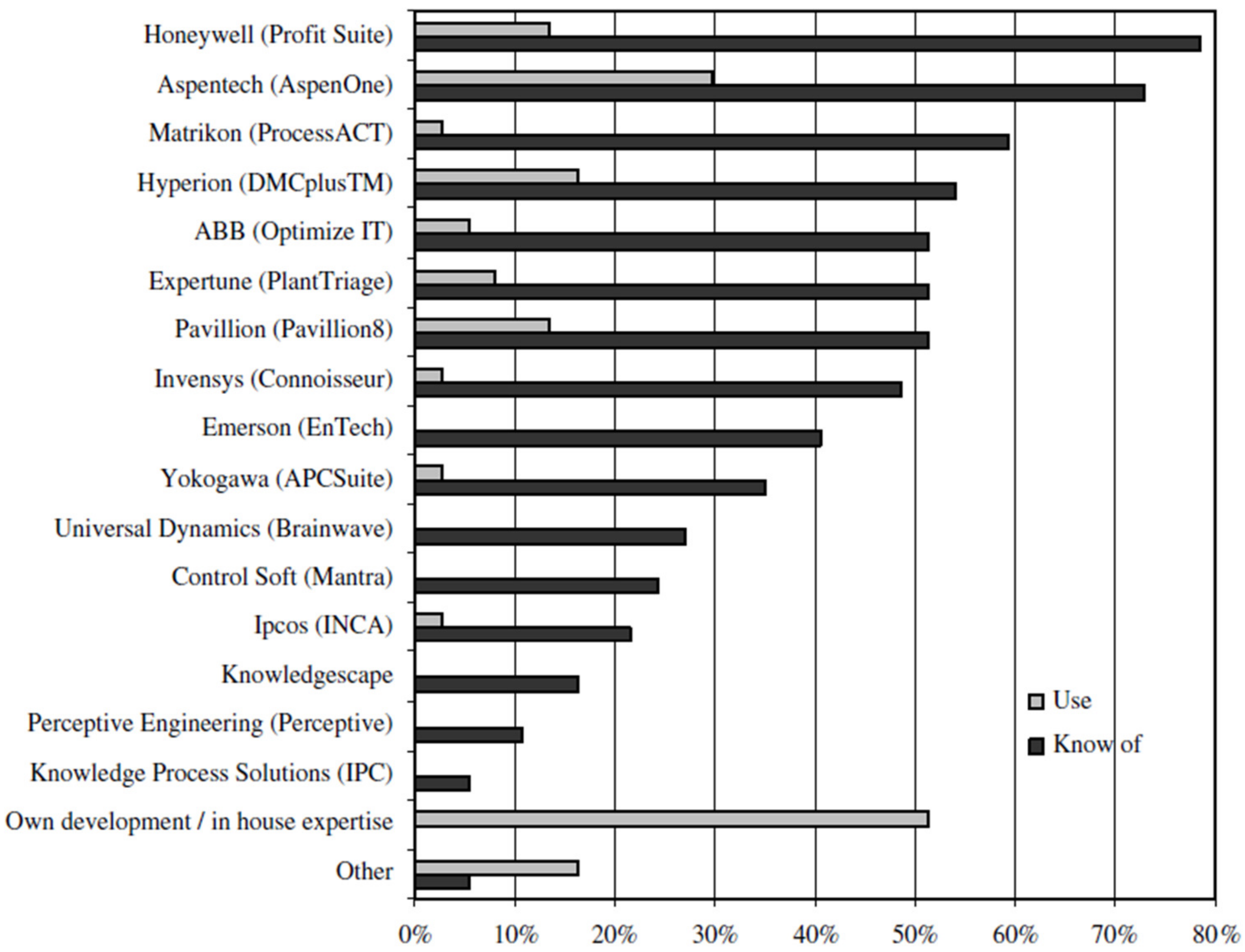
Fig. 4. Industrial use of APC methods: survey results.

Reasons for success in industry:

- On-line optimization
- Multi-level control
- Constraint management

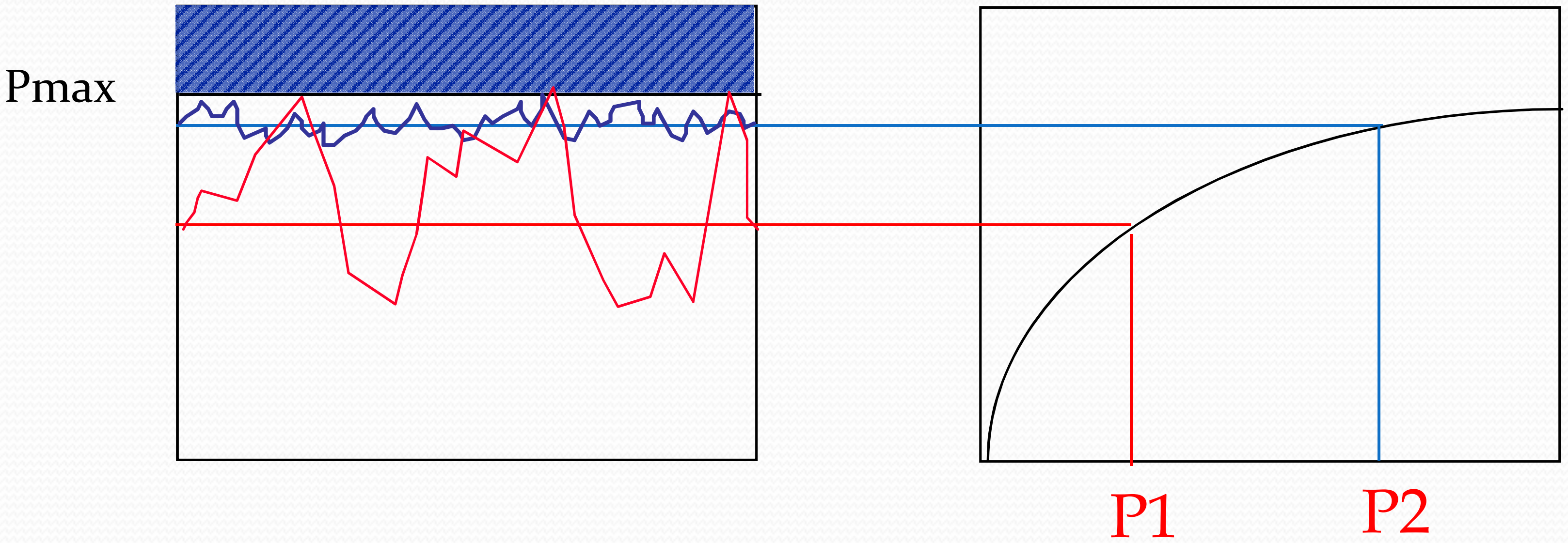
Economic assessment of advanced process control – A survey and framework. M. Bauer & I. K. Craig. Journal of Process Control 18 (2008) pp 2–18.

Most APC products are based on MPC



Real reason of success: Economics

- MPC can be used to **optimize operating points** (economic objectives). Optimum usually at the intersection of a set of **constraints**.
- Obtaining smaller variance and taking constraints into account allow to **operate closer to constraints (and optimum)**
- Repsol reported 2-6 months payback periods for new MPC applications.

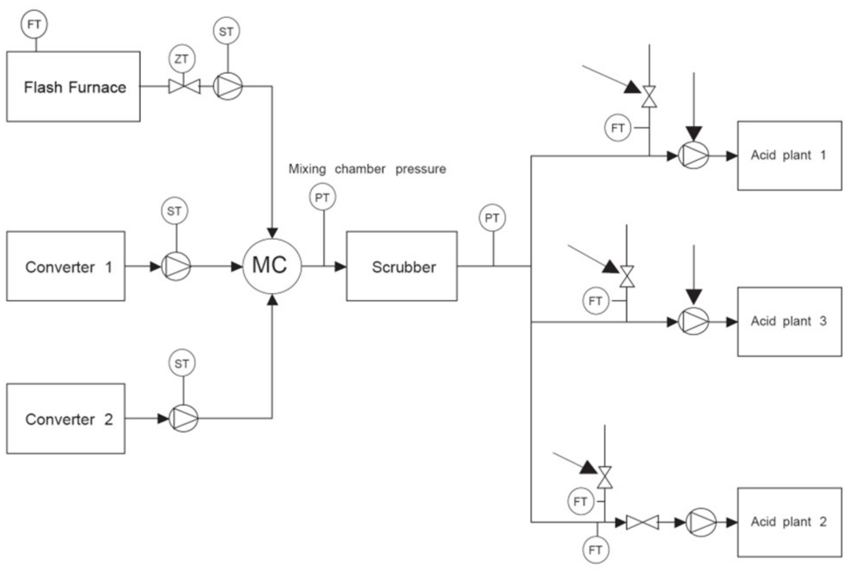


MPC successful in Academia

- Many MPC sessions in control conferences and control journals, MPC workshops.
- Science Direct: 2,101 journal papers in the last decade. 265 in 2010. 250 in 2011
- IEEExplore: 1,194 journals and 11,829 conference papers

Motivating example: mixing chamber in a copper smelter

Energy saving & automatic mode



Pressure control in the mixing chamber
(below atmosphere to avoid emissions)

Objectives:

Minimize environmental impact (emissions)

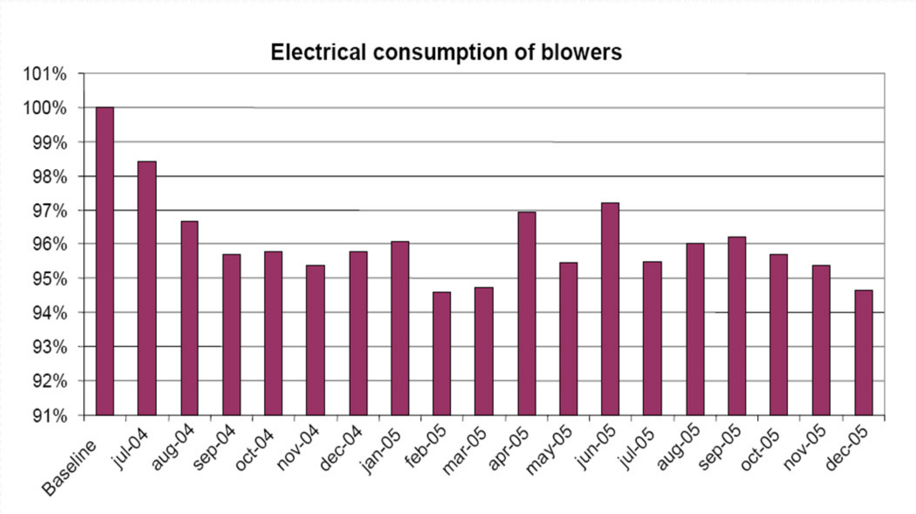
Reduce energy consumption in blowers

Results:

- Increment in the time of automatic operation (from 27% to 84%)
- Energy saving

Yearly savings: 1.900 MWh.

Average reduction: 4.22 %



C. Bordons, M.R. Arahal, E.F. Camacho y J.M. Tejera. *“Energy saving in a copper smelter by means of Model Predictive Control.”*

Is it so much successful in other engineering fields?

- Automotive: Cruise control, power train, power management
- Flight control
- Spacecraft control (attitude, rendez-vous…)
- Anaesthesia, Diabetes
- Energy in buildings
- Lighting
- Distribution networks: smart grids, canals, water distribution networks
- Greenhouses
- Etc.

Outline

1. Introduction
2. **Basic ideas of MPC**
3. Some algorithms
4. MPC and Constraints
5. MPC for “fast” systems
6. Other formulations of MPC
7. Stability
8. Some applications
9. Conclusions and open issues

2. Basic ideas of MPC

- Common features of model predictive controllers:
  - Explicit use of a **dynamic model** to predict the system evolution in the future (horizon)
  - Computation of the control signal **minimizing a cost function**
  - Use of a **sliding strategy**: the horizon is moving towards the future. Only the first element of the sequence is sent to the plant.

- Algorithms differ in the kind of model, the cost function and the optimization procedure.

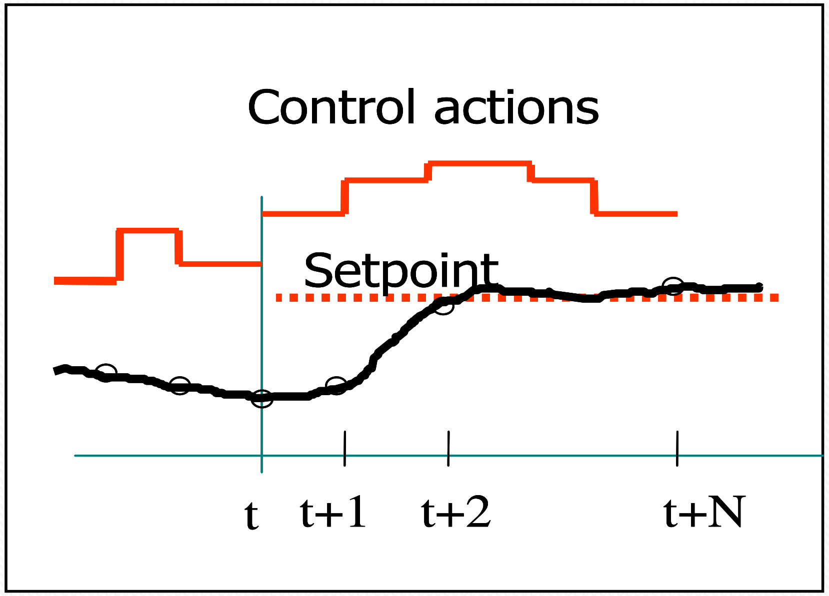
Controller strategy

1.- At each samplig instant k, *y(t+k/t)* are computed for a certain horizon *N*, as functions of the control actions *u(t+k/t)*

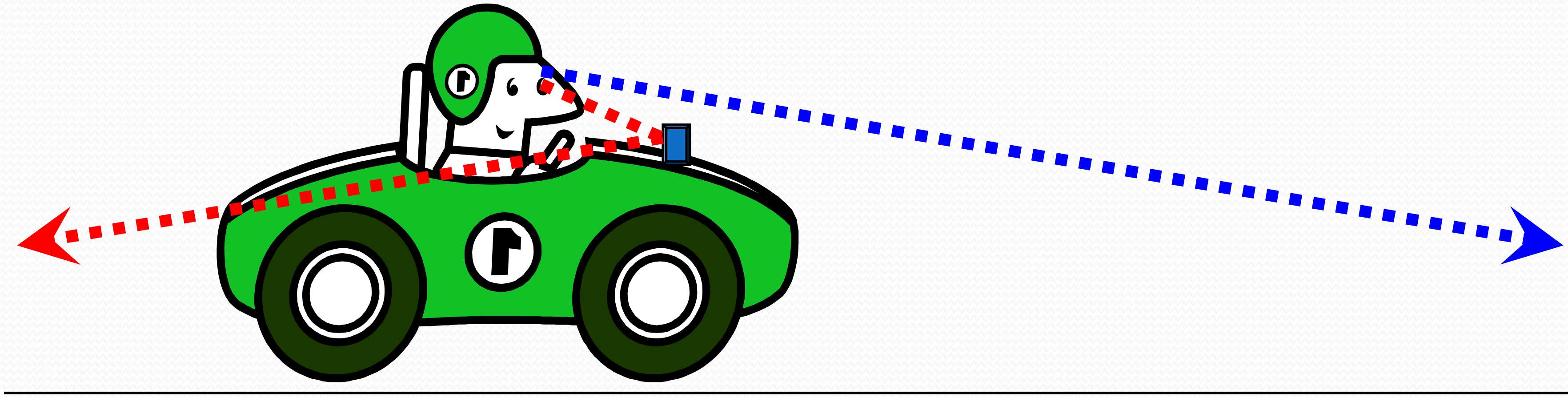
2.- Control actions are computed minimizing a cost function (tracking)

3.- The first control action *u(t)* is used (the rest is neglected)

4.- Go to 1 with the new measured value *y(t+1)*



Controller strategy. MPC vs. PID



PID: $u(t)=u(t-1)+g_0\, e(t) + g_1\, e(t-1) + g_2\, e(t-2)$

Pros & Cons

Advantages:

