Automatic Control 2

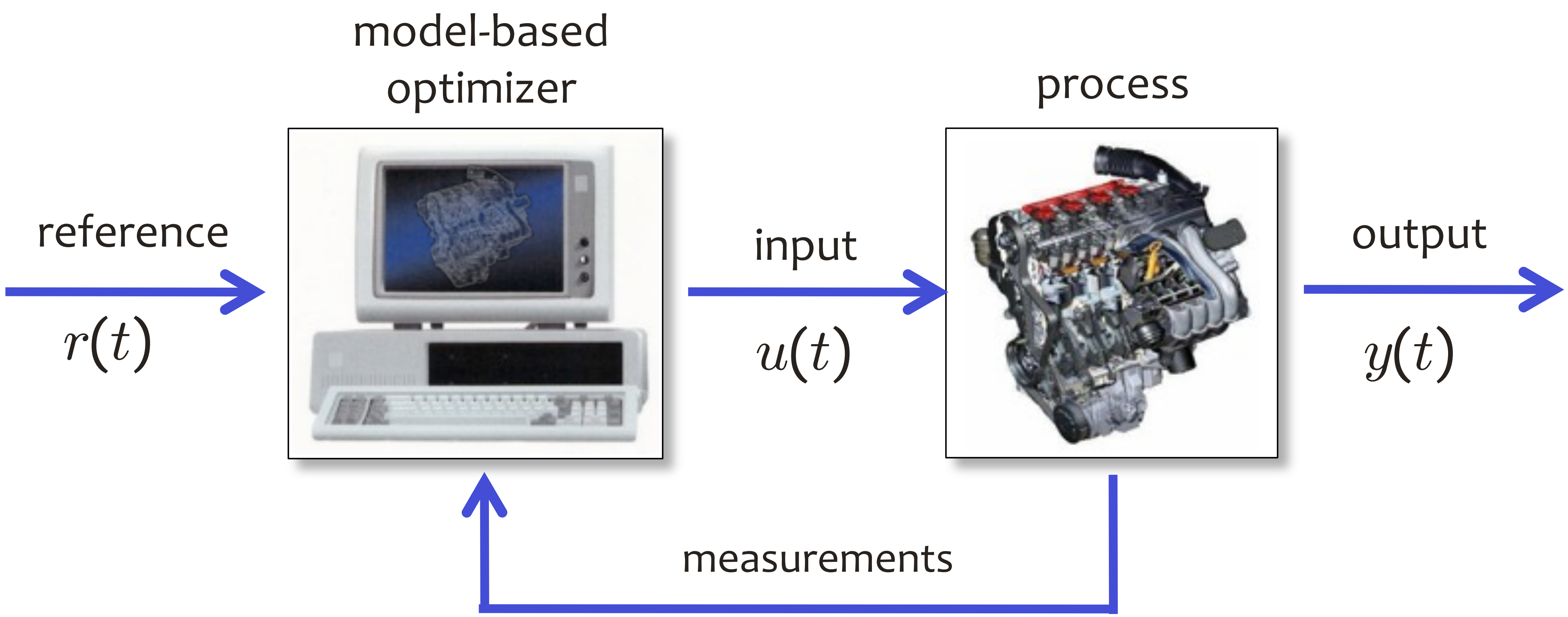
Model predictive control

Prof. Alberto Bemporad

University of Trento

Academic year 2009-2010

Model Predictive Control (MPC)

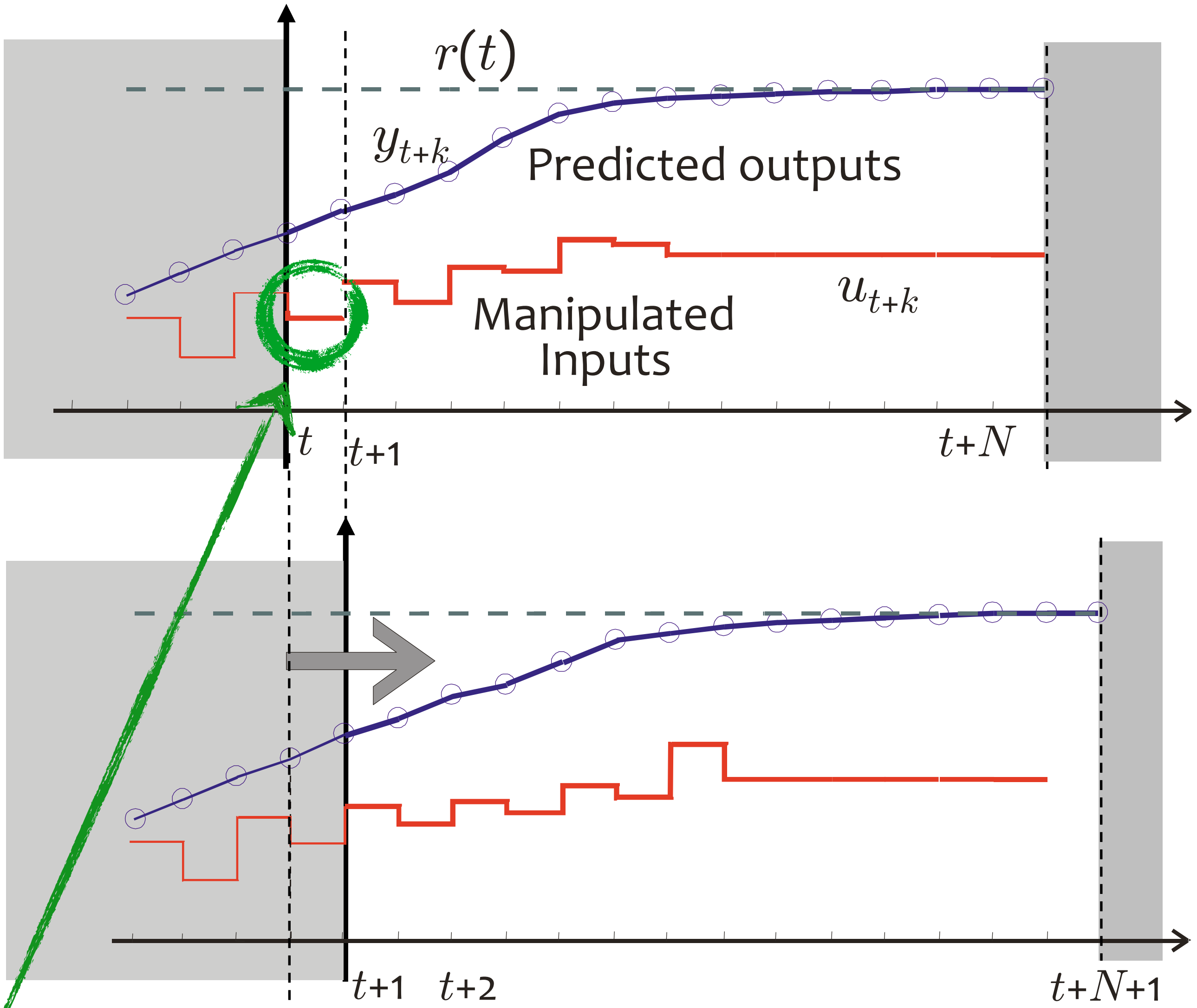


Use a dynamical ***model*** *of the process to* ***predict*** *its future evolution and optimize the* ***control*** *signal*

Receding horizon philosophy

- At time t: solve an **optimal control** problem over a finite future horizon of N steps:

$$\begin{aligned}\min_{z} \quad & \sum_{k=0}^{N-1} \|y_{t+k} - r(t)\|^2 + \rho\|u_{t+k}\|^2 \\ \text{s.t.} \quad & x_{t+k+1} = f(x_{t+k}, u_{t+k}) \\ & y_{t+k} = g(x_{t+k}) \\ & u_{\min} \leq u_{t+k} \leq u_{\max} \\ & y_{\min} \leq y_{t+k} \leq y_{\max} \\ & x_t = x(t),\ \ k = 0, \dots, N-1\end{aligned}$$

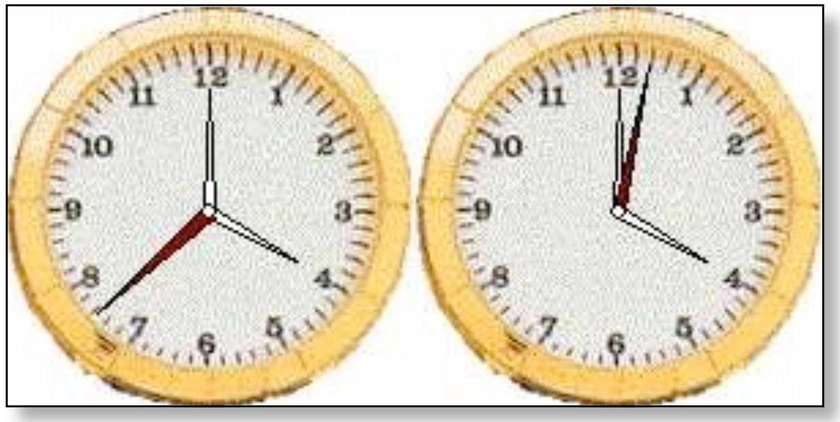


- Only apply the first optimal move $u^*(t)$
- At time $t+1$: Get new measurements, repeat the optimization. And so on ...

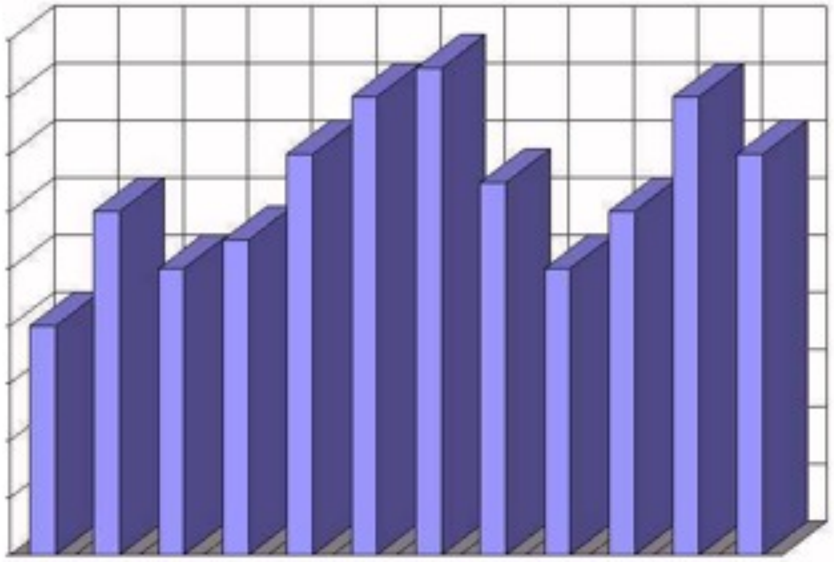
Advantage of repeated on-line optimization: **FEEDBACK !**

Receding Horizon - Examples

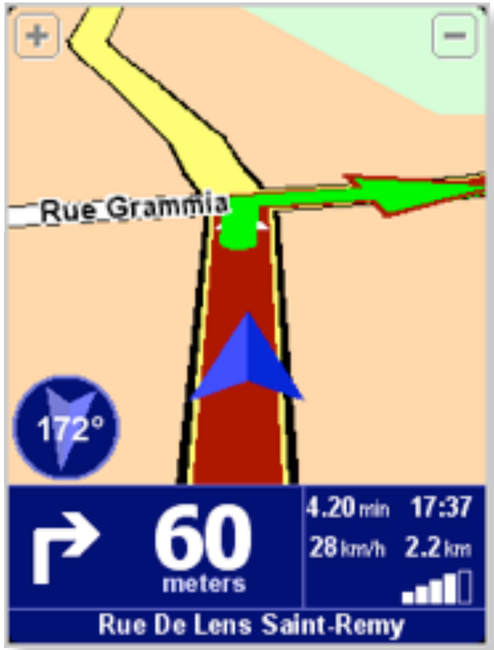
- MPC is like **playing chess** !



- "Rolling horizon" policies are also used frequently **in finance**



Receding Horizon - Examples

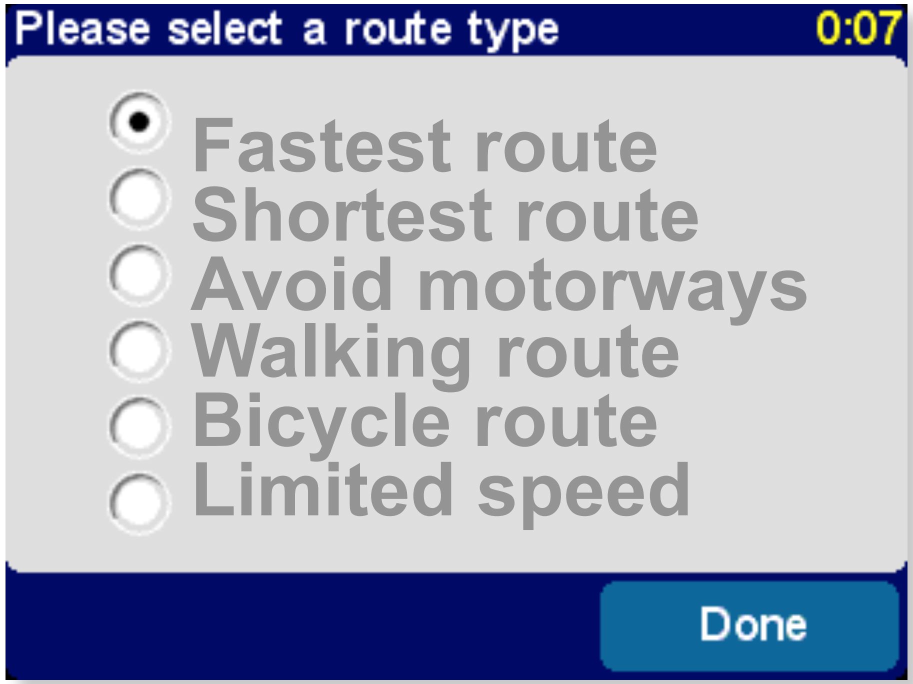


- **prediction model** how vehicle moves on the map
- **constraints** drive on roads, respect one-way roads, etc.
- **disturbances** mainly driver's inattention !

x = GPS position

u = navigation commands

- **set point** desired location
- **cost function** minimum time, minimum distance, etc.



- **receding horizon mechanism**

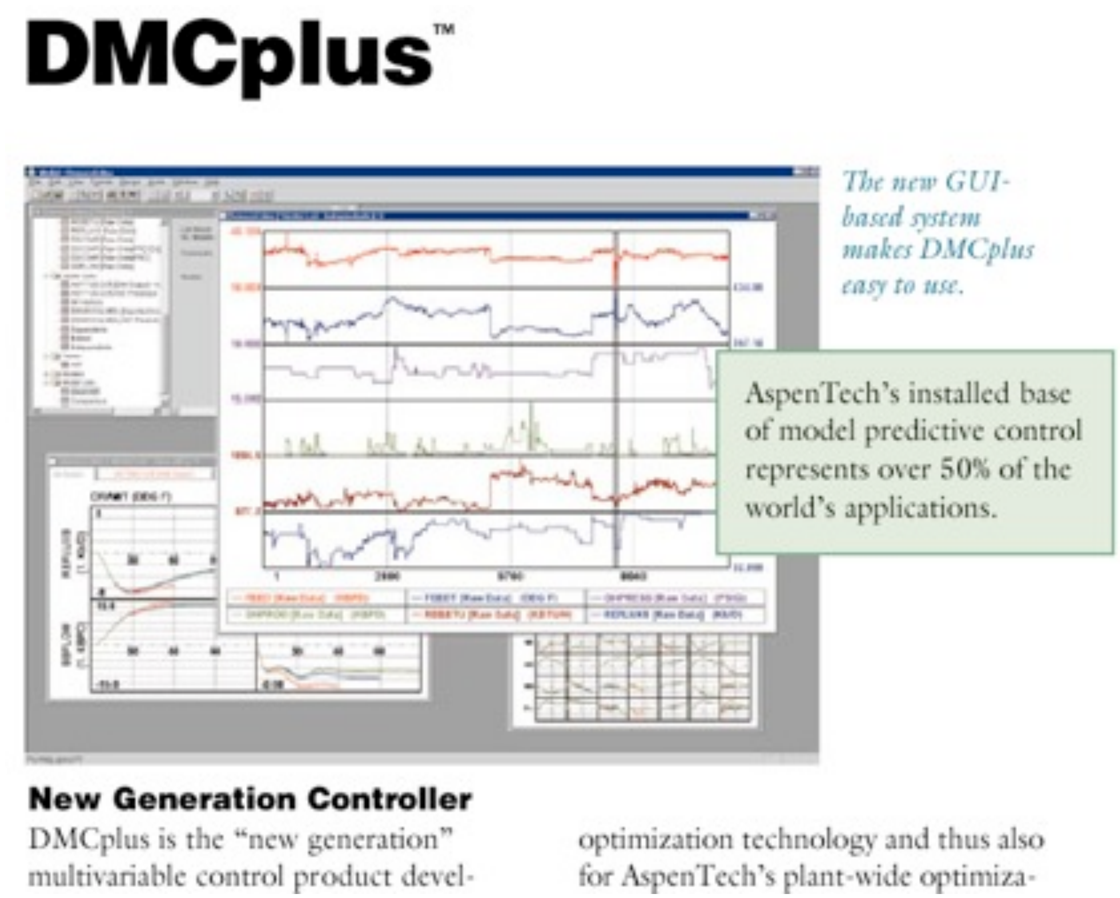
  event-based
  (optimal route re-planned when path is lost)

MPC in Industry

- History: 1979 Dynamic Matrix Control (DMC) by Shell (Motivation: multivariable, constrained)