- Intuitive concepts (optimization, horizon, etc.)
- Used for different types of processes (unstable, dead-time, etc.)
- Easy extension to the multivariable case
- Constraints handling
- Open methodology that allows future extensions

Drawbacks:

- Need of an appropriate **dynamic model** of the plant
- **Computational** burden

Outline

3. Some algorithms

- MPC is a "**family**" of control algorithms
- The history of MPC is **originated from application and then expanded to theoretical** field (ordinary control algorithms often has applications after sufficient theoretical research)
- Based on theoretical research in the 60' and optimization needs in the process industry in late 70'
  - Kalman: LQG (1960)
  - Propoi: use of LP methods (1963)

Family tree

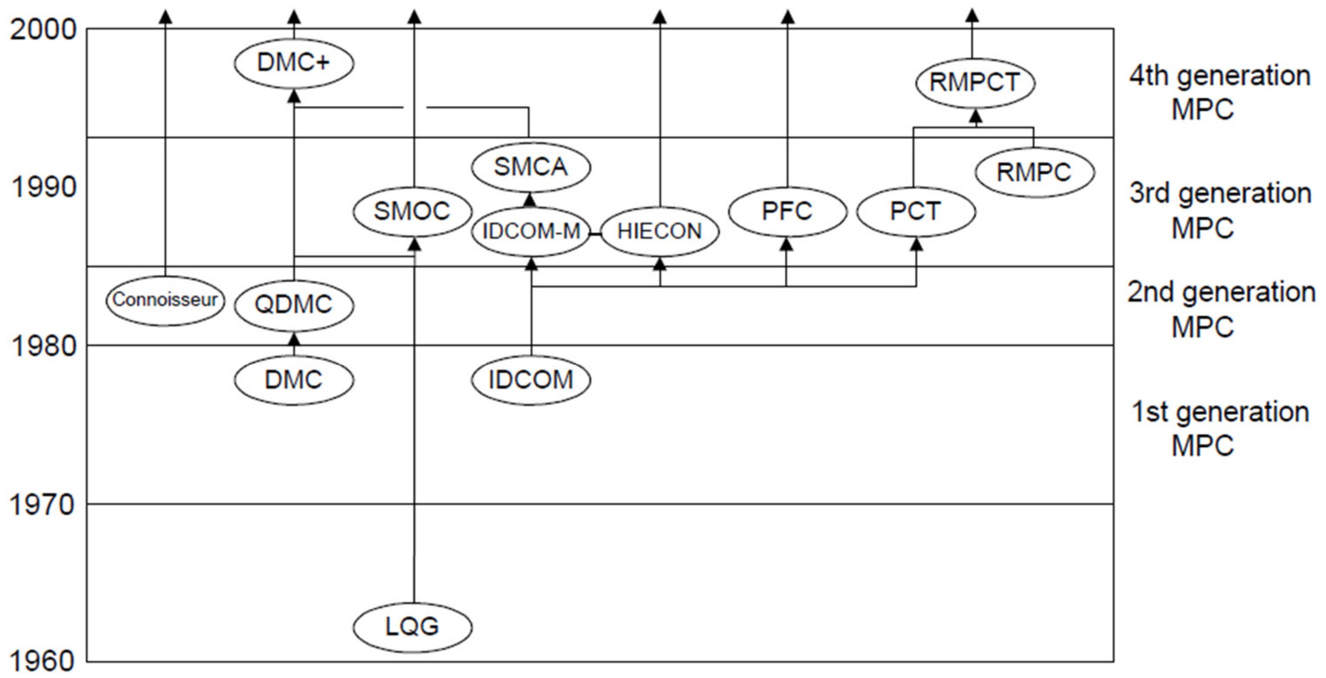


Fig. 1. Approximate genealogy of linear MPC algorithms.

- Richalet *et al*, Model Predictive Heuristic Control (MPHC) IDCOM (1976, 1978. PFC 90'
- Cutler & Ramaker, DMC (1979, 1980)
- Cutler *et al* QDMC (QP+DMC) (1983)
- Clarke *et al* GPC (1987)
- Commercial extensions of these methods

Dynamic Matrix Control (DMC)

[Cutler & Ramaker, 1980]

- **Step Response Model**. Great interest in industry:
  - Easy experimental identification
  - Scarce process knowledge required

$$y(t) = \sum_{i=1}^{\infty} g_i \triangle u(t-i)$$

- Prediction along the horizon:

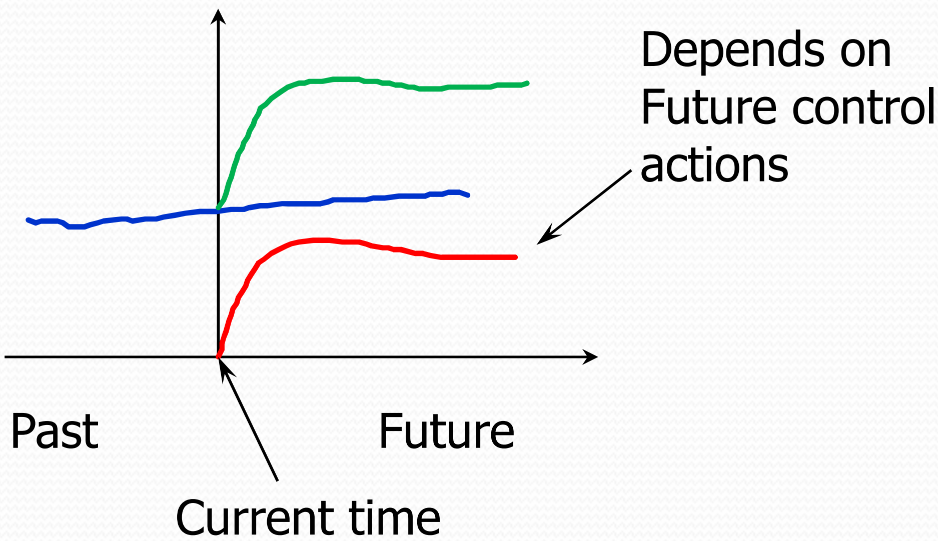
y = Gu + f

Common to all linear MPC

Free response

+

Forced response



DMC. Resolution

- Dynamic Matrix:

$$\mathbf{G} = \begin{bmatrix} g_1 & 0 & \cdots & 0 \\ g_2 & g_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_m & g_{m-1} & \cdots & g_1 \\ \vdots & \vdots & \ddots & \vdots \\ g_p & g_{p-1} & \cdots & g_{p-m+1} \end{bmatrix}$$

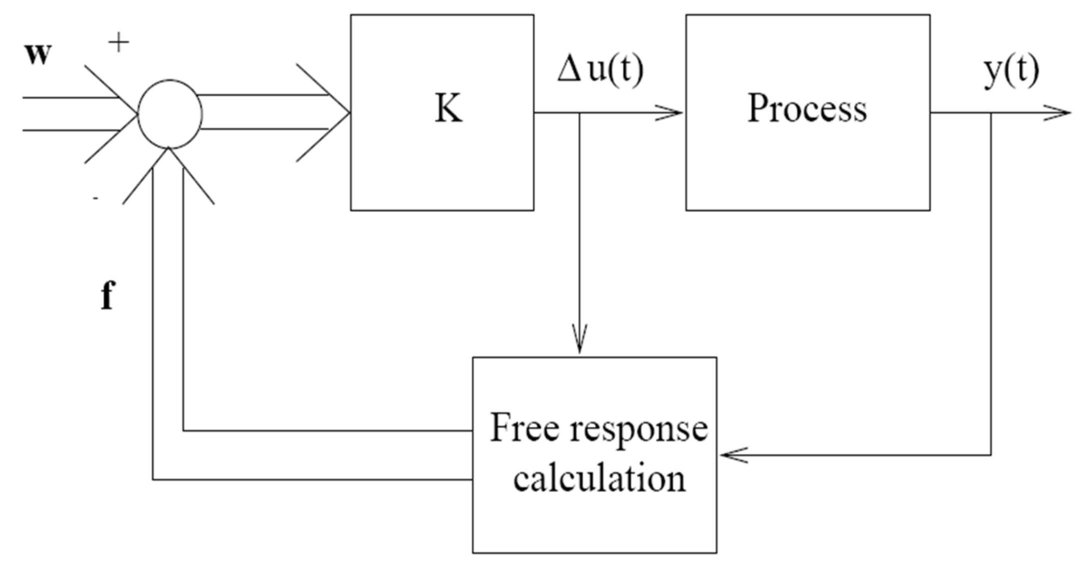
- Objective Function:

$$J = \sum_{j=1}^{p} [\hat{y}(t+j \mid t) - w(t+j)]^2 + \sum_{j=1}^{m} \lambda [\triangle u(t+j-1)]^2$$

- Solution to the **unconstrained** minimization:

$$\mathbf{u} = (\mathbf{G}^T \mathbf{G} + \lambda I)^{-1} \mathbf{G}^T (\mathbf{w} - \mathbf{f})$$

DMC. Control Law



$$\triangle u(t) = \mathbf{K}(\mathbf{w} - \mathbf{f})$$

Linear Control Law

Proportional to the **future error** (unlike PID)

Generalized Predictive Control (GPC)

[Clarke et al., 1987]

- CARIMA Model **transfer function** (valid for unstable processes)

$$A(q^{-1})y_t = B(q^{-1})u_{t-1} + \frac{C(q^{-1})}{\Delta}\xi_t$$

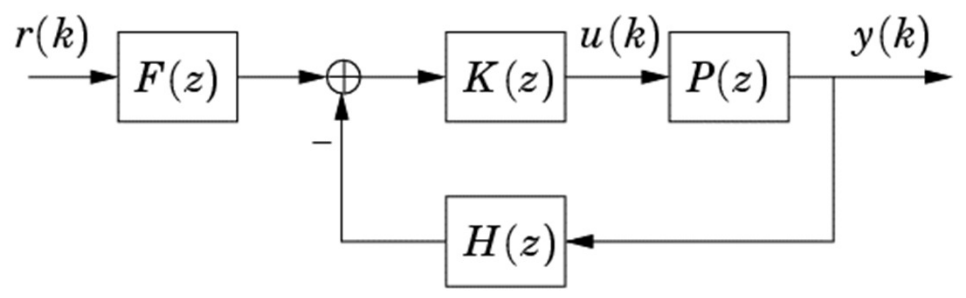
- Cost function like DMC

$$J(u,t) = \sum_{j=N_1}^{N_2} [y_{t+j} - r_{t+j}]^2 + \sum_{j=1}^{N_u} \lambda[\Delta u_{t+j-1}]^2$$

- Control law:

$$u = (G^T G + \lambda I)^{-1} G^T (r - f)$$

- Can be formulated as a 2DOF controller



State Space

- State-space model for predictions

$$\overline{x}(t+1) = M\overline{x}(t) + N \triangle u(t)$$
$$y(t) = Q\overline{x}(t)$$

$$\mathbf{y} = \begin{bmatrix} \hat{y}(t+1|t) \\ \hat{y}(t+2|t) \\ \vdots \\ \hat{y}(t+N_2|t) \end{bmatrix} = \begin{bmatrix} QM\hat{x}(t) + QN \triangle u(t) \\ QM^2\hat{x}(t) + \sum_{i=0}^{1} QM^{1-i}N \triangle u(t+i) \\ \vdots \\ QM^{N_2}\hat{x}(t) + \sum_{i=0}^{N_2-1} QM^{N_2-1-i}N \triangle u(t+i) \end{bmatrix}$$

- Free and forced response

$$\mathbf{y} = \mathbf{F}\hat{x}(t) + \mathbf{H}\mathbf{u}$$

$$J = (\mathbf{H}\mathbf{u} + \mathbf{F}\hat{x}(t) - \mathbf{w})^T(\mathbf{H}\mathbf{u} + \mathbf{F}\hat{x}(t) - \mathbf{w}) + \lambda\mathbf{u}^T\mathbf{u}$$

- Control law

$$\mathbf{u} = (\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{H}^T(\mathbf{w} - \mathbf{F}\hat{x}(t))$$

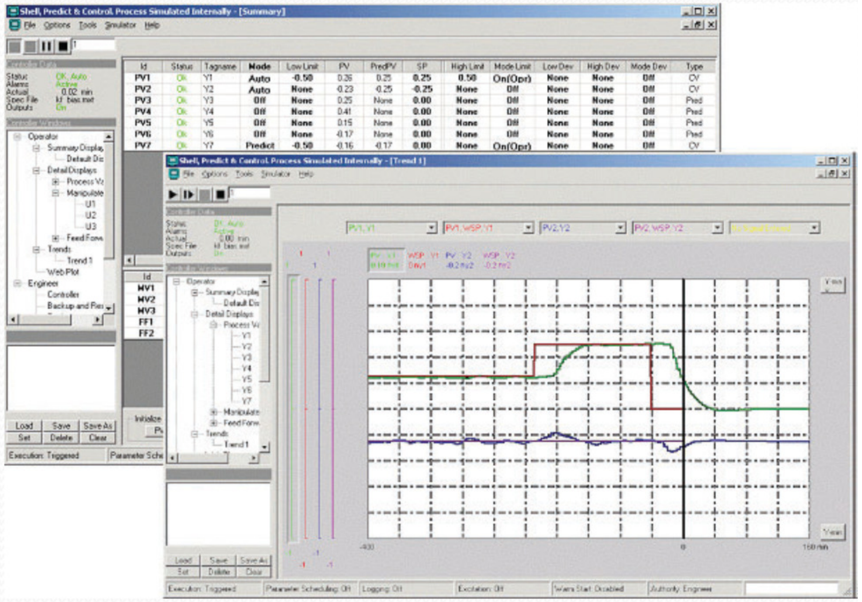
State Space

- Advantages:
  - Very useful for theoretical analysis (stability)
  - Straightforward extension to the multivariable case
  - Very appropriate for nonlinear systems

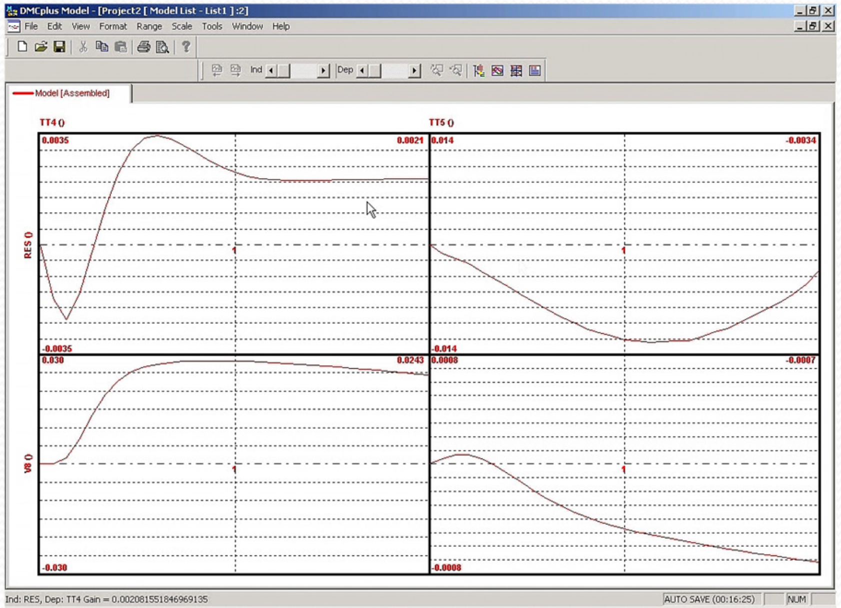
- Drawbacks:
  - May need an observer (Kalman filter)
  - Dead times difficult to handle