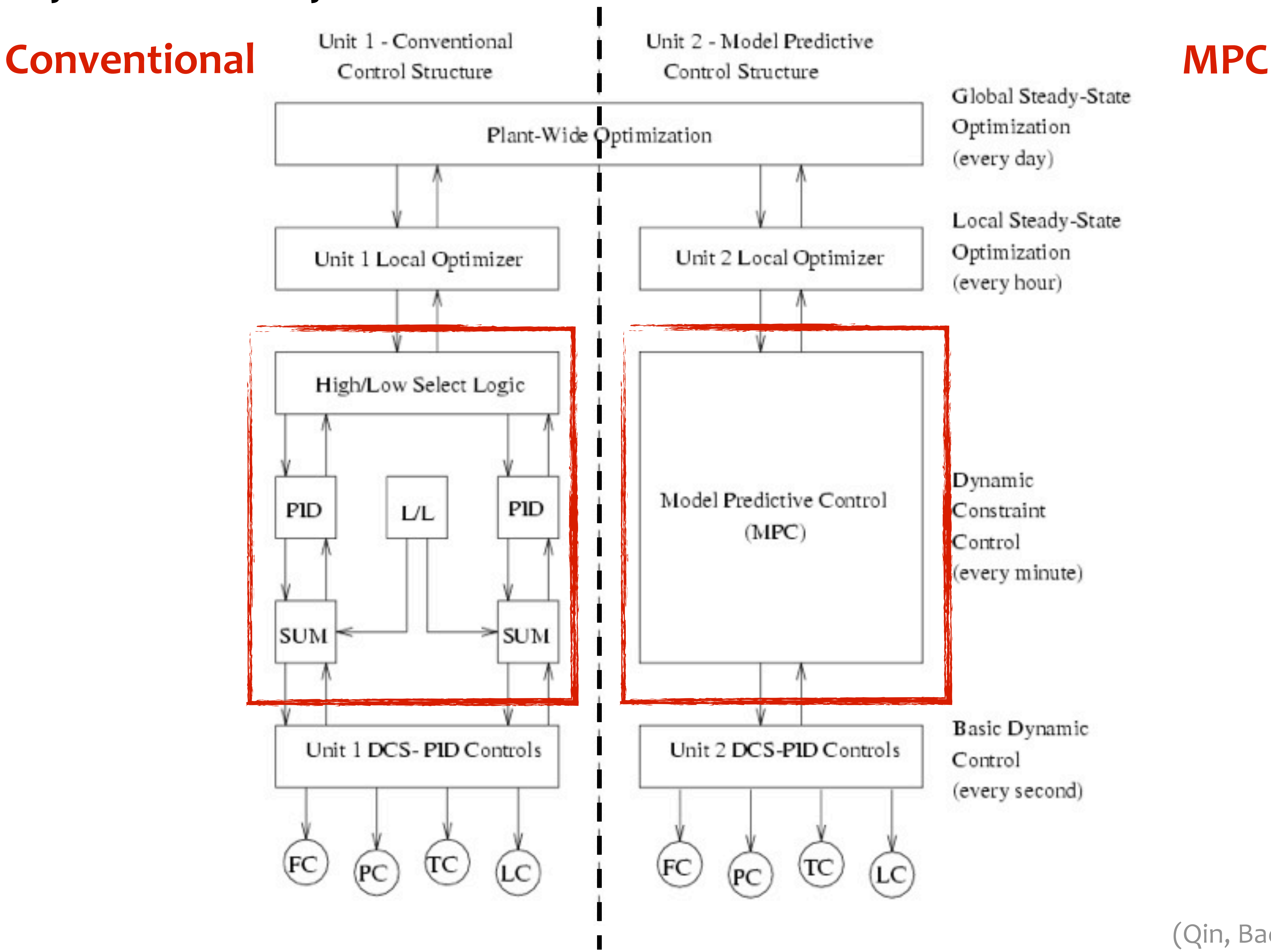
- Present Industrial Practice
  - linear impulse/step response models
  - sum of squared errors objective function
  - executed in supervisory mode

- Particularly suited for problems with
  - many inputs and outputs
  - constraints on inputs, outputs, states
  - varying objectives and limits on variables (e.g. because of faults)



MPC in Industry

Hierarchy of control system functions:



(Qin, Badgewell, 1997)

MPC in Industry

Area	Aspen Technology	Honeywell Hi-Spec	Adersa[b]	Invensys	SGS[c]	Total
Refining	1200	480	280	25		1985
Petrochemicals	450	80	—	20		550
Chemicals	100	20	3	21		144
Pulp and paper	18	50	—	—		68
Air & Gas	—	10	—	—		10
Utility	—	10	—	4		14
Mining/Metallurgy	8	6	7	16		37
Food Processing	—	—	41	10		51
Polymer	17	—	—	—		17
Furnaces	—	—	42	3		45
Aerospace/Defense	—	—	13	—		13
Automotive	—	—	7	—		7
Unclassified	40	40	1045	26	450	1601
Total	1833	696	1438	125	450	4542
First App.	DMC:1985 IDCOM-M:1987 OPC:1987	PCT:1984 RMPCT:1991	IDCOM:1973 HIECON:1986	1984	1985	
Largest App.	603 × 283	225 × 85	—	31 × 12	—	

(snapshot survey conducted in mid-1999) (Qin, Badgewell, 2003)

“For us multivariable control is predictive control ”

Tariq Samad, *Honeywell* (past president of the IEEE Control System Society) (1997)

Unconstrained Optimal Control

- Linear model:

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

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

- Goal: find $u^*(0),\ u^*(1),\ \ldots,\ u^*(N-1)$

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

$$U = [u'(0)\ u'(1)\ \ldots\ u'(N-1)]'$$

$u^*(0),\ u^*(1),\ \ldots,\ u^*(N-1)$ is the input sequence that steers the state to the origin "optimally"

Unconstrained Optimal Control

$$J(x(0),U) \;=\; \frac{1}{2}U'HU + x'(0)FU + \frac{1}{2}x'(0)Yx(0)$$

$$U = [u'(0)\ u'(1)\ \ldots\ u'(N-1)]'$$

The optimum is obtained by zeroing the gradient

$$\nabla_U J(x(0),U) = HU + F'x(0) = 0$$

and hence

$$U^* = \begin{bmatrix} u^*(0) \\ u^*(1) \\ \vdots \\ u^*(N-1) \end{bmatrix} = -H^{-1}F'x(0)$$

batch least squares

Alternative approach: use dynamic programming to find U^* (Riccati iterations)

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

- Constrained optimal control problem (quadratic performance index):

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

$$\text{s.t.} \quad \begin{aligned} & u_{\min} \leq u(k) \leq u_{\max},\ k = 0, \dots, N-1 \\ & y_{\min} \leq y(k) \leq y_{\max},\ k = 1, \dots, N \end{aligned}$$

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

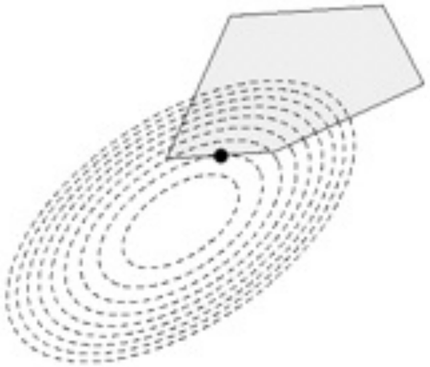
Constrained Optimal Control

- Optimization problem:

$$V(x(0)) = \frac{1}{2}x'(0)Yx(0) + \min_{U} \ \frac{1}{2}U'HU + x'(0)FU \quad \text{(quadratic)}$$

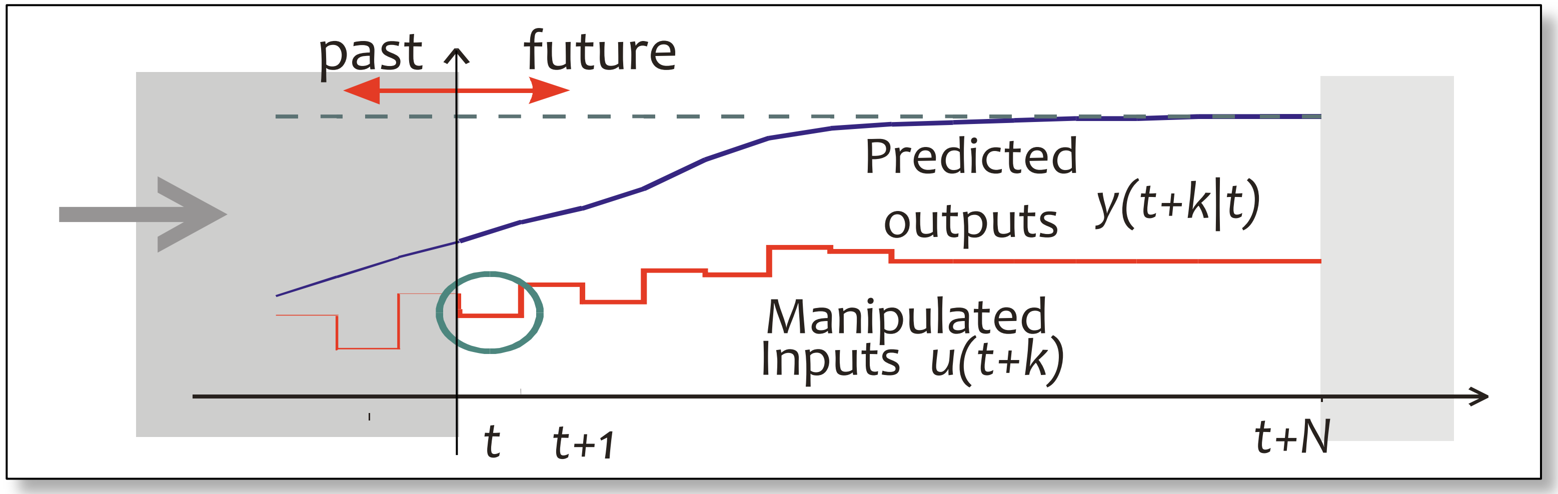
$$\text{s.t.} \quad GU \leq W + Sx(0) \quad \text{(linear)}$$

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

Linear MPC Algorithm



At time t:

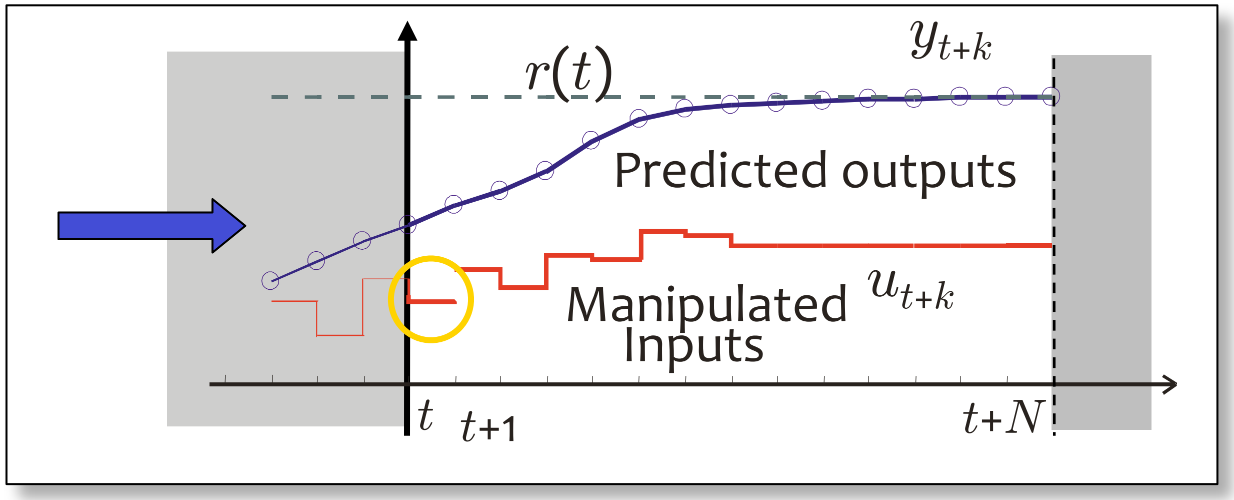
- Get/estimate the current state $x(t)$
- Solve the QP problem

$$\begin{aligned} \min_U \quad & \tfrac{1}{2}U'HU + x'(t)FU \\ \text{s.t.} \quad & GU \leq W + Sx(t) \end{aligned}$$

and let $U=\{u^*(0),\ldots,u^*(N\text{-}1)\}$ be the solution
(=finite-horizon constrained open-loop optimal control)

- Apply only $u(t) = u^*(0)$ and discard the remaining optimal inputs
- Repeat optimization at time t+1. And so on ...

Unconstrained Linear MPC



- Assume no constraints
- Problem to solve on-line:

$$\min_U \quad J(x(t),U) = \frac{1}{2}U'HU + x'(t)FU$$

- Solution: $\nabla_U J(x(t),U) = HU + F'x(t) = 0$

$$U = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

$$\longrightarrow \quad U^* = -H^{-1}F'x(t)$$

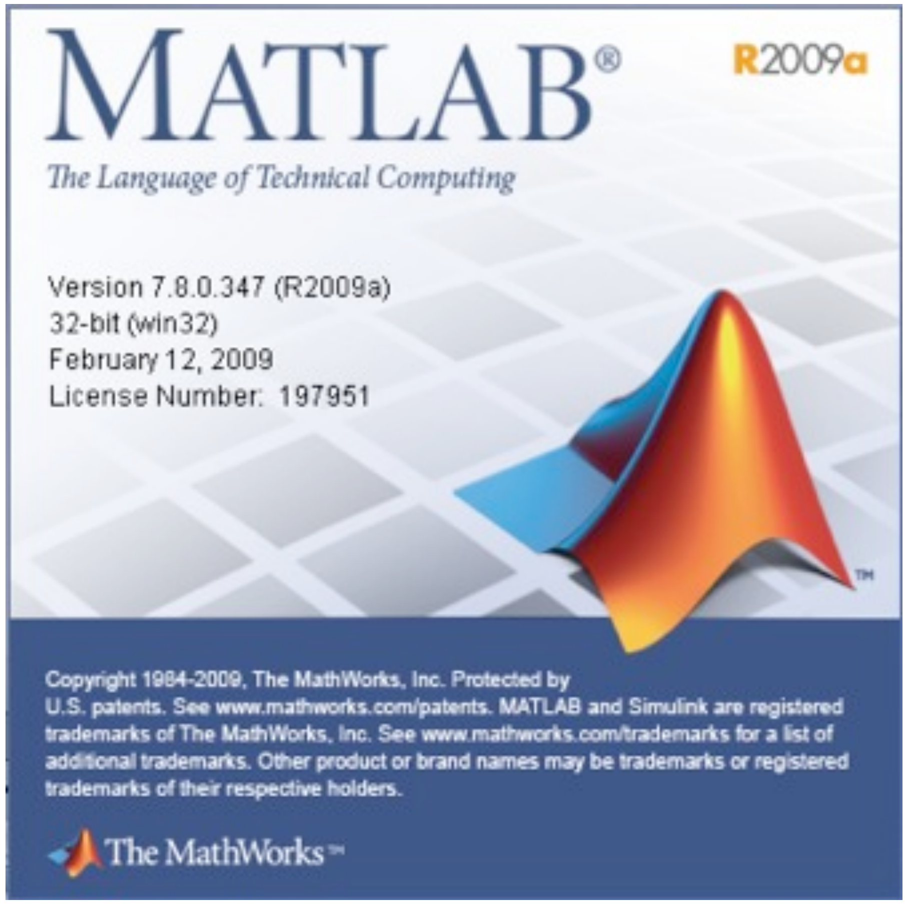
$$\Rightarrow \quad u(t) = -[I \;\; 0 \;\; \ldots \;\; 0]H^{-1}Fx(t) \triangleq Kx(t)$$

Unconstrained linear MPC is nothing else than a standard linear state-feedback law !

Model Predictive Control Toolbox 3.0

(Bemporad, Ricker, Morari, 1998-today)

- **MPC Toolbox 3.0** (The Mathworks, Inc.)
  - Object-oriented implementation (MPC object)
  - MPC Simulink Library
  - MPC Graphical User Interface
  - RTW extension (code generation) [xPC Target, dSpace, etc.]
  - Linked to OPC Toolbox v2.0.1



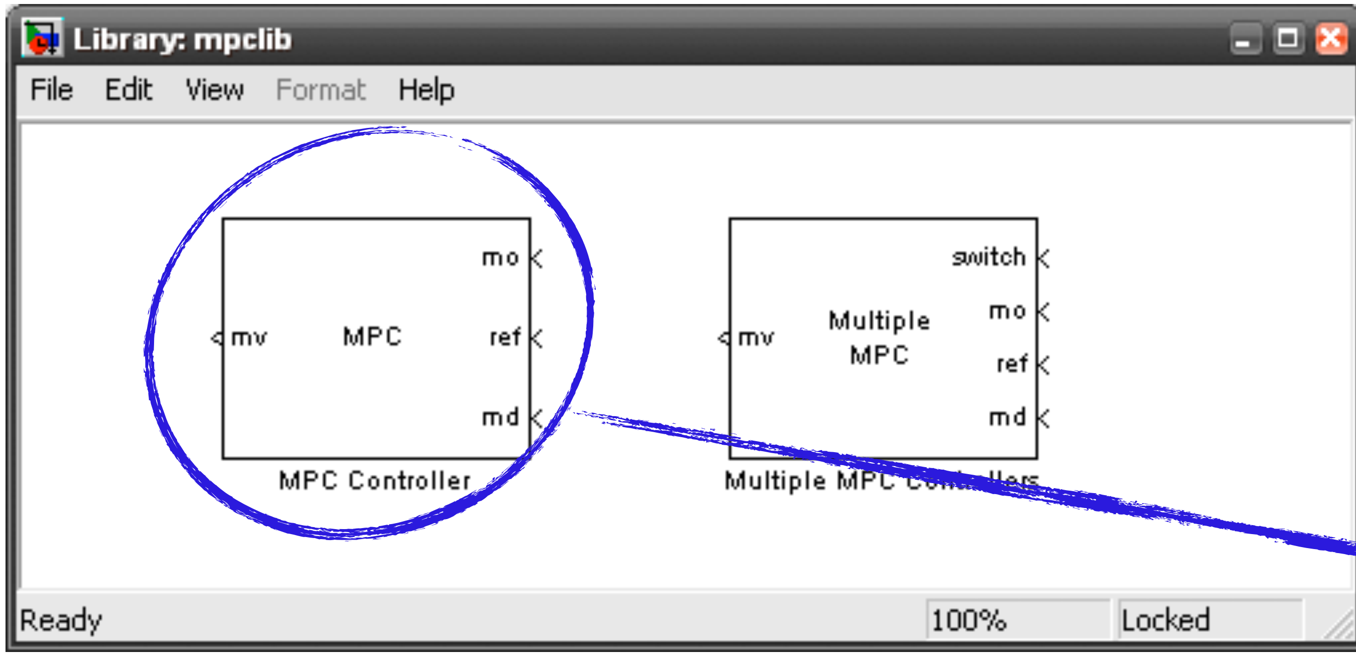
Complete solution for linear MPC design based on on-line QP