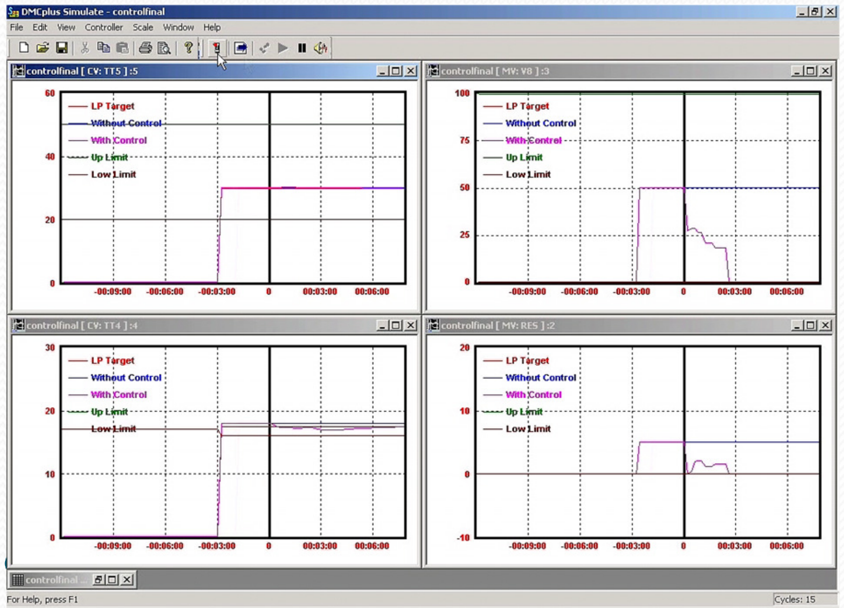
Some market products

OPTIMIZE-IT (ABB)

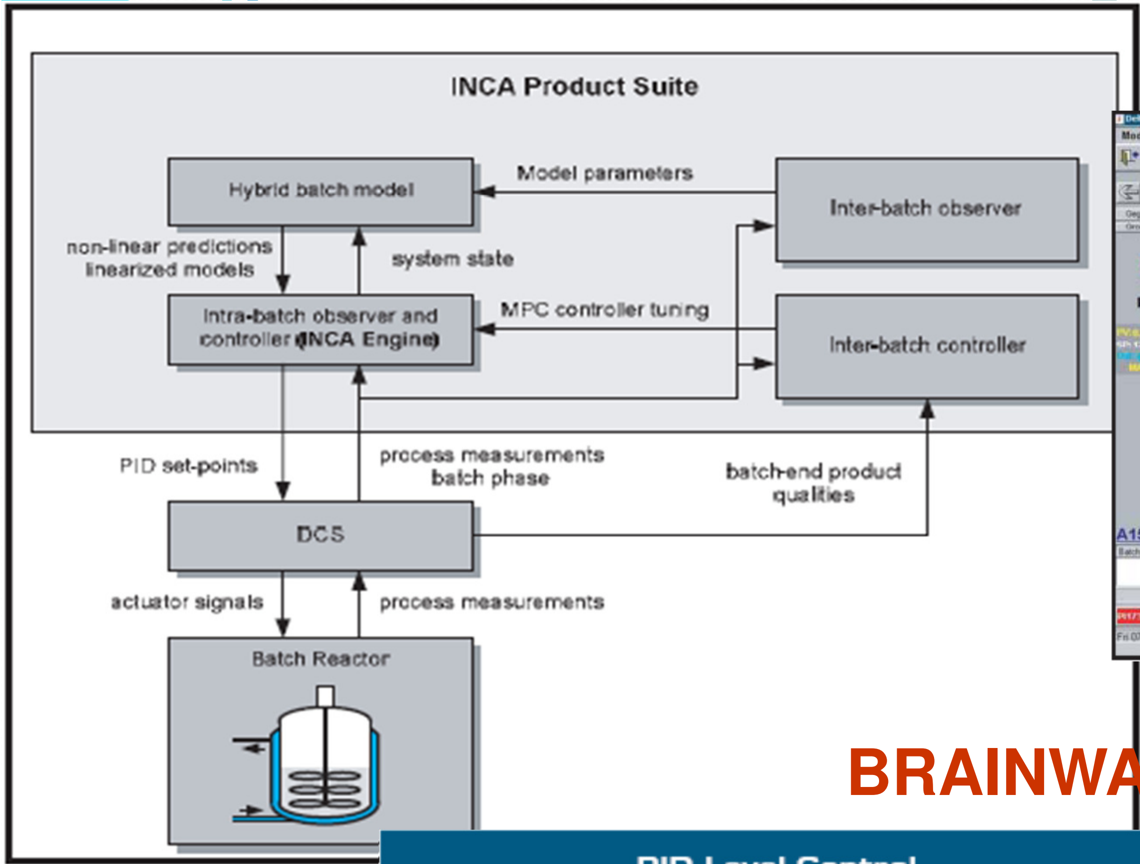


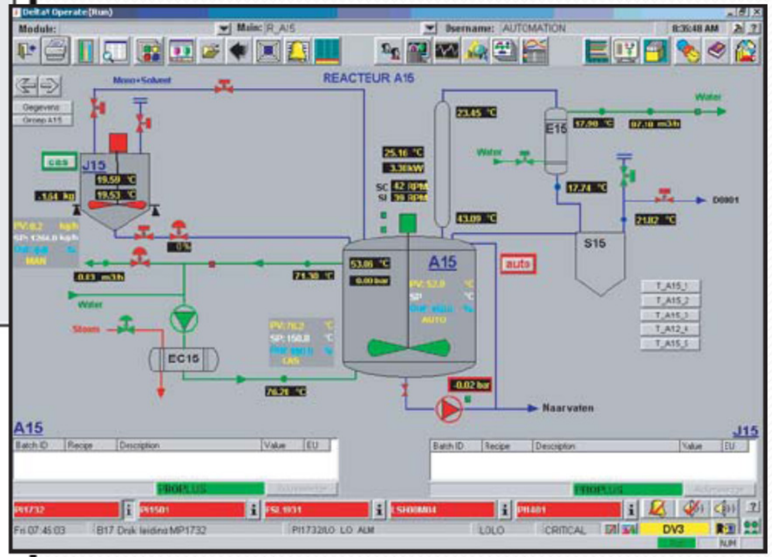
DMCplus (ASPENTECH)



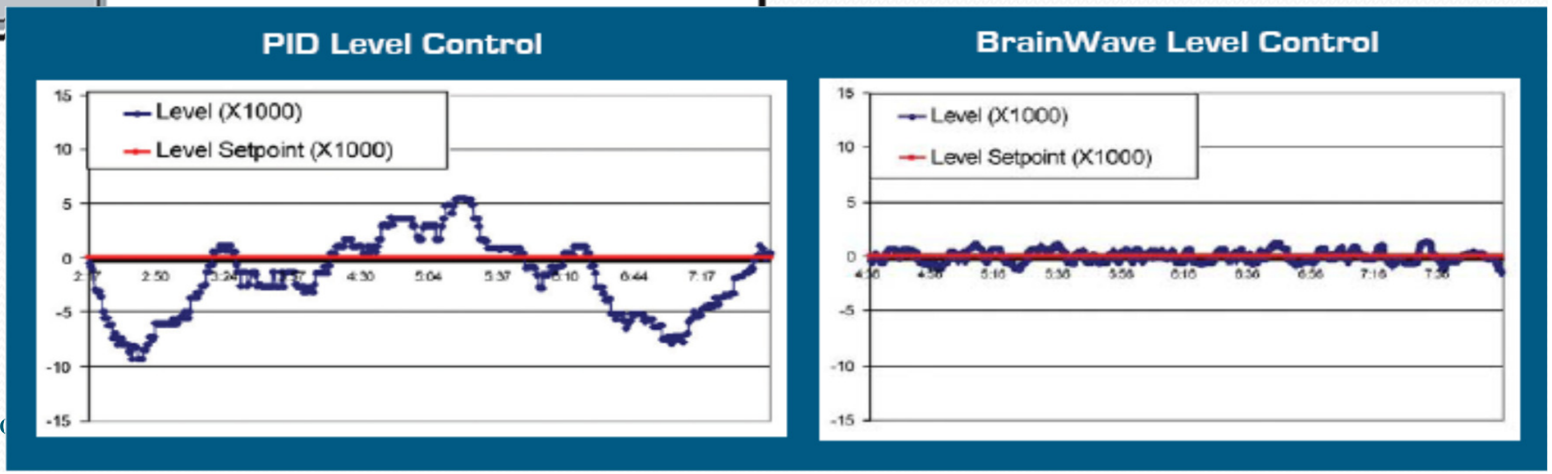


INCA for BATCH (IPCOS)





BRAINWAVE (ANDRITZ)



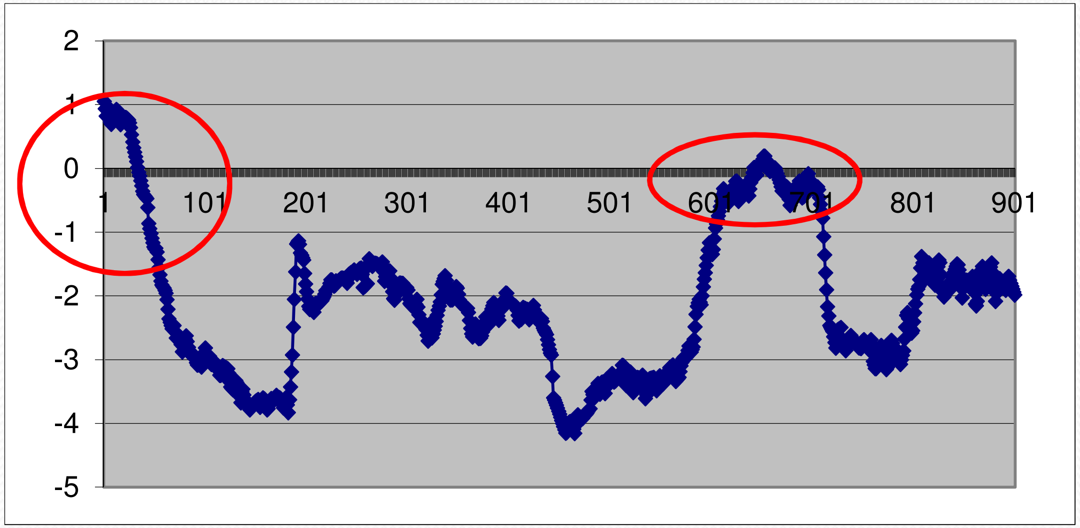
Outline

1. Introduction
2. Basic ideas of MPC
3. Some algorithms
4. **MPC and Constraints**
5. MPC for “fast” systems
6. Other formulations of MPC
7. Stability
8. Some applications
9. Conclusions and open issues

4. MPC and Constraints

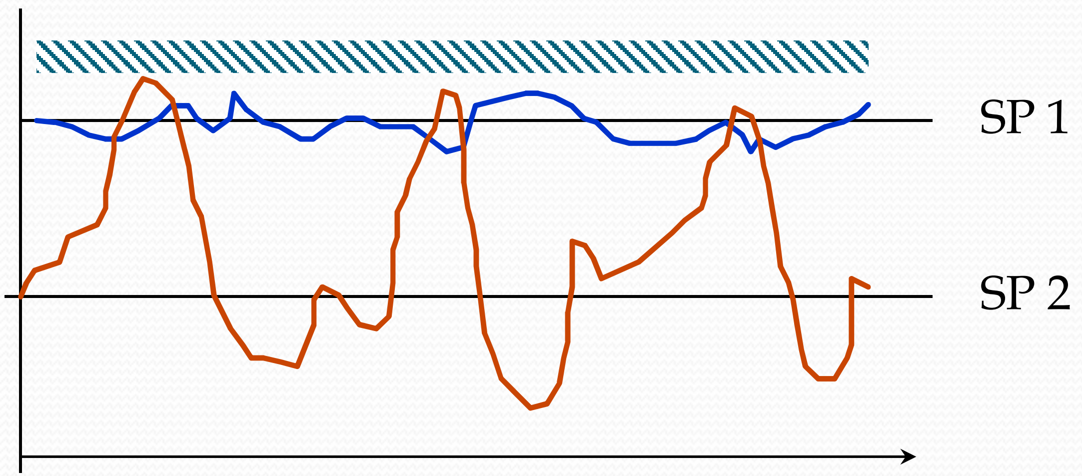
In practice, all processes are subject to constraints

- **Actuators** have a limited range (valves, pumps, etc)
- **Safety limits:** maximum pressure or temperature
- **Technological requirements** (temperature profile)
- Product **Quality** or **environmental** regulations



Contraints in process control

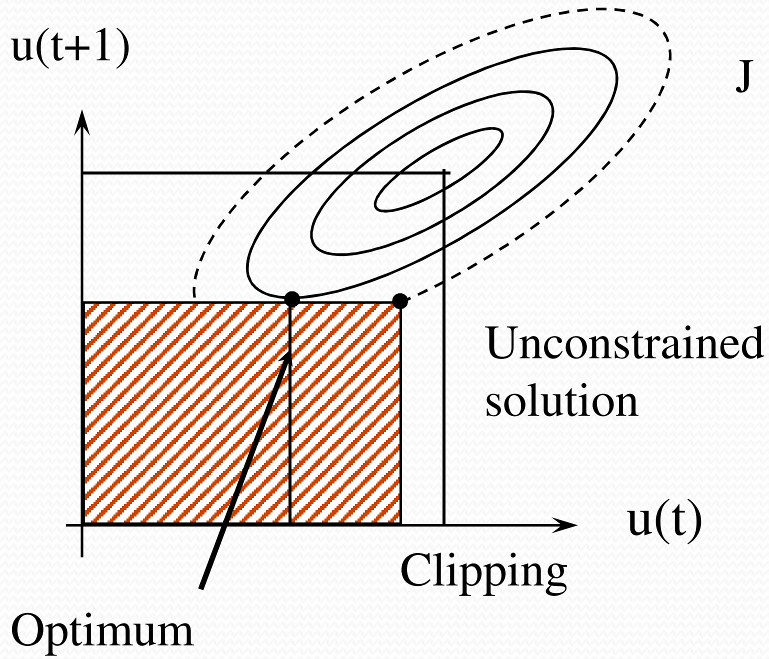
- Usually the optimum is close to the constraints. (Formula 1: speed, rpm)
- Objective: **work as close as possible** to the constraints but without **violation**
- If constraints are not considered: work far from the optimum (**safety**). Decrease in quality and yield (**benefits**)



MPC and constraints

MPC is the only methodology able to incorporate constraints in a systematic way in the **design phase** (there are other *ad-hoc* solutions like *override* control) **>>>>> great success in industry**

- Using a **dynamic model**, the controller can **anticipate** to the future evolution of the output
- Not considering constraints on **manipulated variables** (clipping) may result in higher values of the objective function.



Problem formulation

Now the objective is to **minimize a cost function subject to constraints.**

Constraints must be expressed as function of the **minimization (independent) variable u**

$$u_{min} < u(t) < u_{max}$$

$$U_{min} < U(t) < U_{max}$$

$$y_{min}(t) < y(t) < y_{max}(t)$$

for all *t* along the horizon

Since ***y = Gu + f***, the output constraints can be expressed as a function of ***u***

Computing the control action

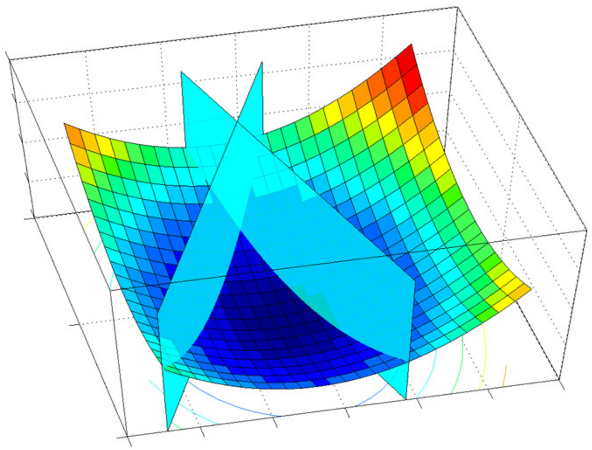
All constraints (**inequalities**) can be grouped and expressed as:

$$Ru \leq c$$

Now the problem to be solved is:

Min J

Subject to: $Ru \leq c$



Minimization of a **quadratic function** subject to **linear constraints**: **Quadratic Programming (QP).** Well-known optimization algorithm (iterative)

Control law computation

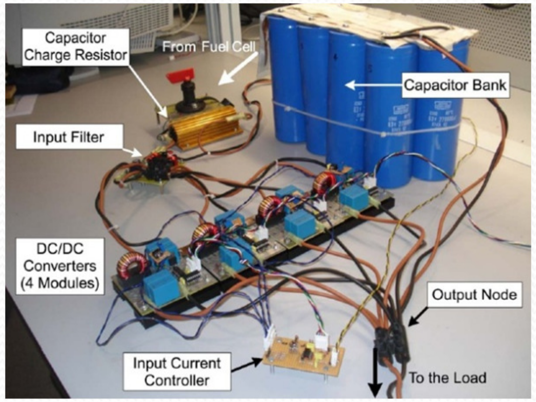
MODEL	COST FUNCTION	CONSTRAINTS	SOLUTION
Linear	Quadratic	None	Explicit
Linear	Quadratic	Linear	QP
Linear	Norm-1	Linear	LP

Outline

1. Introduction
2. Basic ideas of MPC
3. Some algorithms
4. MPC and Constraints
5. **MPC for “fast” systems**
6. Other formulations of MPC
7. Stability
8. Some applications
9. Conclusions and open issues

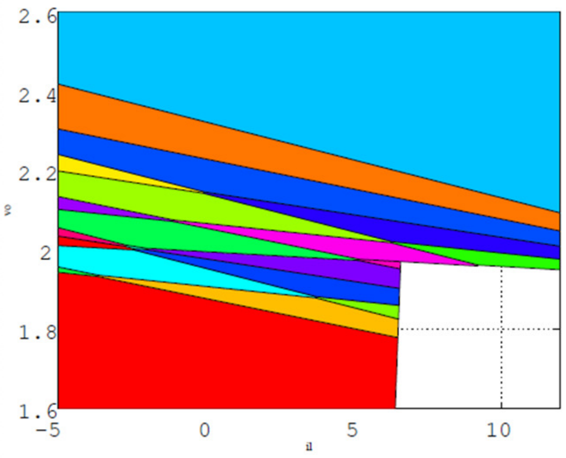
5. Fast Systems

- MPC originally developed for "slow" processes (chemical, thermal, etc.). Lot of time for on-line optimization
- Processes with small sampling times:
  - Power converters
  - Flight control
  - Automotive (ESP, ABS, power management)
  - Electromechanical systems
- Constrained MPC implies solving a QP (computational cost). Lack of time to solve a QP.



Explicit MPC

- *Use Multiparametric Programming.* Influence of parameter changes in a certain problem [Bemporad et al., 2002]
- The vector of changing parameters is ***x(t)***
- Idea:
  - Use the fact that the control law is PWA
  - Make a partition of the state space
  - Compute a controller for each region beforehand
  - On-line: apply the corresponding control law
- On-line optimization is avoided



A. Bemporad, M. Morari, V. Dua and E. Pistikopoulous. *The explicit linear quadratic regulator for constrained systems*, Automatica 38 (2002)

Explicit MPC

Explicit MPC solution:

- Off line computation + On line search
- The solution of the multi-parametric problem gives rise to a **PWA control law**

$$x = [y_k \ \cdots \ y_{k-4} \ \Delta u_{k-1} \ \cdots \ \Delta u_{k-3} \ \Delta v_{k-1} \ \cdots \ \Delta v_{k-4}]^T$$

$$\begin{array}{ll} min & \frac{1}{2} \cdot \mathbf{u}^T \cdot Q \cdot \mathbf{u} + \theta^T \cdot C^T \cdot \mathbf{u} \\ s.t. & R \cdot \mathbf{u} \leq b + S_\theta \cdot \theta \end{array}$$

$$\theta = \left[x \ w_{k+N_1} \ \cdots \ w_{k+N_2} \ u_{k-1}\right]^T$$

regions

$$u(t) = f(\theta(t))$$

$$f(\theta) = F^i \cdot \theta + g^i \quad if \quad H^i \cdot \theta \leq k^i, \quad i = 1, \ldots, N_{mpc}$$

Some drawbacks

- The number of regions increases rapidly with the horizon
- Efficient algorithms for region calculation and simplification must be used
- Online execution:
  - Search in the space of regions (branch and bound, etc.)
  - Memory for storage
- The state-space dimension can be really large for dead-time processes

Outline

1. Introduction
2. Basic ideas of MPC
3. Some algorithms
4. MPC and Constraints
5. MPC for "fast" systems
6. **Other formulations of MPC**
7. Stability
8. Some applications
9. Conclusions and open issues

6. Other formulations

- MPC for nonlinear processes
- Hybrid MPC
- Distributed MPC (just a few words)

6.1. MPC and nonlinear processes

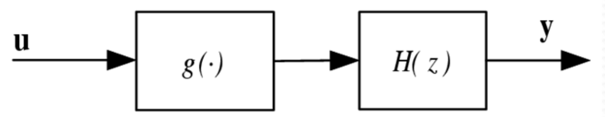
- Most processes are non-linear
- Linear approximations work for small perturbations around the operating point (well in most cases)
- There are processes with
  - **continuous transitions** (startups, shutdowns, etc.) and spend a great deal of time away from a steady-state operating region or
  - **never in steady-state** operation (i.e. batch processes, solar plants), where the whole operation is carried out in transient mode.
  - **severe nonlinearities** (even in the vicinity of steady states)

Linear vs Nonlinear

- Linear MPC is a mature discipline. The number of applications seems to duplicate every 4 years.
- Main issues concerning NMPC:
  - Modeling and identification
  - Optimization procedure
  - Analysis: stability, robustness, tuning.
- Some vendors have NMPC products: Adersa (PFC), Aspen Tech (Aspen Target), Continental Control (MVC), DOT Products (NOVA-NLC), Pavilon Tech. (Process Perfecter)

Nonlinear models

- First principles: equations obtained from the knowledge of the underlying process: Physics and Chemistry laws.
- Input/output. NARMAX
- $y(t)= \Phi(y(t-1), \ldots, y(t-n_y), u(t-1),\ldots,u(t-n_u),e(t),\ldots,e(t-n_{e+1}))$
  - Volterra, Hammerstein, Wiener (FIR, bilineal)
  - Neural Networks
  - Piece-wise affine (PWA)
  - Local model network



- State Space $x(t+1)=f(x(t),u(t))$, $y(t)=g(x(t))$
- Identification and state estimation are difficult (EKF, MVE)

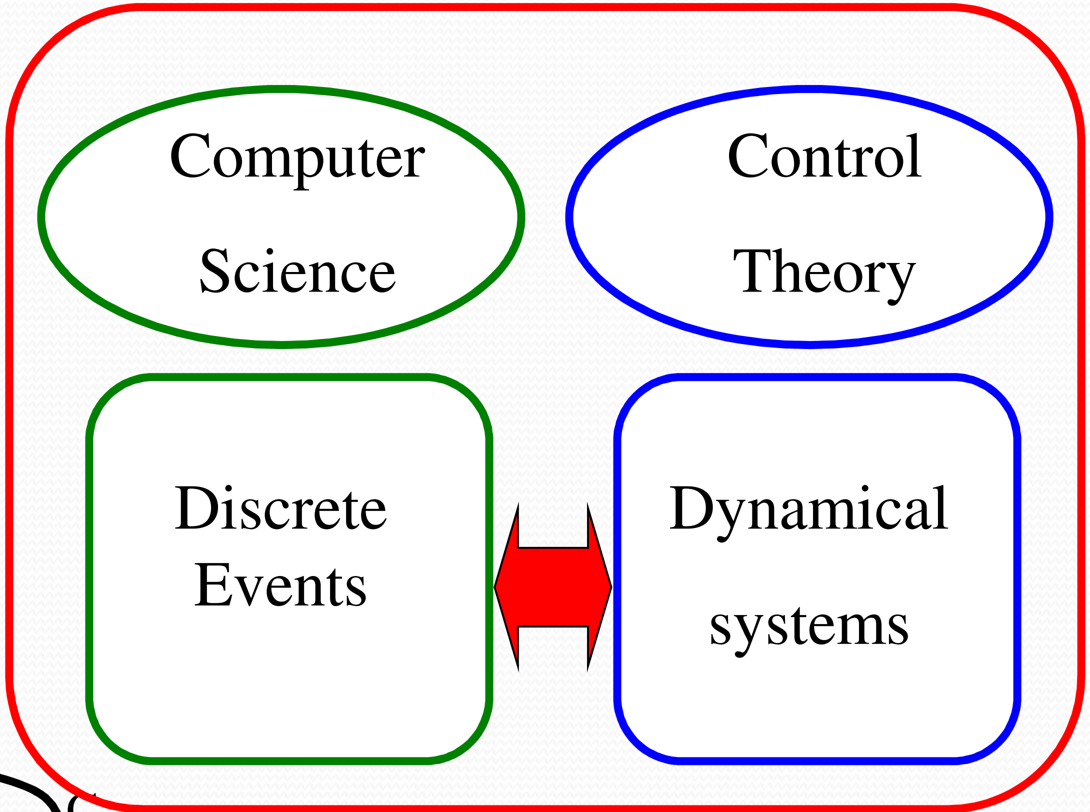
NMPC implementation

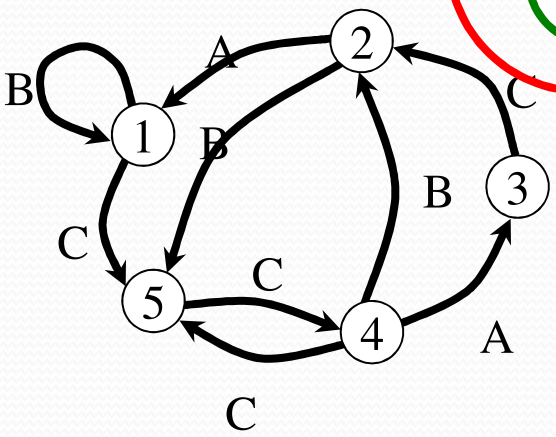
- Solving a **Nonlinear** (non QP), possibly **nonconvex**.
- **Computation** time increased (NLP)
- Real time and no convexity >>> **suboptimal solutions**
  - Sequential Quadratic Programming (SQP)
  - Simultaneous approach (Findeisen and Allgower'02)
  - Using a sequential approach with successive linearization around the previous trajectory.
  - PWA: Mixed Integer Programming Problem
  - Use the model structure: Volterra & Hammerstein models
  - Neural Networks (possibly combined with state-space)
  - Using short horizons
  - Etc.

6.2 MPC for Hybrid Systems

X={1,2,3,4,5}

U={A,B,C}





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

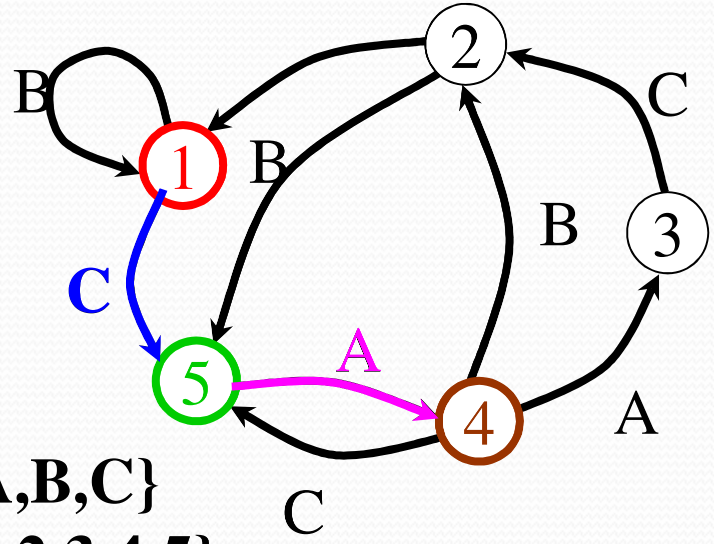
$u \in \mathbf{R}^m$

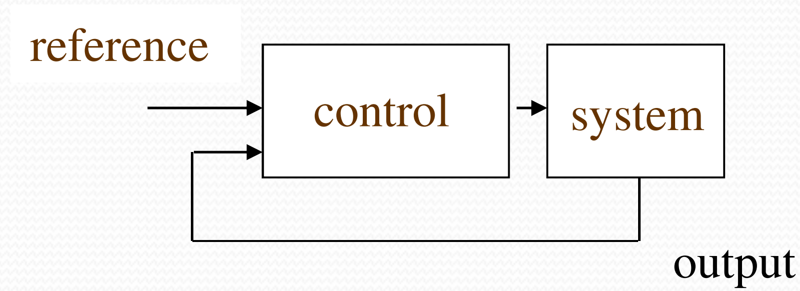
$y \in \mathbf{R}^p$

$$\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k))$$

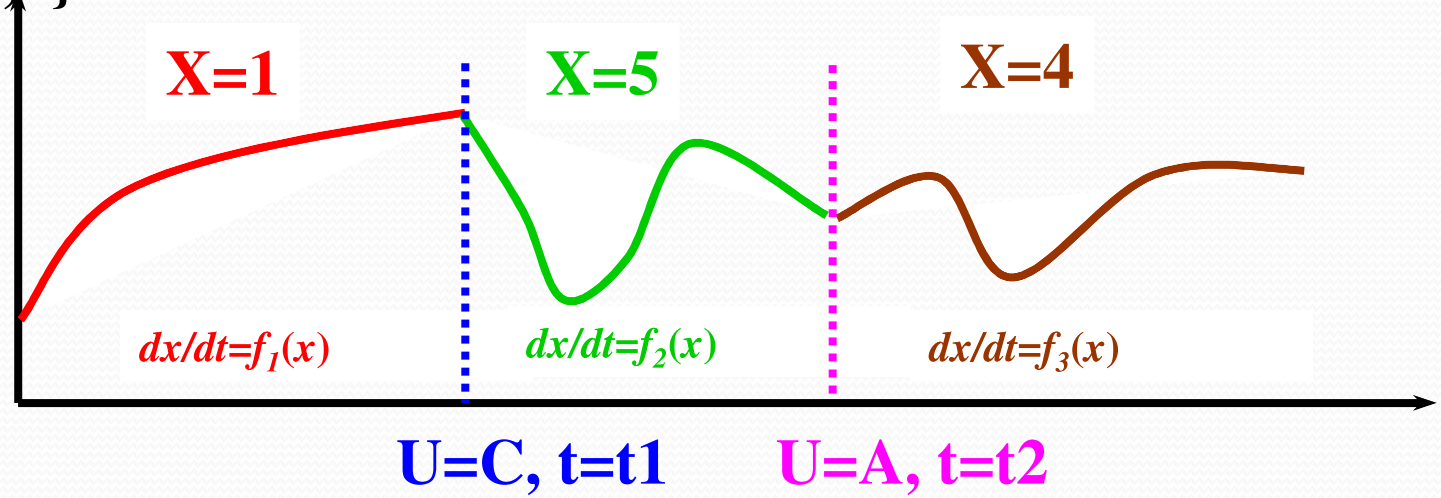
$$\mathbf{y}(k) = g(\mathbf{x}(k), \mathbf{u}(k))$$