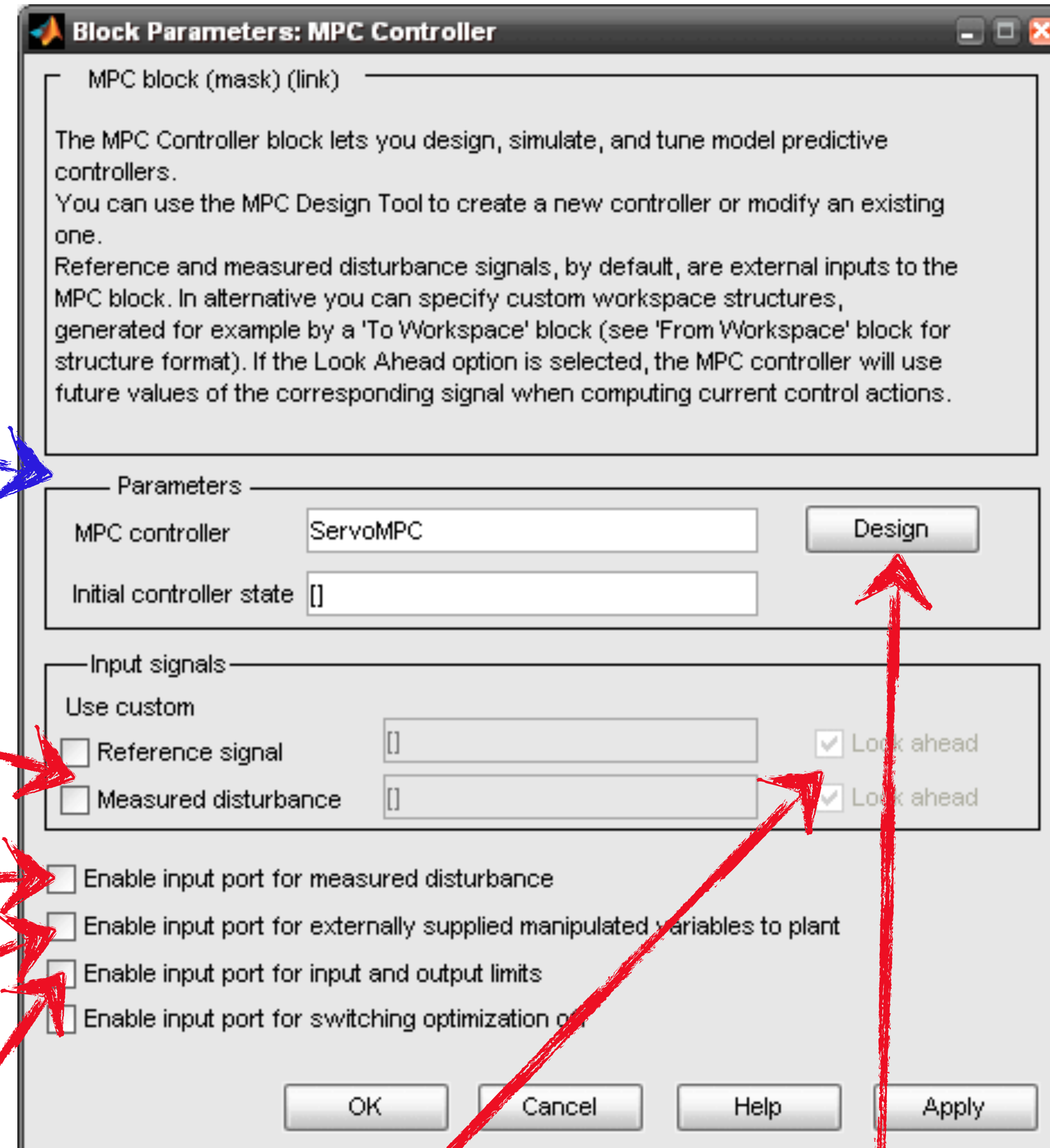
http://www.mathworks.com/products/mpc/

Model Predictive Control Toolbox 3.0

- Several **linear MPC** design features available:
  - preview on references/measured disturbances
  - time-varying weights and constraints, non-diagonal weights
  - integral action for offset-free tracking
  - soft constraints
  - linear time-varying models (*to appear in next release*)
- Prediction models generated by Identification Toolbox supported
- Automatic linearization of prediction models from Simulink diagrams
- Linear stability/frequency analysis of closed-loop (inactive constraints)
- Very fast command-line closed-loop simulation (C-code), with very versatile simulation options (e.g. analysis of model mismatch effects)

MPC Simulink Library





read reference and/or measured disturbance signals from workspace

measured disturbances from simulation diagram

feed actuator commands (for bumpless transfer)

input and output limits change during simulation

anticipative action

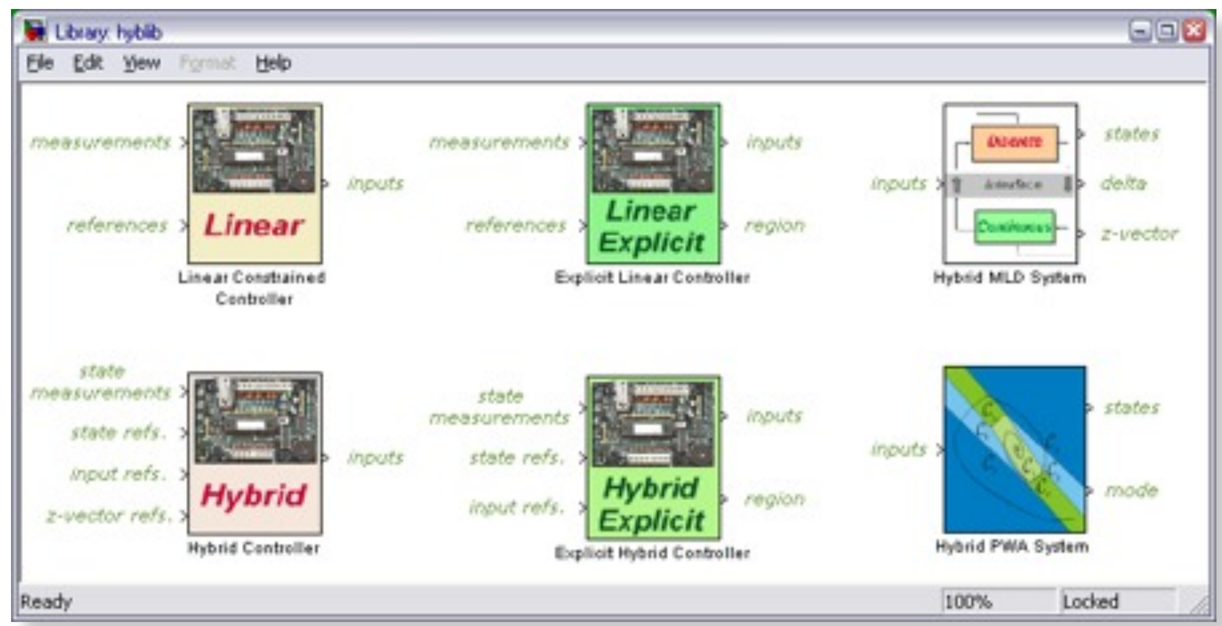
open MPC GUI for design

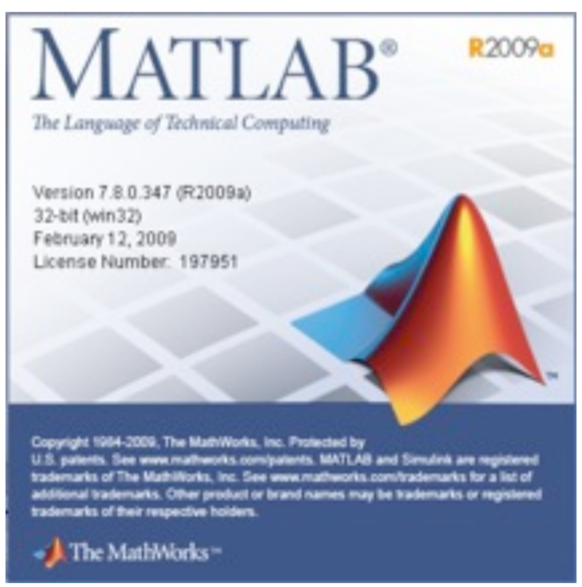
(Bemporad, 2003-today)

Hybrid Toolbox for MATLAB

Features:

- Hybrid models: design, simulation, verification
- Control design for linear systems w/ constraints and hybrid systems (on-line optimization via QP/MILP/MIQP)
- Explicit MPC control (via multi-parametric programming)
- C-code generation
- Simulink library





3000+ download requests
(since October 2004)

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

Double Integrator Example

- System: $y(\tau) = \frac{1}{s^2}u(\tau)$ → sampling + ZOH T_s=1 s →

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

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

- Control objective: min $\left(\sum_{k=0}^{1} y^2(k) + \frac{1}{10}u^2(k) \right) + x'(2) \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix} \right] x(2)$

- Optimization problem matrices:

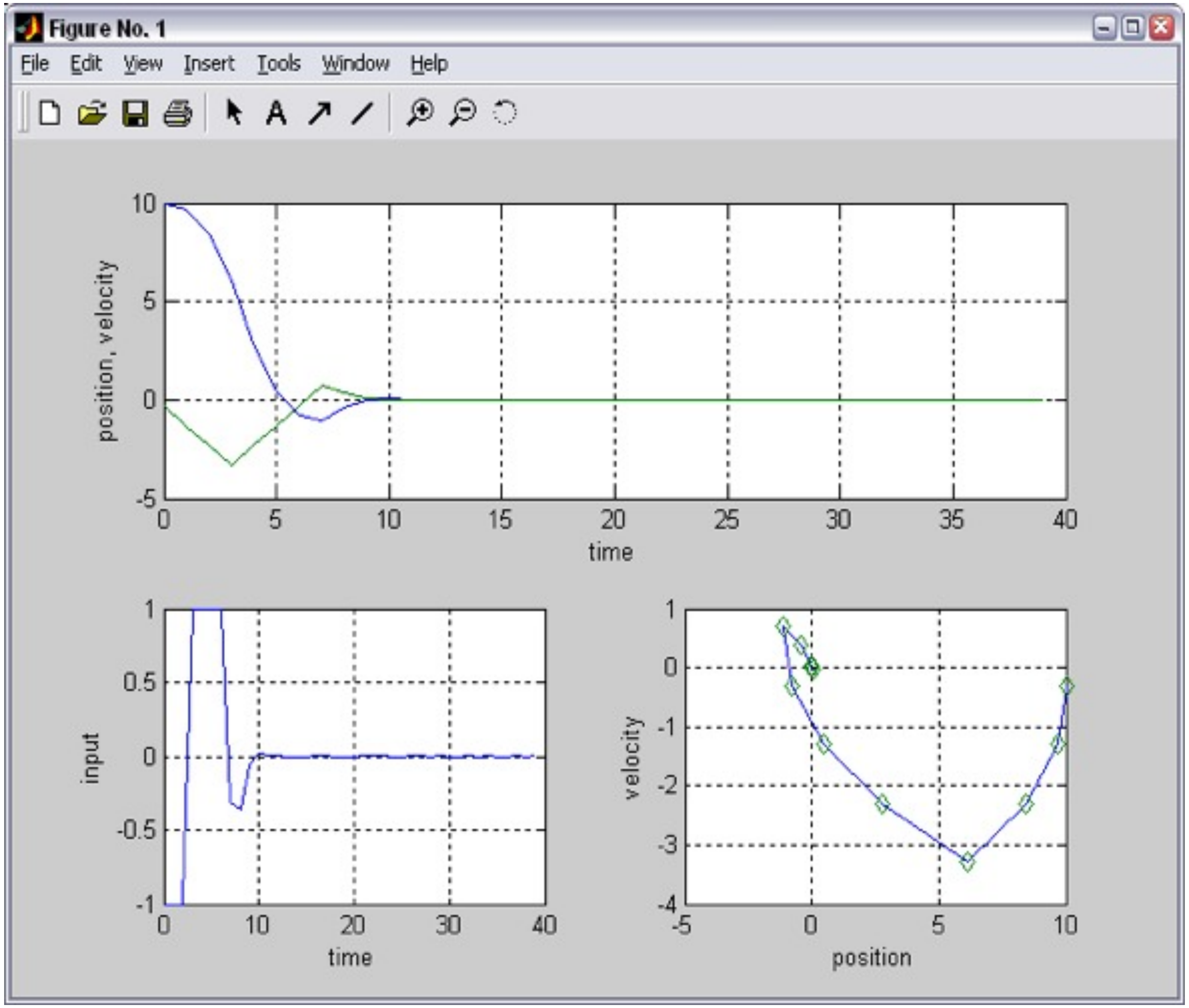
$$H = \begin{bmatrix} 4.2 & 2 \\ 2 & 2.2 \end{bmatrix}, \quad F = \begin{bmatrix} 2 & 0 \\ 6 & 2 \end{bmatrix}, \quad Y = \begin{bmatrix} 4 & 6 \\ 6 & 12 \end{bmatrix}$$

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

cost: $\frac{1}{2}U'HU + x'(t)FU + \frac{1}{2}x'(t)Yx(t)$

constraints: $GU \leq W + Sx(t)$

Double Integrator Example



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

see also **mpcdoubleint.m** (MPC-Tbx)

Double Integrator Example

- Add a state constraint:

$$x_2(k) \geq -1, \; k = 1$$

- Optimization problem matrices:

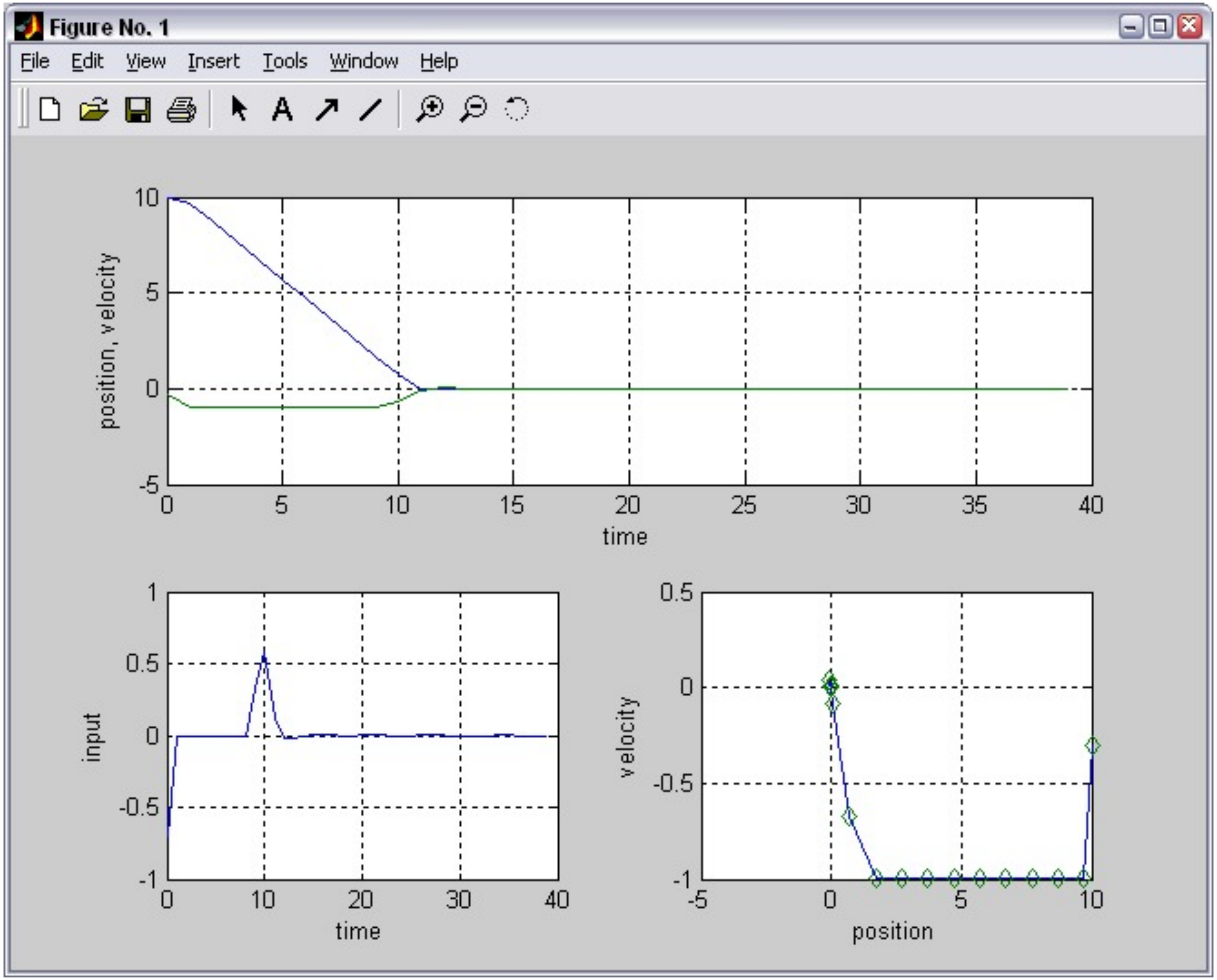
$$H = \begin{bmatrix} 4.2 & 2 \\ 2 & 2.2 \end{bmatrix}, \; F = \begin{bmatrix} 2 & 0 \\ 6 & 2 \end{bmatrix}, \; Y = \begin{bmatrix} 4 & 6 \\ 6 & 12 \end{bmatrix}$$

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

cost: $\frac{1}{2}U'HU + x'(t)FU + \frac{1}{2}x'(t)Yx(t)$

constraints: $GU \leq W + Sx(t)$

Double Integrator Example



Linear MPC - Tracking

- Objective: make the output $y(t)$ track a reference signal $r(t)$

- Idea: parameterize the problem using input increments

$$\Delta u(t) = u(t) - u(t-1) \quad \Longrightarrow \quad u(t) = u(t-1) + \Delta u(t)$$

- Extended system: let $x_u(t)=u(t\text{-}1)$

$$\begin{cases} x(t+1) = Ax(t) + Bu(t-1) + B\Delta u(t) \\ x_u(t+1) = x_u(t) + \Delta u(t) \end{cases}$$

$$\Downarrow$$

$$\begin{cases} \begin{bmatrix} x(t+1) \\ x_u(t+1) \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} x(t) \\ x_u(t) \end{bmatrix} + \begin{bmatrix} B \\ I \end{bmatrix} \Delta u(t) \\ y(t) = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ x_u(t) \end{bmatrix} \end{cases}$$

Again a linear system with states $x(t)$, $x_u(t)$ and input $\Delta u(t)$

Linear MPC - Tracking

- Optimal control problem (quadratic performance index):

$$\min_{\Delta U} \sum_{k=0}^{N-1} \|W^y(y(k+1)-r(t))\|^2 + \|W^{\Delta u}\Delta u(k)\|^2$$

$$[\Delta u(k) \triangleq u(k) - u(k-1)]$$

$$\text{subj. to} \quad u_{\min} \le u(k) \le u_{\max}, \quad k = 0, \ldots, N-1$$

$$\Delta u_{\min} \le \Delta u(k) \le \Delta u_{\max}, \quad k = 0, \ldots, N-1$$

$$y_{\min} \le y(k) \le y_{\max}, \quad k = 1, \ldots, N$$

optimization vector:

$$\Delta U = \begin{bmatrix} \Delta u(0) \\ \Delta u(1) \\ \vdots \\ \Delta u(N-1) \end{bmatrix}$$

- Note: $\|Wz\|^2 = (Wz)'(Wz) = z'(W'W)z = z'Qz$

⇒ same formulation as before (W=Cholesky factor of weight matrix Q)

- Optimization problem:

Convex Quadratic Program (QP)

$$\min_{\Delta U} \quad J(\Delta U, x(t)) = \frac{1}{2}\Delta U' H \Delta U + [x'(t) \;\; r'(t) \;\; u'(t-1)] F \Delta U$$

$$\text{s.t.} \quad G\Delta U \le W + S \begin{bmatrix} x(t) \\ r(t) \\ u(t-1) \end{bmatrix}$$

MPC vs. Conventional Control

Single input/single output control loop w/ constraints:

equivalent performance can be obtained with other simpler control techniques (e.g.: PID + anti-windup)

HOWEVER

MPC allows (in principle) UNIFORMITY
(i.e. same technique for wide range of problems)

- reduce training
- reduce cost
- easier design maintenance

Satisfying control specs and walking on water is similar …

both are not difficult if frozen !