Discrete Events & Dynamical Systems





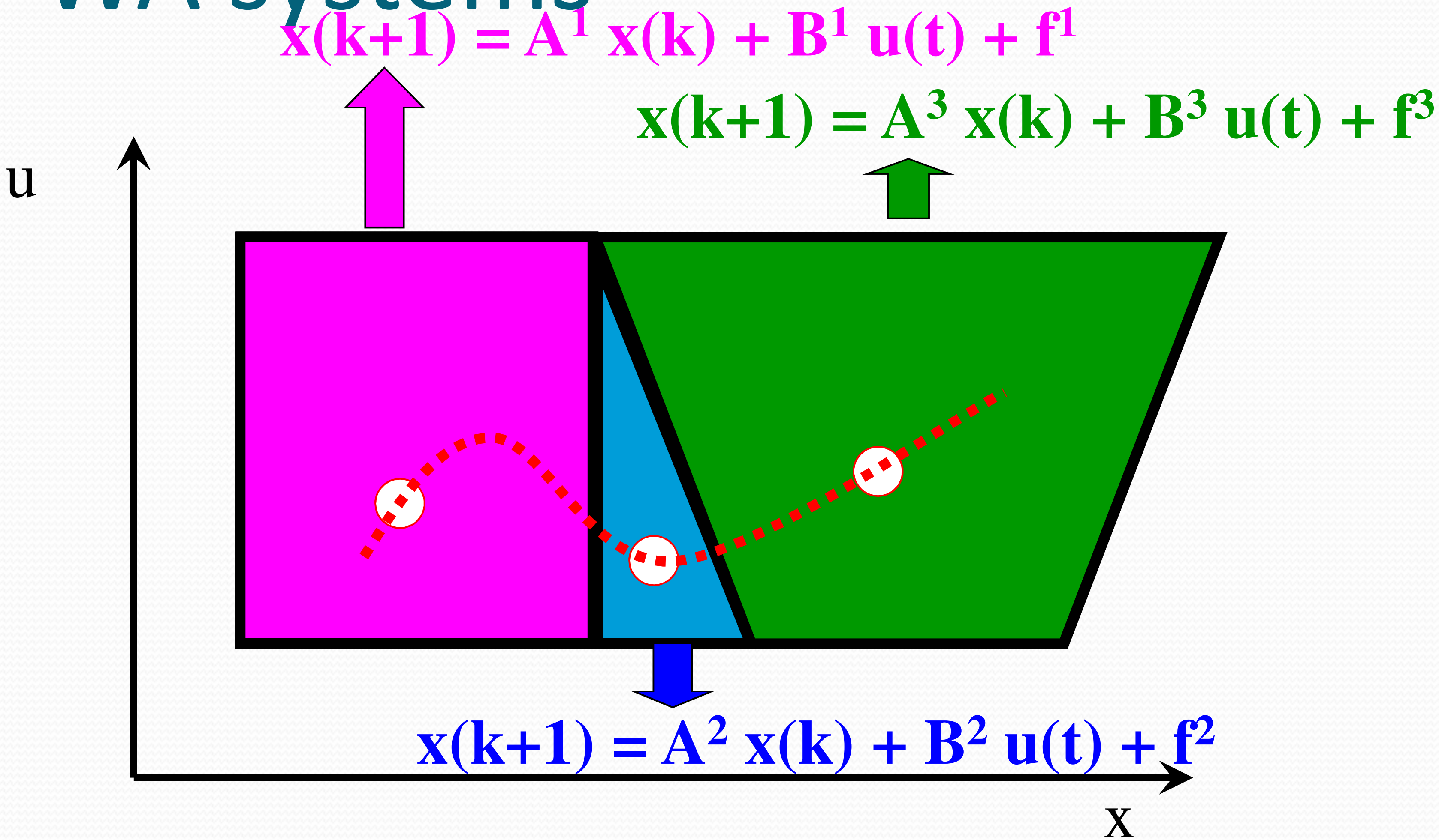
U={A,B,C}
X={1,2,3,4,5}



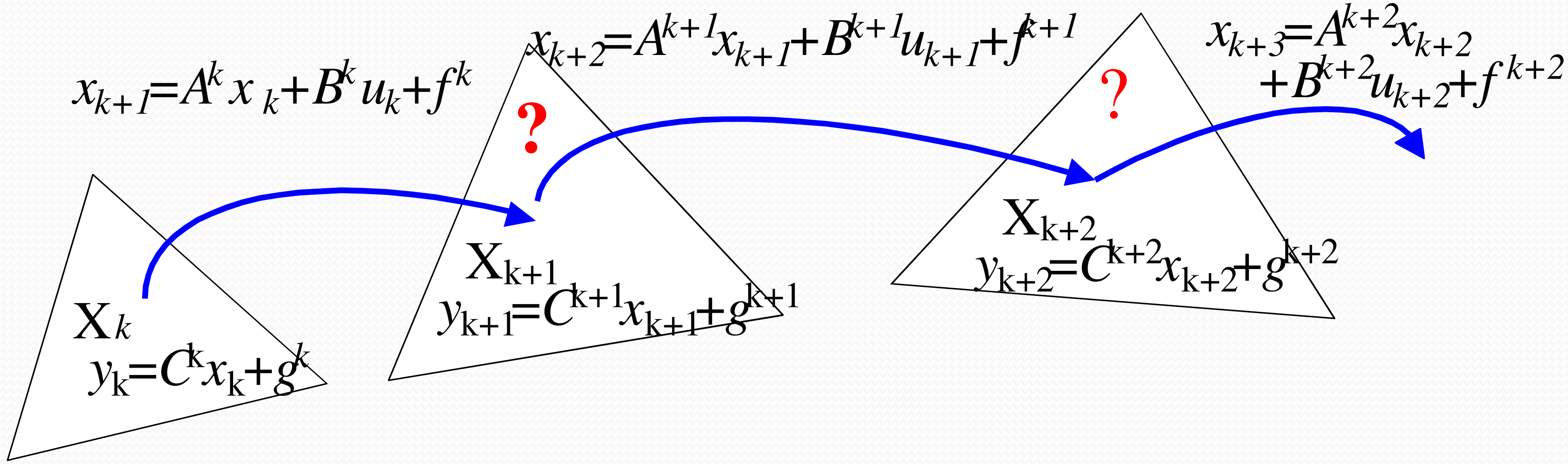
Different Hybrid Models

1 - Piecewise Affine (PWA) Systems
2 - Mixed Logical Dynamical (MLD) Systems
3 - Linear Complementary (LC) Systems
4 - Extended Linear Complementary (ELC) Systems
5 - Max-Min-Plus-Scaling (MMPS) Systems

PWA systems



Mixed Integer-Real Optimization (MILP, MIQP, MINLP)



The resulting optimization problem

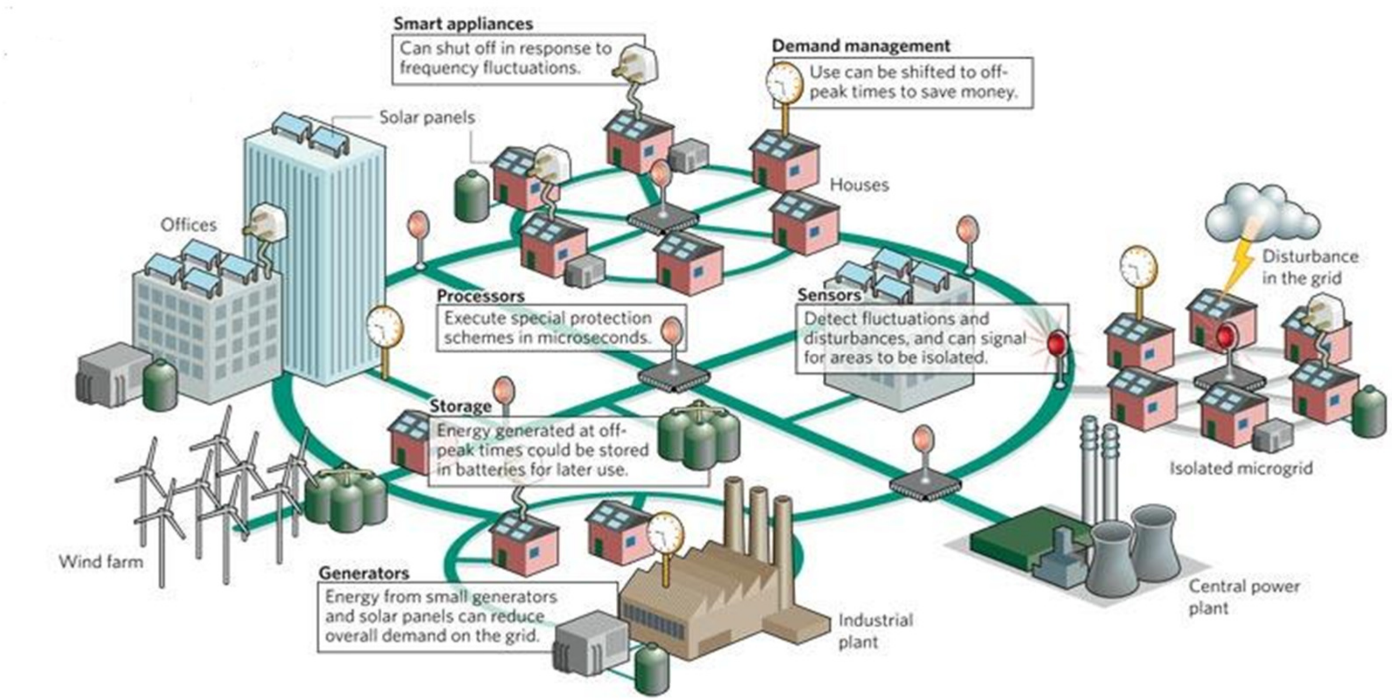
$$U = \arg(\min_{U,I} J)$$

U = {u(k), u(k+1), u(k+2), …,u (k+N-1)} real

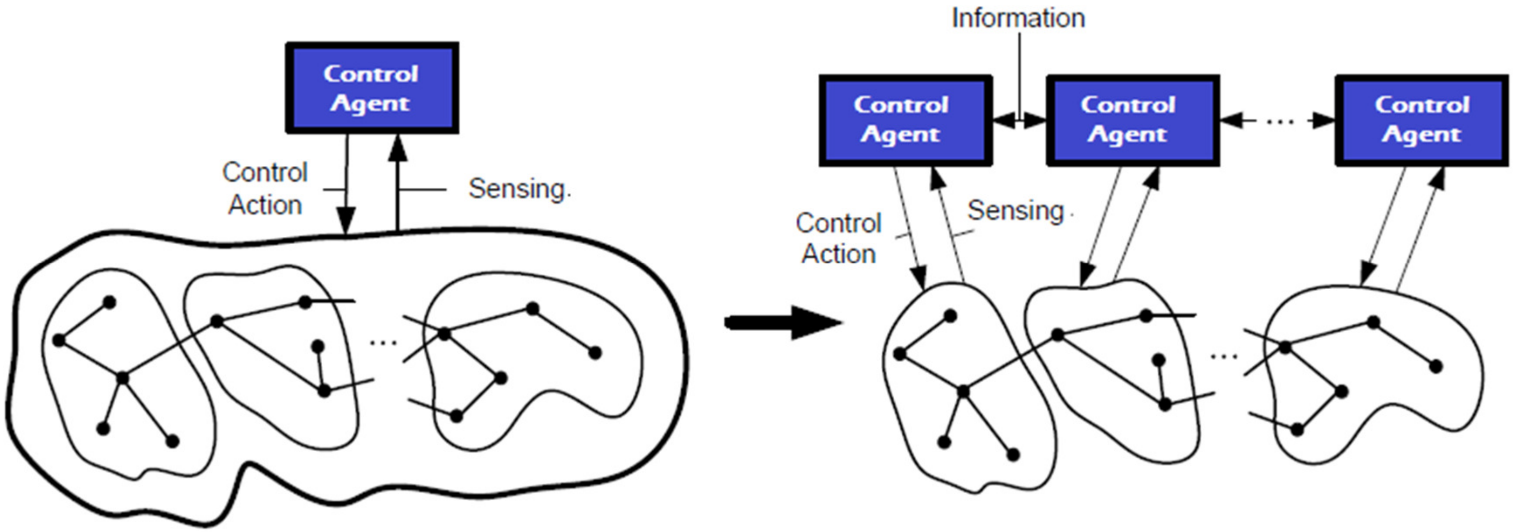
I = {I(k), I(k+1), I(k+2),…, I(k+N-1)} Integer

6.3. Distributed MPC

- Large-scale systems: distribution networks, smart grids, etc.
- Overcome **computational** and **communication** limitation of centralized approaches



Distributed control techniques



- Decentralized
- Cooperation-based
- Communication-based
- Sensitivity-driven
- Feasible cooperation
- Lagrange multipliers

"A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark." I. Alvarado, D. Limon, D. Muñoz de la Peña, J.M. Maestre, M.A. Ridao, H. Scheu, W. Marquardt, R.R. Negenborn, B. De Schutter, F. Valencia and J. Espinosa. Journal of Process Control 21 (2011) 800–815.

Outline

1. Introduction
2. Basic ideas of MPC
3. Some algorithms
4. MPC and Constraints
5. MPC for “fast” systems
6. Other formulations of MPC
7. **Stability**
8. Some applications
9. Conclusions and open issues

Stability

- Optimal controllers with infinite horizon guarantee stability.
  - the objective function can be considered a Lyapunov function, providing nominal stability. **Cannot be implemented:** an infinite set of decision variables.
  - **Practice: use long horizon**
- Optimal **finite** horizon and the presence of **constraints** make it very difficult to prove stability

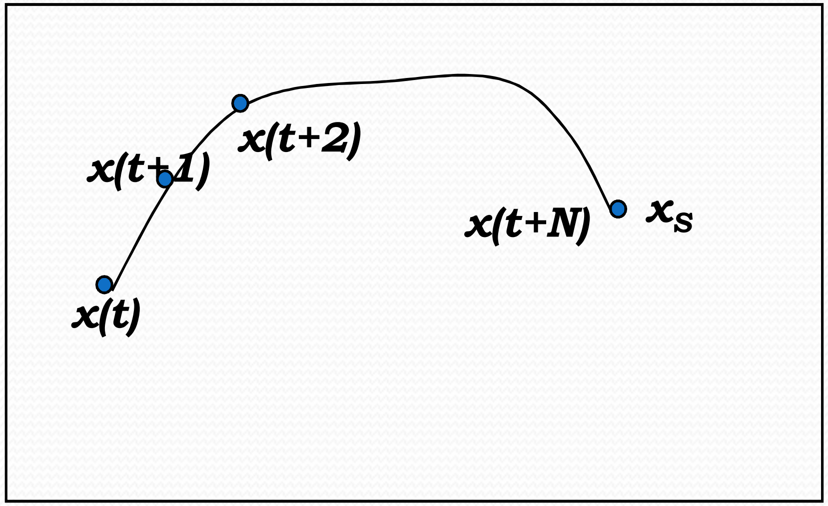
MPC stability

- **Terminal cost**. Bitmead et al'90 (linear unconstrained), Rawlings & Muske'93 (linear constrained).

$$J(\bar{x}, u_{t,t+N-1}, N) = \sum_{k=t}^{t+N-1} l(x(k), u(k)) + V_f(x(t+N))$$

- The terminal cost is an associated Control Lyapunov function

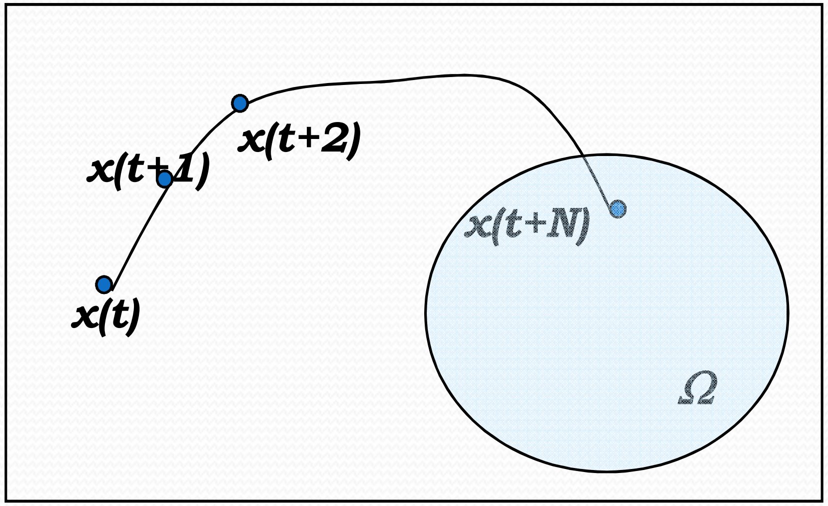
- **Terminal state equality constraint**. Kwon & Pearson'77 (LQR constraints), Keerthi and Gilbert'88, $x(k+N) = x_S$. Too strict: **Feasibility** problems



MPC stability: all ingredients

- **Dual control (terminal set)**. Michalska and Mayne (1993) $\boldsymbol{x(N)} \in \Omega$

  Once the state enters Ω the controller switches to a previously computed stable linear strategy.



Asymptotic stability theorem (Mayne 2001):

- The terminal set Ω is a control invariant set. (It ensures **feasibility**)
- The terminal cost $\boldsymbol{F(x)}$ is an associated Control Lyapunov function such that

  $$\min_{\{\mathbf{u} \in \mathbf{U}\}} \{F(f(x,u))-F(x) + l(x,u) \mid f(x,u)\in\Omega\} \leq 0 \ \forall x\in\Omega$$

- Then **the closed loop system is asymptotically stable** in $X_N(\Omega)$

Formulation that guarantees stability

$$J(\bar{x}, u_{t,t+N-1}, N) = \sum_{k=t}^{t+N-1} l(x(k), u(k)) + V_f(x(t+N))$$

subject to

(i) $x(k+1) = f(x(k), u(k)), \quad x(t) = \bar{x}, \quad k \geq t$

(ii) the constraints $x \in X, u \in U, k \in [t, t+N-1]$

(iii) the terminal state constraint $x(t+N) \in X_f$

- The invariant condition on terminal set W ensures **feasibility** while the condition on the terminal function V_f guarantees **convergence**.
- Important issue: how to choose the terminal cost and region?