MPC Features

- Multivariable constrained “non-square” systems
(i.e. #inputs and #outputs are different)

- Delay compensation

- Anticipative action for future reference changes

- “Integral action”, i.e. no offset for step-like inputs

Price to pay:

- Substantial on-line computation

- For simple small/fast systems other techniques dominate
(e.g. PID + anti-windup)

- New possibility for MPC: *explicit* piecewise affine solutions (Bemporad et al., 2002)

MPC Theory

- Historical Goal: Explain the success of DMC
- Present Goal: Improve, simplify, and extend industrial algorithms
- Areas:
  - Linear MPC: linear model
  - Nonlinear MPC: nonlinear model
  - Robust MPC: uncertain (linear) model
  - Hybrid MPC: model integrating logic, dynamics, and constraints
- Issues:
  - Feasibility
  - Stability (Convergence)
  - Computations

(Mayne, Rawlings, Rao, Scokaert, 2000)

Convergence Result

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

$$\begin{array}{ll} \min_U J(U,x(t)) = & \sum_{k=0}^{N-1} \left\{x'(t+k|t)Qx(t+k|t)+u'(t+k)Ru(t+k)\right\} \\ \text{subj. to} & y_{min} \leq y(t+k) \leq y_{max} \\ & u_{min} \leq u(t+k) \leq u_{max} \\ & x(t+N|t) = 0 \end{array}$$

Assume that the optimization problem is feasible at time $t=0$*. Then, for all* $R>0$, $Q>0$,

$$\begin{aligned} \lim_{t\to\infty} x(t) &= 0, \\ \lim_{t\to\infty} u(t) &= 0, \end{aligned}$$

and the constraints are satisfied at all time instants $t \geq 0$.

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

Proof: Use value function as Lyapunov function

Convergence Proof

- 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

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

- $V(t)$ 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 $x'(t)Qx(t), u'(t)Ru(t) \to 0$
- Since $R, Q > 0$, $u(t), x(t) \to 0$

Global optimum is not needed to prove convergence !

MPC and LQR

- Consider the MPC control law:

Jacopo Francesco Riccati (1676 - 1754)

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

$R = R' > 0$, $Q = Q' \geq 0$, and P satisfies the Riccati equation

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

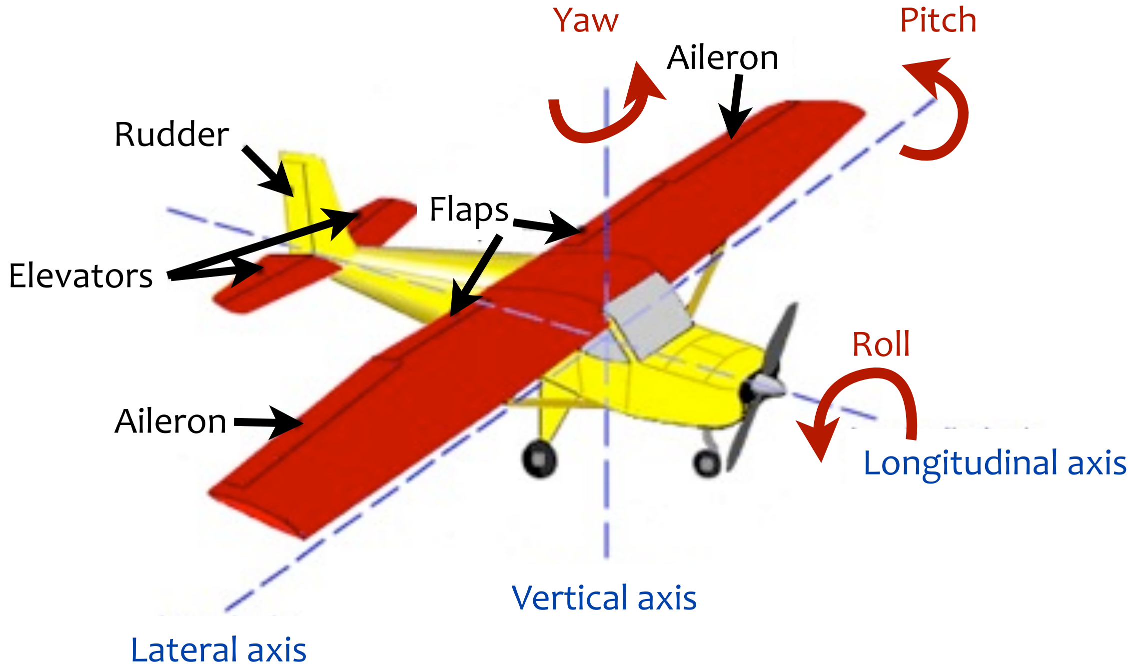
(Unconstrained) MPC = LQR

Example: AFTI-16

- Linearized model:

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

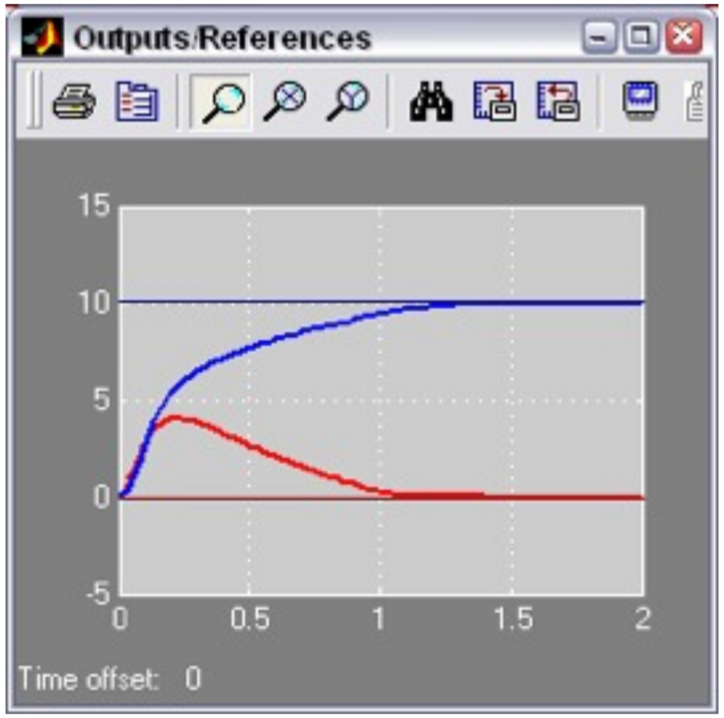
- Inputs: elevator and flaperon angle
- Outputs: attack and pitch angle
- Sampling time: $T_s = .05$ s (+ zero-order hold)
- Constraints: max 25° on both angles
- Open-loop response: unstable
  (open-loop poles: $-7.6636, -0.0075 \pm 0.0556j, 5.4530$)

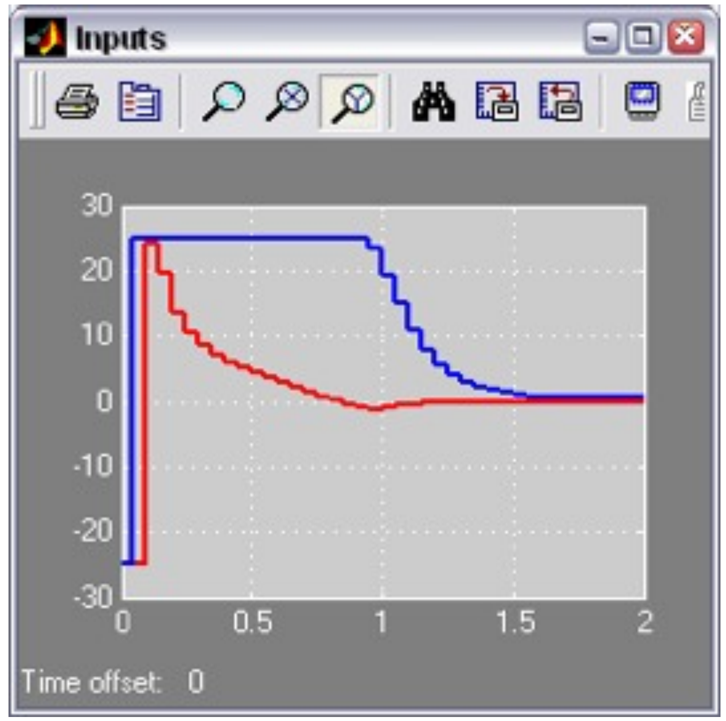


go to demo /**demos/linear/afti16.m** (Hyb-Tbx)

afti16.m (MPC-Tbx)

Example: AFTI-16

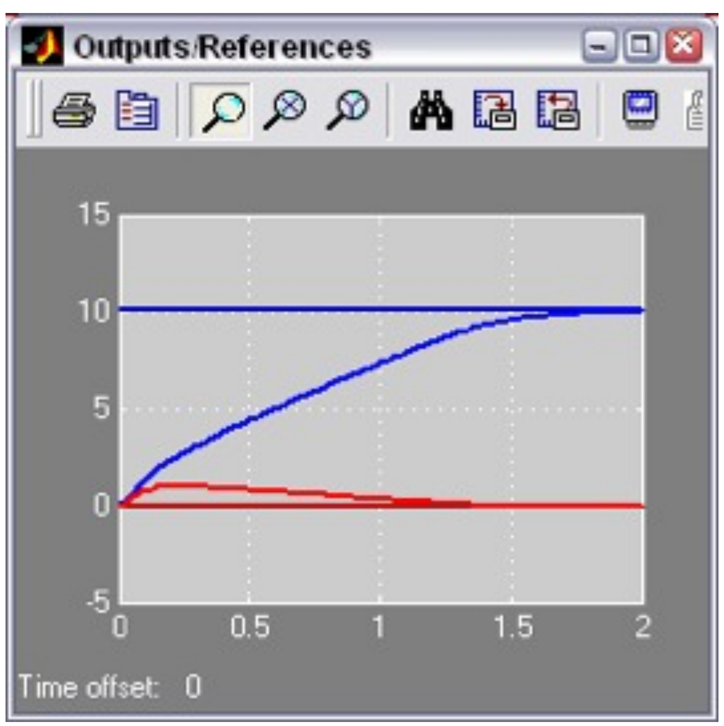


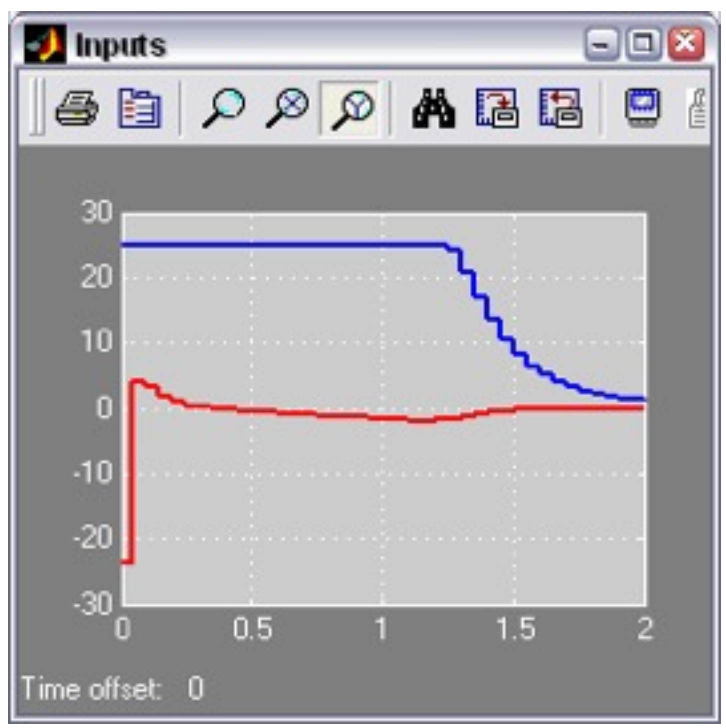


$N_y = 10,\ N_u = 3,$

$w_y = \{10, 10\},\ w_{\delta u} = \{.01, .01\},$

$u_{\min} = -25^\circ,\ u_{\max} = 25^\circ$



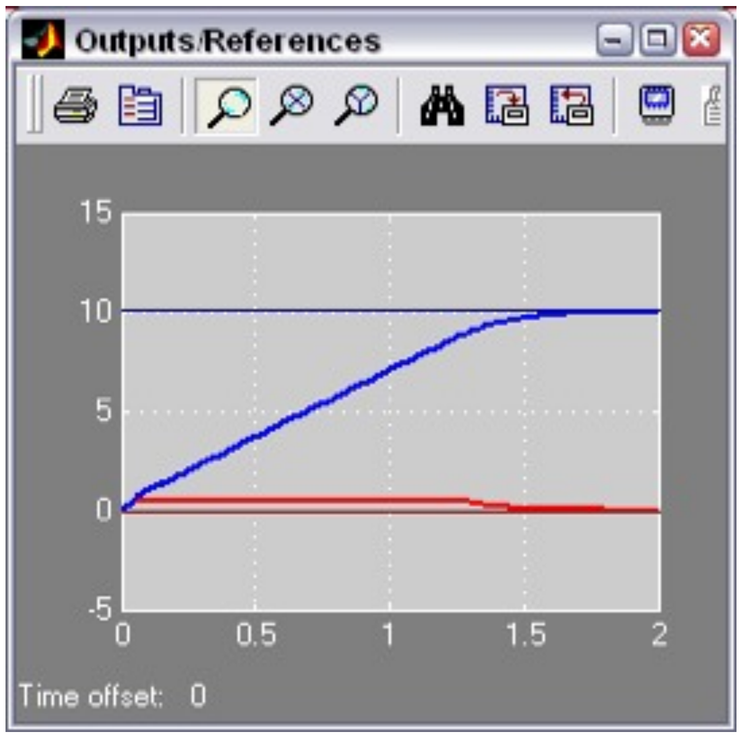


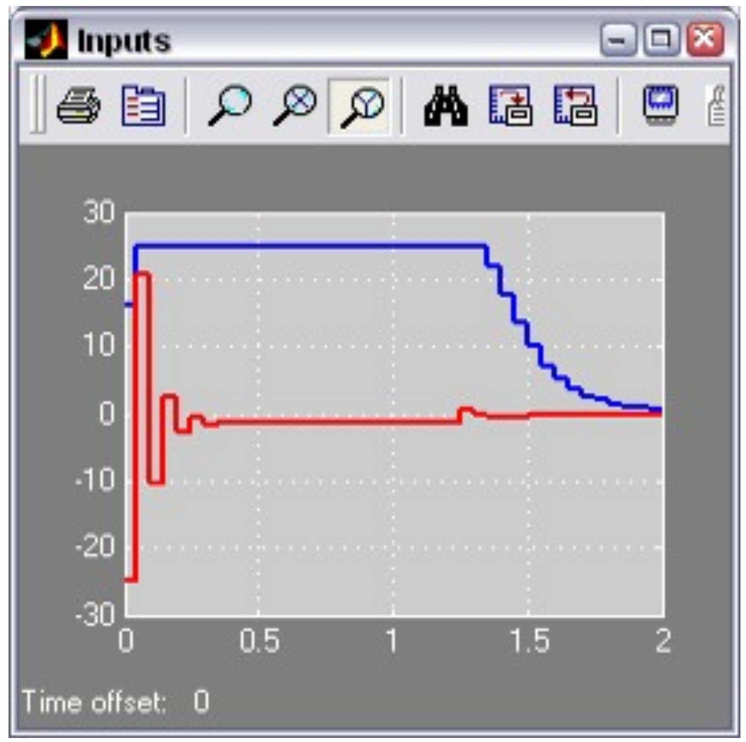
$N_y = 10,\ N_u = 3,$

$w_y = \{100, 10\},\ w_{\delta u} = \{.01, .01\},$

$u_{\min} = -25^\circ,\ u_{\max} = 25^\circ$

Example: AFTI-16





$N_y = 10, \; N_u = 3,$

$w_y = \{10, 10\}, \; w_{\delta u} = \{.01, .01\},$

$u_{\min} = -25^\circ, \; u_{\max} = 25^\circ,$

$y_{1,\min} = -0.5^\circ, \; y_{1,\max} = 0.5^\circ$

Tuning Guidelines

$$
\begin{aligned}
\min_{\Delta U} \quad & \sum_{k=0}^{N-1} \|W^y(y(t+k+1|t)-r(t))\|^2 + \|W^{\Delta u}\Delta u(t)\|^2 \\
\text{subj. to} \quad & u_{\min} \leq u(t+k) \leq u_{\max},\ k=0,\dots,N-1 \\
& \Delta u_{\min} \leq \Delta u(t+k) \leq \Delta u_{\max},\ k=0,\dots,N_u-1 \\
& y_{\min} \leq y(t+k|t) \leq y_{\max},\ k=1,\dots,N \\
& \Delta u(t+k)=0,\ k=N_u,\dots,N-1
\end{aligned}
$$

- **Weights**: the larger the ratio $W^y/W^{\Delta u}$ the more aggressive the controller
- **Input horizon**: the larger N_u, the more "optimal" but the more complex the controller
- **Output horizon**: the smaller N, the more aggressive the controller
- **Limits**: controller less aggressive if $\Delta u_{\min}$, $\Delta u_{\max}$ are small

Always try to set N_u as small as possible !

Conclusions on MPC

- Main **pros** of MPC:
  - Can handle *nonlinear/switching/MIMO* dynamics with *delays*
  - Can enforce *constraints* on inputs and outputs
  - Performance is *optimized*
  - Systematic design approach, MPC designs are *easy to maintain*
  - *MATLAB tools* exist to assist the design and for code generation

- Main **cons** of MPC:
  - Requires a (simplified) *prediction model*, as every model-based technique
  - Needs full-state estimation (*observers*)
  - *Computation issues* more severe than in classical (linear) methods. This is partially mitigated by *explicit* reformulations of MPC
  - *Calibration* of MPC requires additional expertise (multiple tuning knobs)

- MPC is constantly spreading in industry (more powerful control units, more efficient numerical algorithms)

- Started in the 80‘s in the process industries, now reaching automotive, avionics aerospace, power systems, ...

Conclusions of the course

- Automatic control is an engineering discipline that is transversal (and helpful) to a wide variety of other disciplines
- Although a lot of industrial products would not work without feedback controllers, control suffers the fact of being a “hidden technology”
- Control engineering is well established in many areas (process industries, automotive, avionics, space, military, energy, naval, ...)
- The role of control engineering is steadily increasing in traditional but also in new application areas !

Master thesis projects on various control-related topics are available !

Italian-English Vocabulary

	
model predictive control	*controllo predittivo*
receding horizon control	*controllo a orizzonte recessivo*
quadratic programming	*programmazione quadratica*

Translation is obvious otherwise.