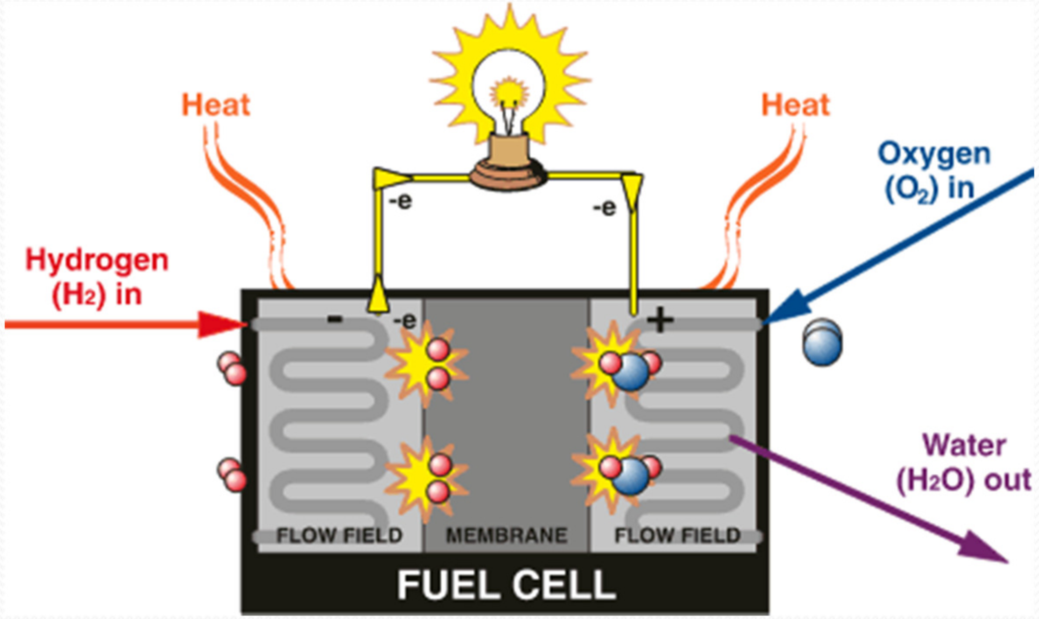
Outline

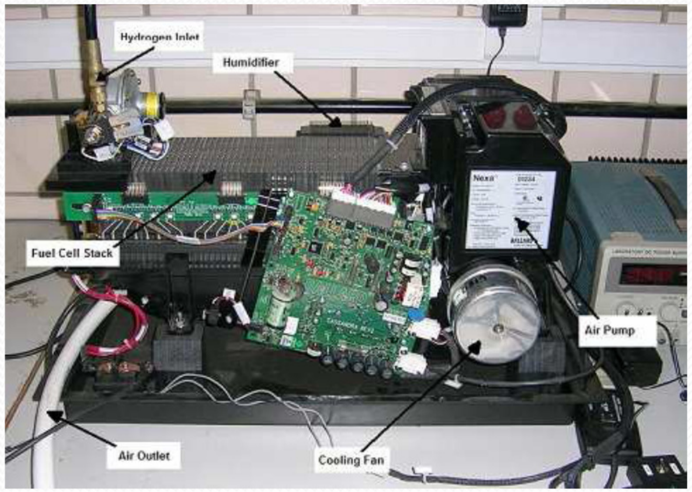
1. Introduction
2. Basic ideas of MPC
3. Some algorithms
4. MPC and Constraints
5. MPC for “fast” systems
6. Other formulations of MPC
7. Stability
8. **Some applications**
9. Conclusions and open issues

Some applications not considered as “process control”

- Fuel Cells (Explicit)
- Hybrid vehicle (Hybrid)
- Power network (Distributed)

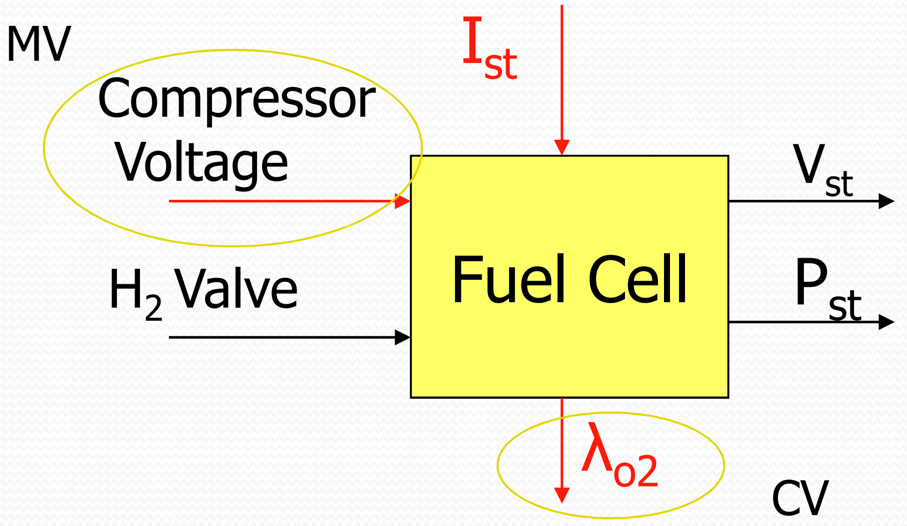
8.1. “Fast system”: Fuel Cell





Airflow control

- **Objective**: supply in an effective way the necessary flow of reactans, providing a good transient response and minimizing auxiliary consumption
- Important: keep the oxygen excess ratio. *Starvation* danger
- Control the flow of oxygen



$$\lambda_{O2} = \frac{W_{O2,in}}{W_{O2,react}}$$

Constrained Predictive Controller

$$J = \sum_{j=N_1}^{j=N_2} [y(t+d|t) - w(t+d|t)]^2 + \sum_{j=1}^{j=N_u} \lambda(j) \cdot [\Delta u(t+j-1)]^2$$

Subject to:

$$V_{min} \leq V_{cp} \leq V_{max}$$

Input constraint (physical limits)

$$\lambda_{O_2,min} \leq \lambda_{O_2} \leq \lambda_{O_2,max}$$

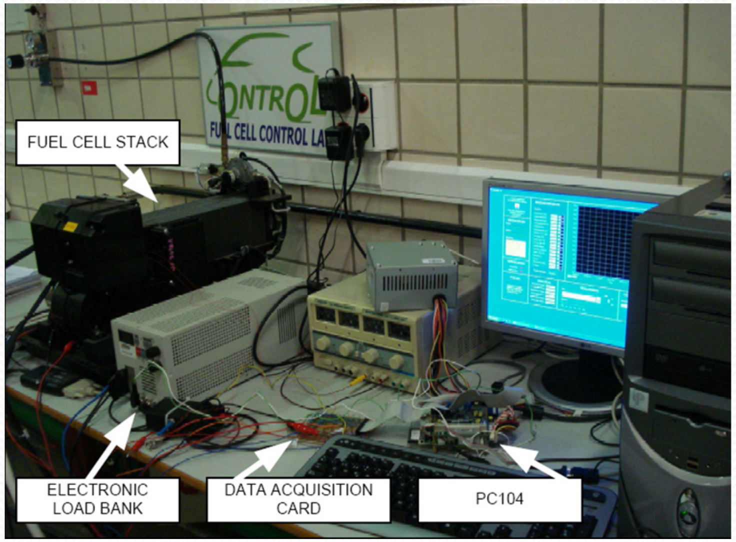
Desired output constraint

- Implicit feed-forward effect:

$$J = (G \cdot u + H \cdot v + f - w)^T \cdot (G \cdot u + H \cdot v + f - w) + \lambda \cdot u^T \cdot u$$

- Sampling time according to system dynamics: **10 milliseconds**
- **Explicit solution**

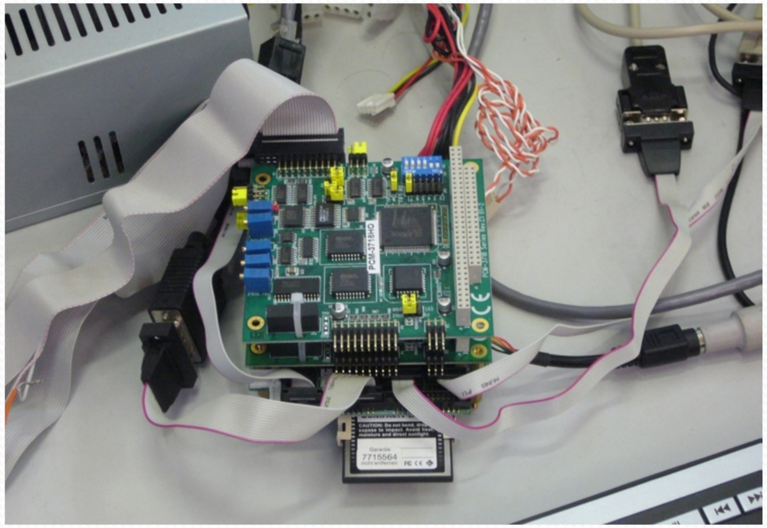
Experimental setup

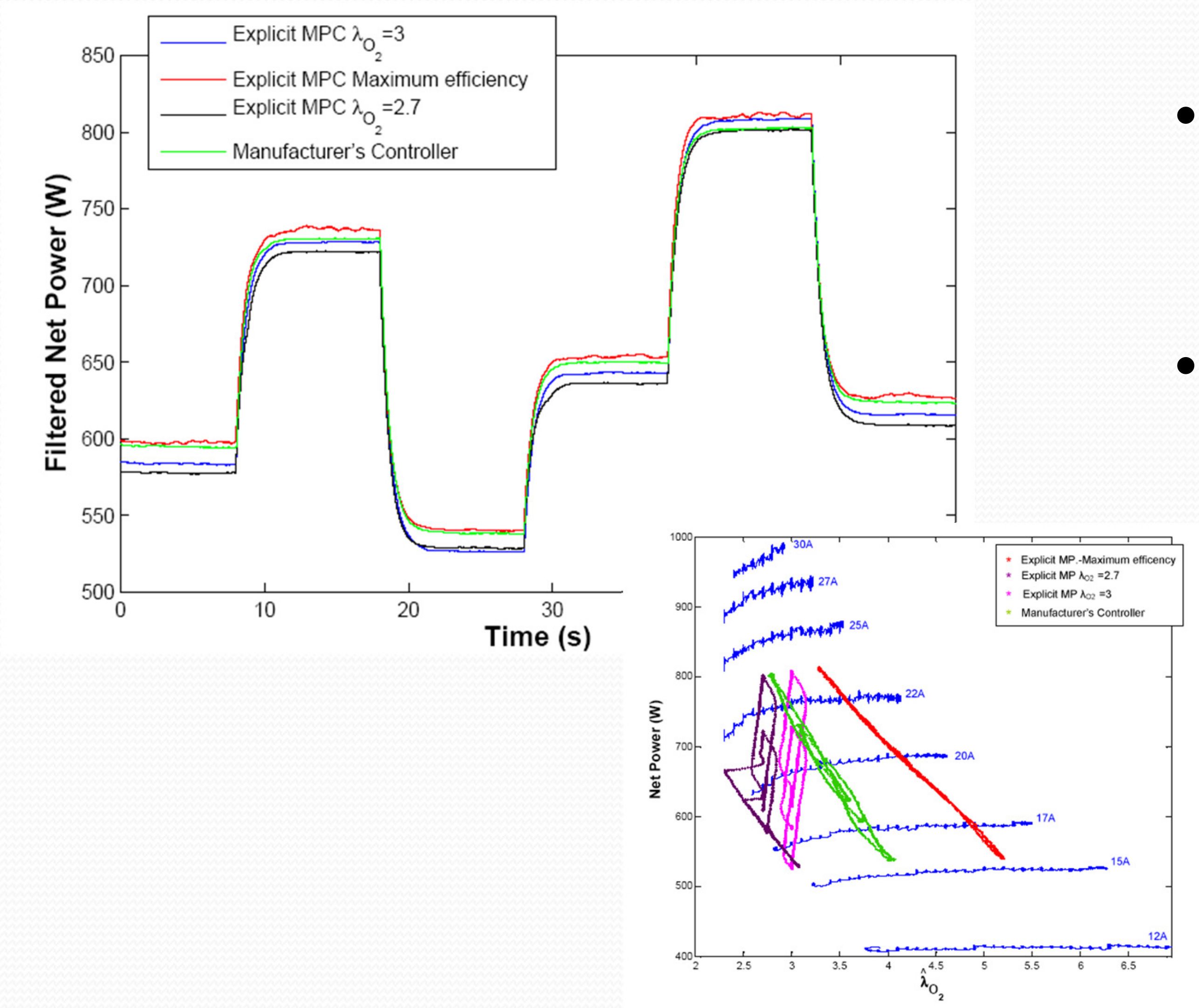


Horizons: 4 (small)
221 regions
Sampling time: 10 ms
Average exec time: 0.245 ms

Advantech PCM-3370 CPU: 650MHz
Pentium III with 256MB
2 Advantech PCM-3718HO
multifunction cards (16 AI and 1 AO)

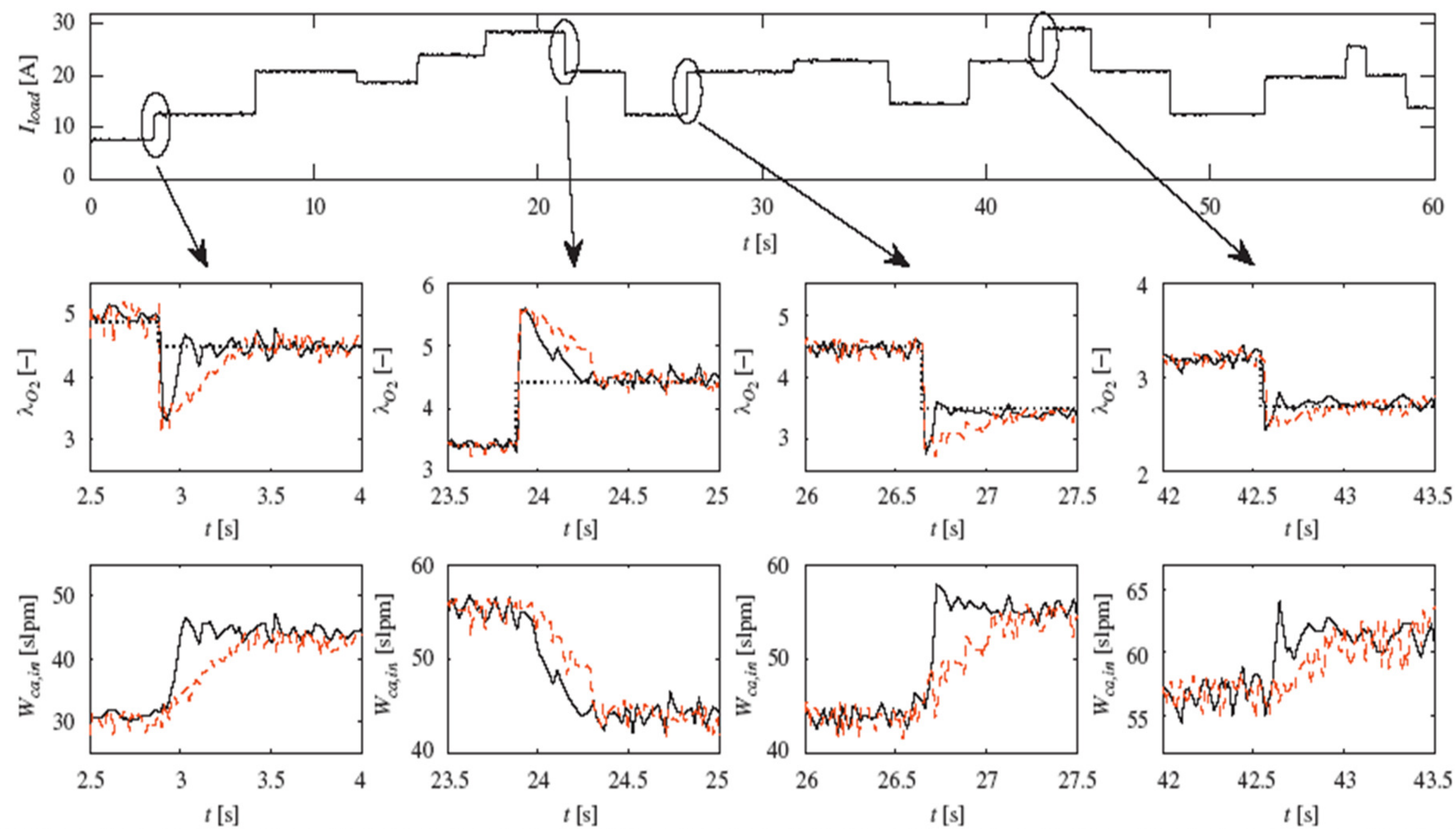
Simulink Real Time Workshop





- Allows performance improvements of up to 3.46%.
- Improved transient responses compared to those of the manufacturer's control law.

Experimental results



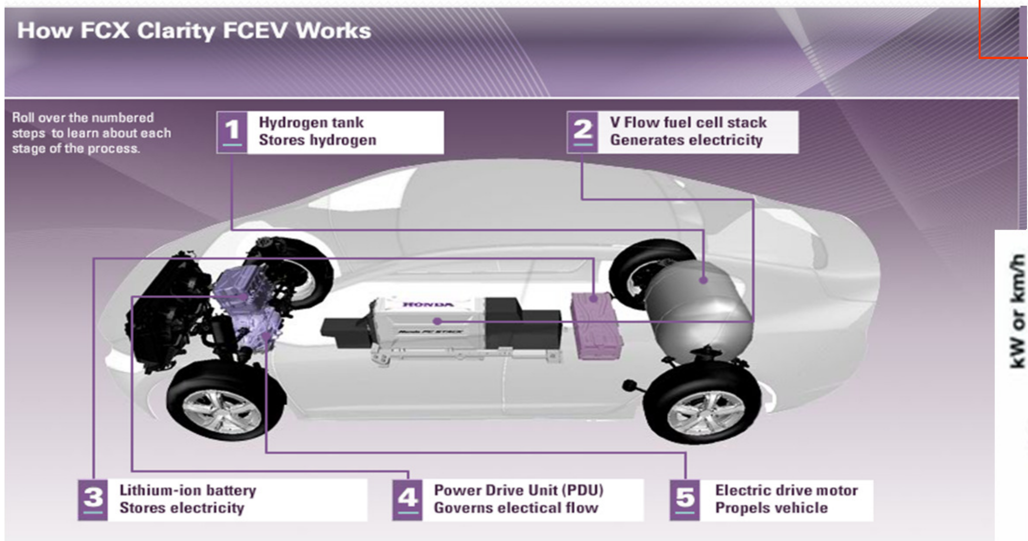
Real-Time Implementation of a Constrained MPC for Efficient Airflow Control in a PEM Fuel Cell. Alicia Arce, Alejandro J. del Real, Carlos Bordons and Daniel R. Ramírez, *IEEE Transactions on Industrial Electronics* (2009)

8.2. Hybrid system: FC hybrid vehicle

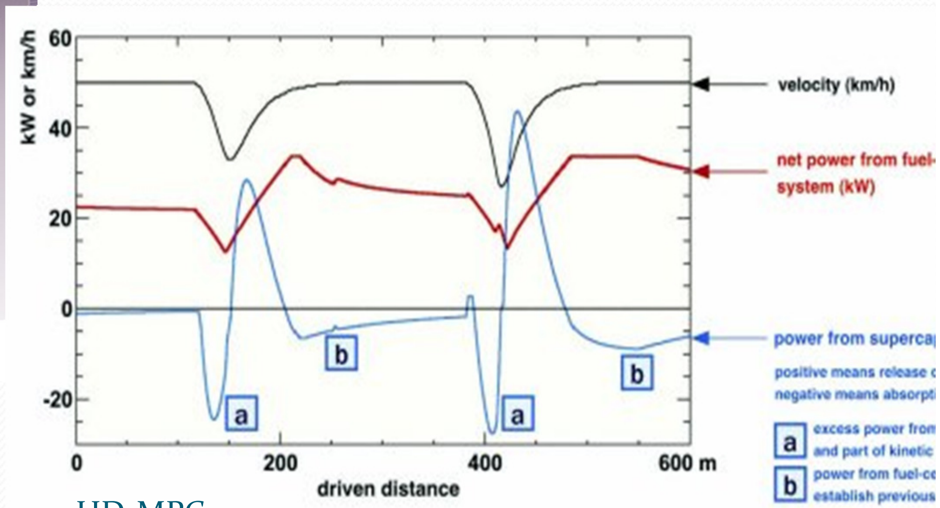
System that is **hybrid by nature**: different power sources-different dynamics

Two energy sources

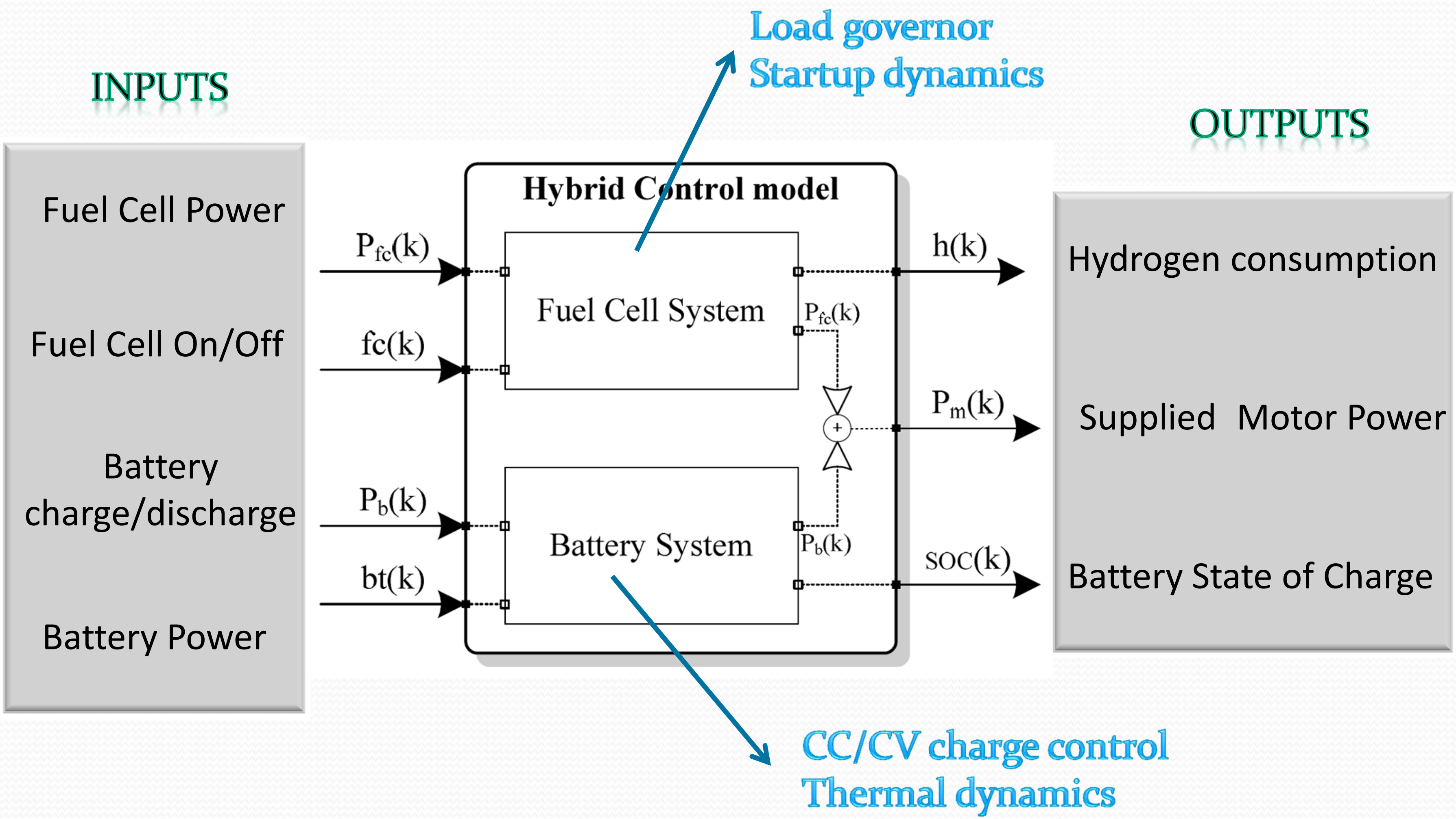
On-board power control for a fuel cell/battery-powered vehicle propulsion system



The most distinctive feature of the FCX Clarity Fuel Cell Electric Vehicle (FCEV)—other than the fuel cell itself—is the streamlined layout made



Hybrid Model



Hybrid Controller

MIXED-LOGICAL DYNAMICAL FORMULATION (MLD)

FUEL CELL EVENTS

$\delta_e^1(k) = 1 \quad if \quad P_{fc,d}(k) \leq 4000 \quad \text{or} \quad 0 \quad \text{otherwise}$

$\delta_e^2(k) = 1 \quad if \quad P_{fc,d}(k) \leq 0 \quad \text{or} \quad 0 \quad \text{otherwise}$

$\delta_e^3(k) = 1 \quad if \quad t_1(k) \geq 5 \quad \text{or} \quad 0 \quad \text{otherwise}$

$\delta_e^4(k) = 1 \quad if \quad t_2(k) \geq 25 \quad \text{or} \quad 0 \quad \text{otherwise}$

BATTERY EVENTS

$\delta_e^5(k) = 1 \ if \ P_b(k) \geq 0 \ \text{or } 0 \text{ otherwise}$

$\delta_e^6(k) = 1 \ if \ i_b(k) = 125 \ \text{or } 0 \text{ otherwise}$

$\delta_e^7(k) = 1 \ if \ i_b(k) \geq 125 \ \text{or } 0 \text{ otherwise}$

$\delta_e^8(k) = 1 \ if \ t_3(k) \leq 60 \ \text{or } 0 \text{ otherwise}$

$\delta_e^9(k) = 1 \ if \ t_3(k) \geq 30 \ \text{or } 0 \text{ otherwise}$

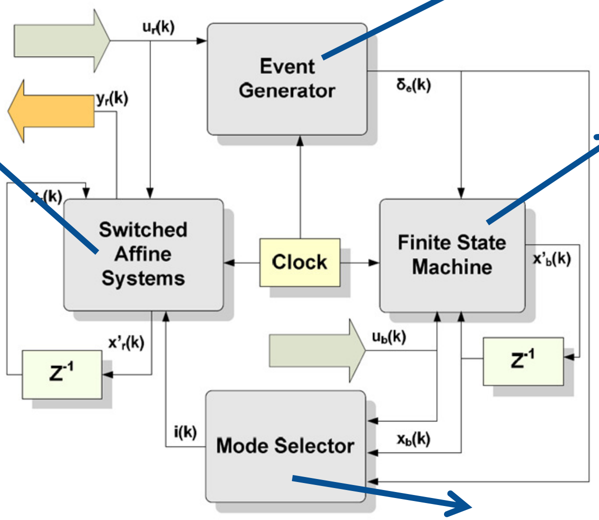
$\delta_e^{10}(k) = 1 \quad if \quad P_b(k) = ccontrol \quad \text{or} \quad 0 \quad \text{otherwise}$

$$x_r(k+1) = A_{i(k)}x_r(k) + B_{i(k)}u_r(k) + f_{i(k)}$$
$$y_r(k) = C_{i(k)}x_r(k)$$

$x_r(k) = [P_m(k) \quad q_{H_2}(k) \quad SOC(k) \quad i_b(k) \quad t_1(k) \quad t_2(k) \quad t_3(k)]^T$

$y_r(k) = [P_m(k)]^T$

$u_r(k) = [P_b(k) \quad P_{fc,d}(k)]^T$



$x_{b1}(k) = \neg fc(k) \wedge \neg bt(k)$

$x_{b2}(k) = \neg fc(k) \wedge bt(k)$

$x_{b3}(k) = fc(k) \wedge bt(k)$

$x_{b4}(k) = fc(k) \wedge \neg bt(k)$

CONSTRAINTS

$0.2 \leq SOC(k) \leq 0.8$

$0 \leq P_{fc,d}(k) \leq 56kW$

$-4.7kW \leq P_b(k)$

$I_b(k) \leq 220A$

$\delta_e^1(k) \to \neg fc(k) \quad , \quad \neg\delta_e^1(k) \to fc(k)$

$\delta_e^5(k) \to \neg bt(k) \quad , \quad \neg\delta_e^5(k) \to bt(k)$

$\neg fc(k) \to \delta_e^2(k)$

$\neg fc(k) \leftarrow (x_{b1}(k) \vee x_{b2}(k)) \wedge \neg\delta_e^3(k)$

$fc(k) \leftarrow (x_{b3}(k) \wedge x_{b4}(k)) \vee \neg\delta_e^4(k)$

$\delta_e^6(k) \leftarrow \delta_e^8(k) \wedge \delta_e^9(k)$

$x_{b2}(k) \vee x_{b3}(k) \to \delta_e^{10}(k)$

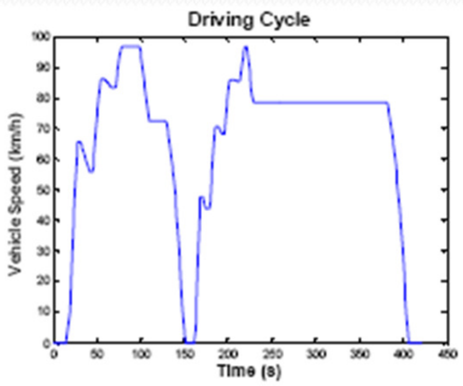
$i(k) = j \quad if \quad f_{Mj}(x_{b1}(k), \ldots, x_{b4}(k), \delta_e^5(k), \ldots, \delta_e^9(k)) = 1$
$\forall j = 1, \ldots, 34$

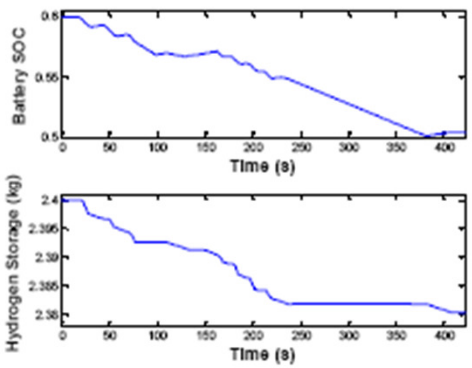
Simulation results

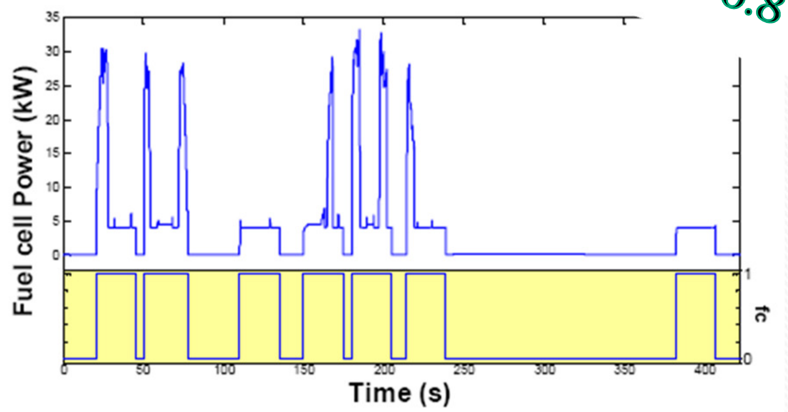
City driving cycle

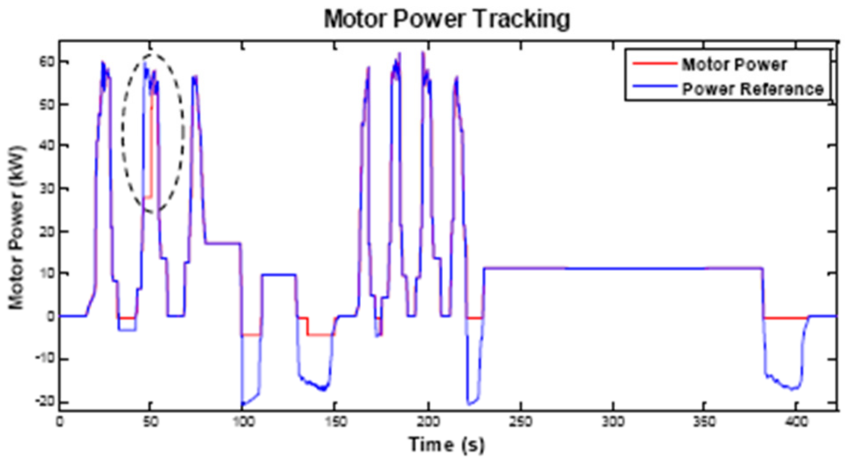
Execution time: less than 10 ms

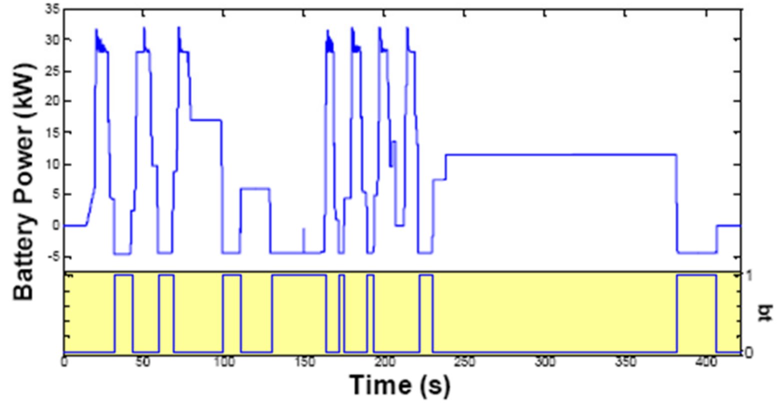
SOC=0.8



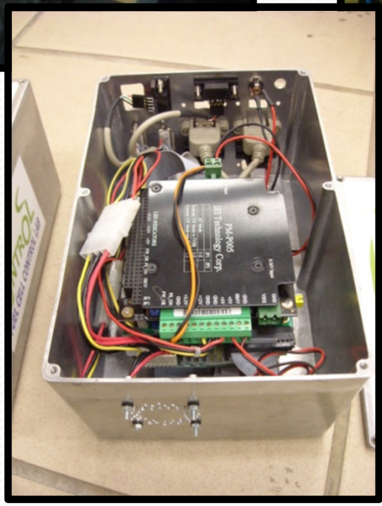


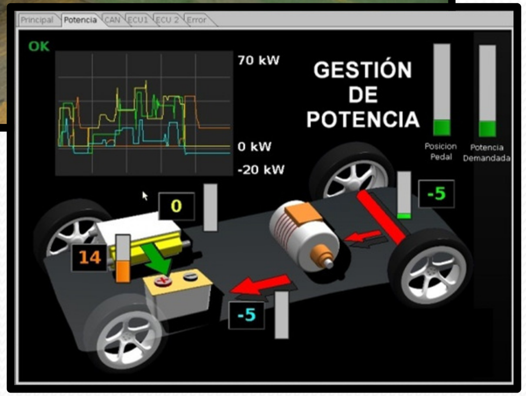




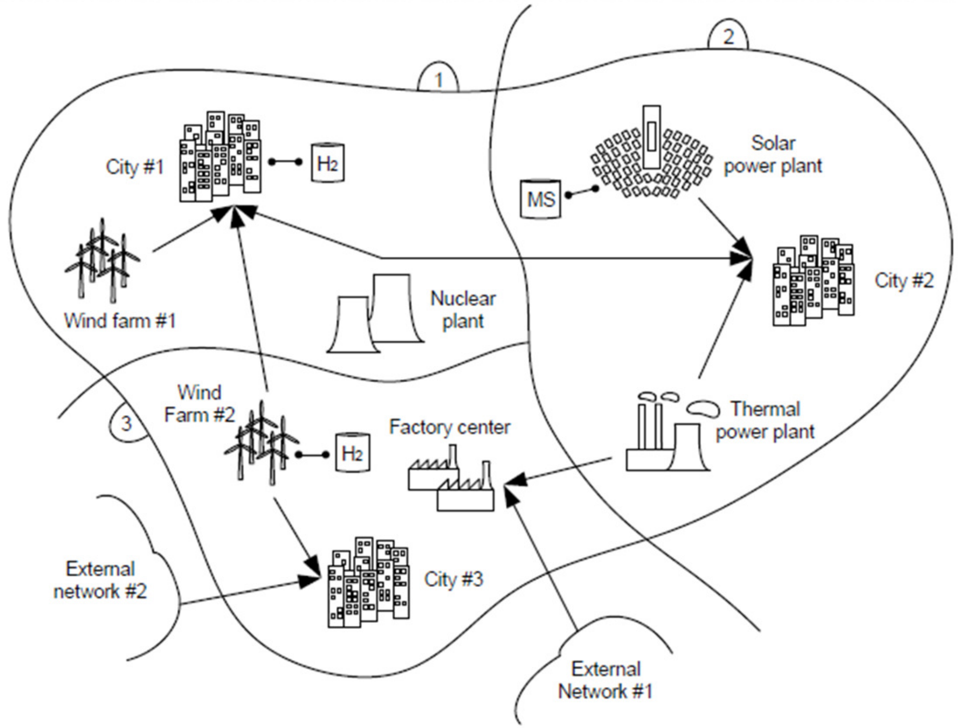


Prototype. Ready for tests.





Application of Distributed MPC. Power network

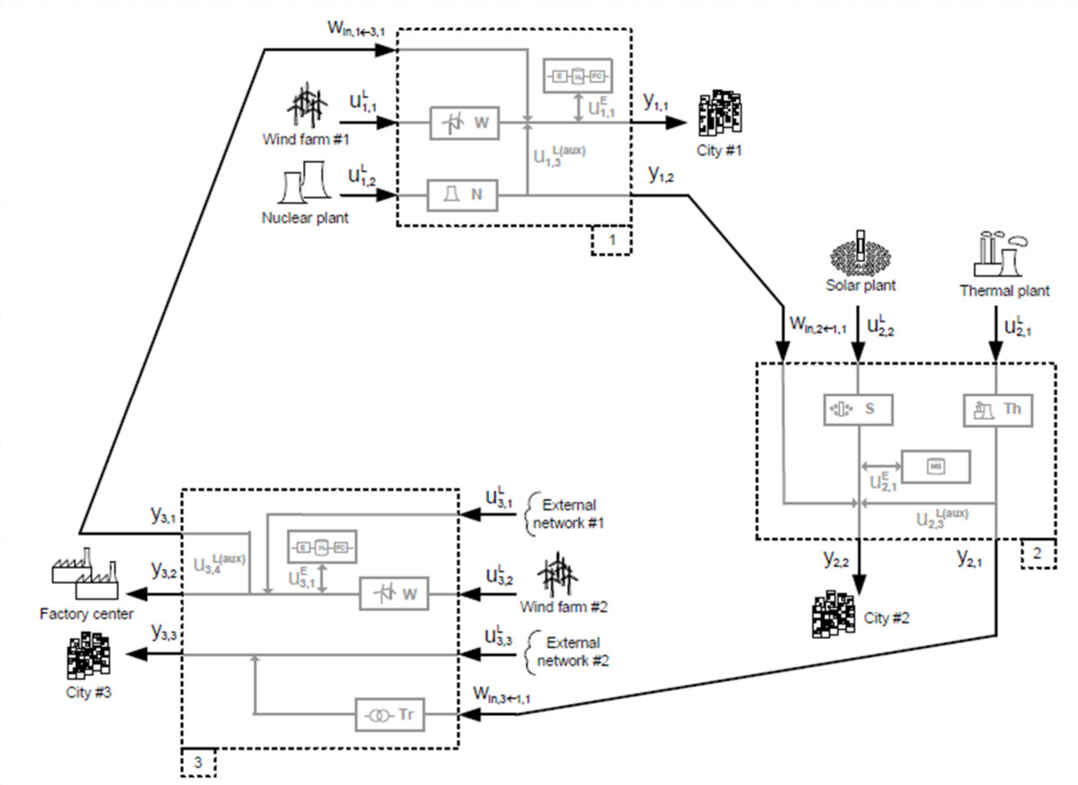


Integrating RES and storage

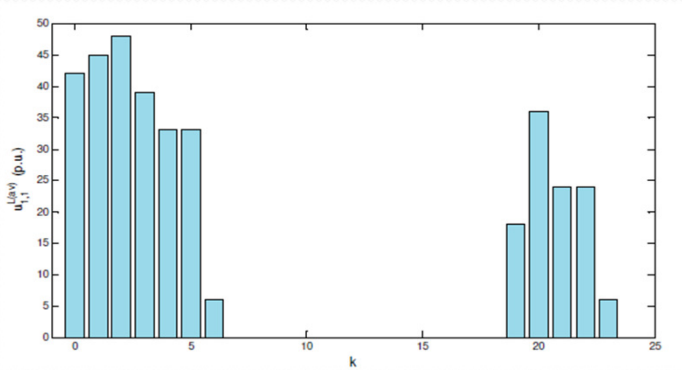
Solved using **Energy hubs** as modelling framework and **lagrangian distributed** MPC as control approach

PhD Thesis: *An Integrated Framework for Distributed Model Predictive Control of Large Scale Networks. Applications to Power Networks.* Alejandro J. del Real, June 2011.

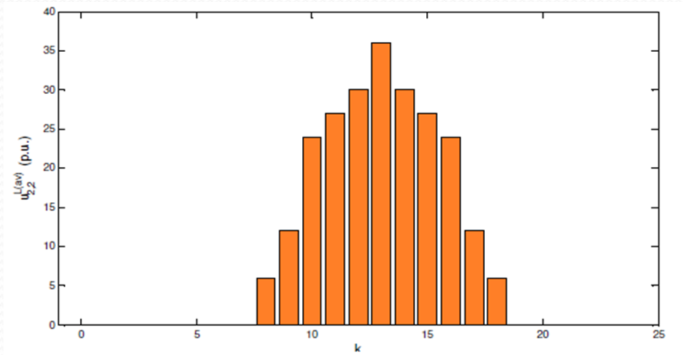
Modelling the network as energy hubs (integrating electrical and thermal energy)



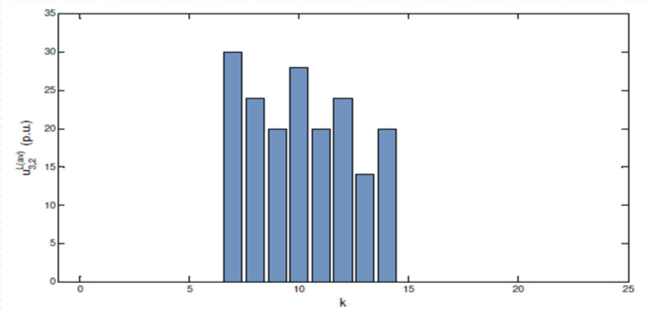
Availability of power sources



wind farm#1



Solar thermal



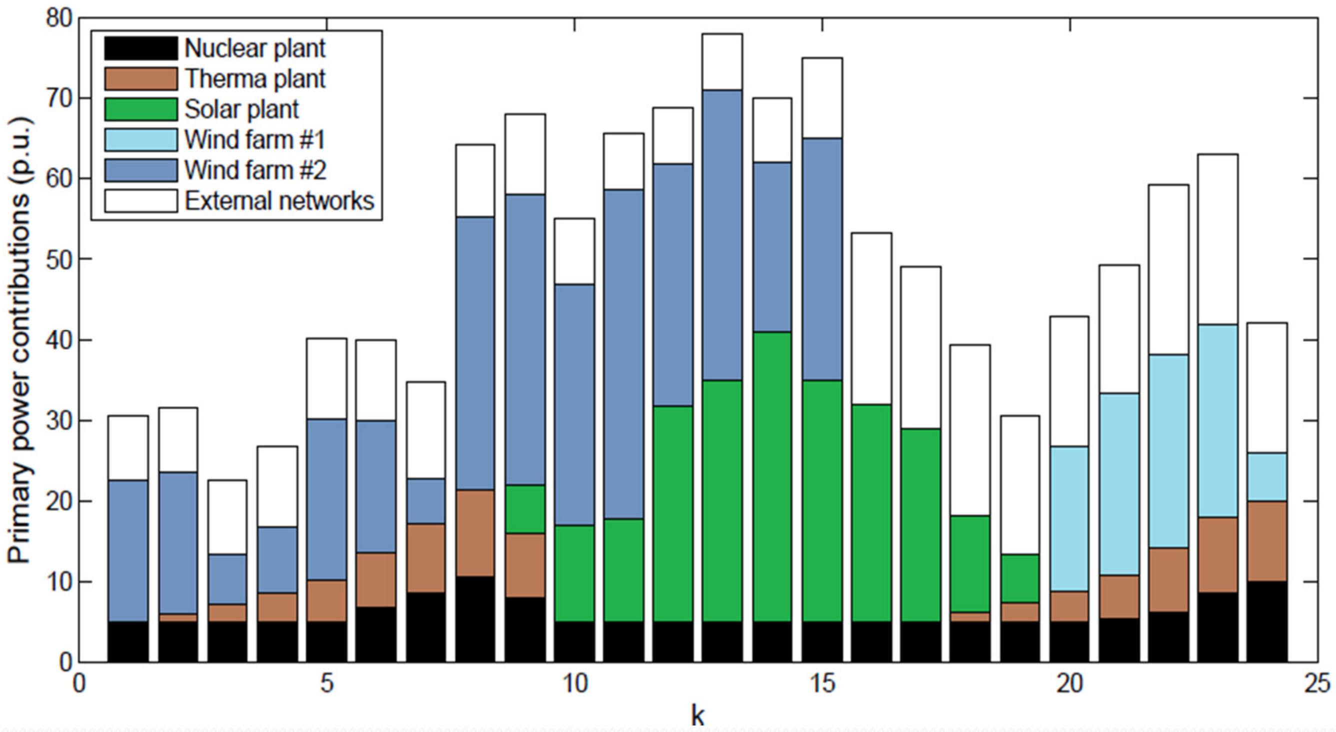
wind farm#2

74

Resulting power mix

Distributed computation of the solution. Three agents, including constraints, prices and storage limits

Allows sensitivity analysis: energy prices, storage capabilities, generation limits, etc.



Outline

1. Introduction
2. Basic ideas of MPC
3. Some algorithms
4. MPC and Constraints
5. MPC for "fast" systems
6. Other formulations of MPC
7. Stability
8. Some applications
9. **Conclusions and open issues**

Conclusions

- General overview. Different “flavors” of MPC
- Nowadays, MPC is a specific thought in controller design, from which many kinds of computer control algorithms can be derived for different systems.
- Well established in industry and academia.
- Great expectations for MPC
- Not only for “slow” processes
- Many open issues

Open issues. Is better MPC possible/needed?

- Efforts to develop MPC for more difficult situations:
  - Multiple and logical objectives (Morari, Floudas)
  - Hybrid processes (Morari, Bemporad, Borrelli, De Schutter, van den Boom …)
  - Nonlinear (Alamir, Alamo, Allgower, Biegler, Bock, Bravo, Chen, De Nicolao, Findeisen, Jadbadbadie, Limon, Magni, …)
  - Large-scale systems: hierarchical and distributed (Scattolini, Rawlings, De Schutter, Negenborn, …)
- MPC robustness, adaptation, nonlinearity handling, performance monitoring, model building, computation, and implementation
- **Practitioners** wish: increase the time that MPC is not in the manual mode!

Introduction to Model Predictive Control

Amazing tour around MPC (in just 45 minutes!)

Carlos Bordons
University of Seville, Spain

Industry Workshop on HD-MPC. 24th June 2011. Leuven

HD–MPC