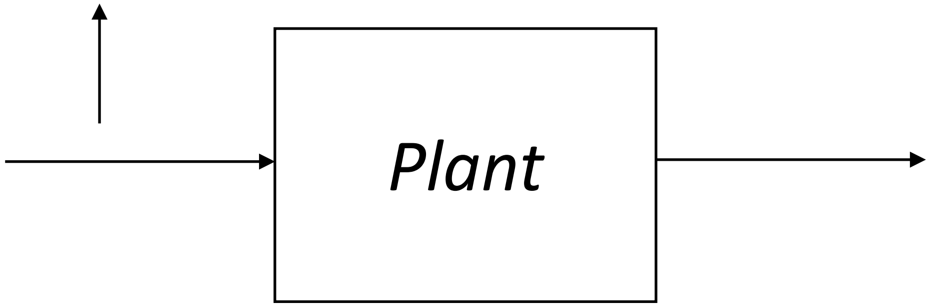
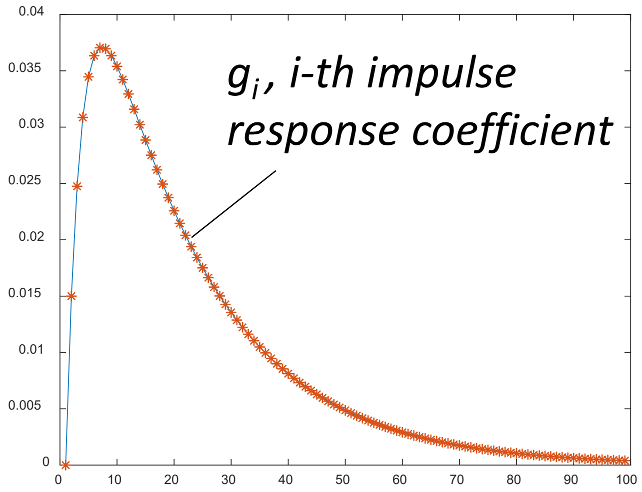
Advanced and Multivariable Control

Model Predictive Control – Part 2

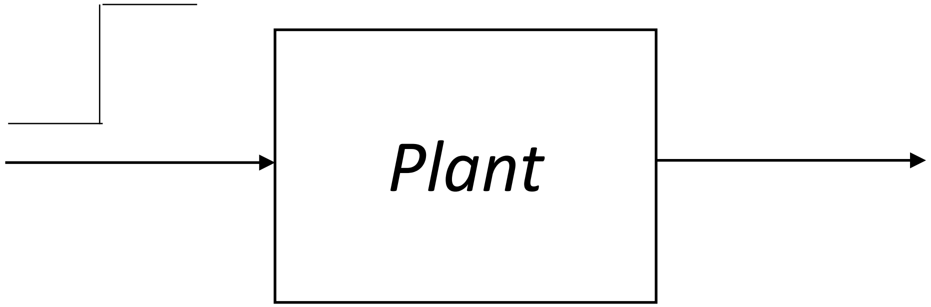
Riccardo Scattolini

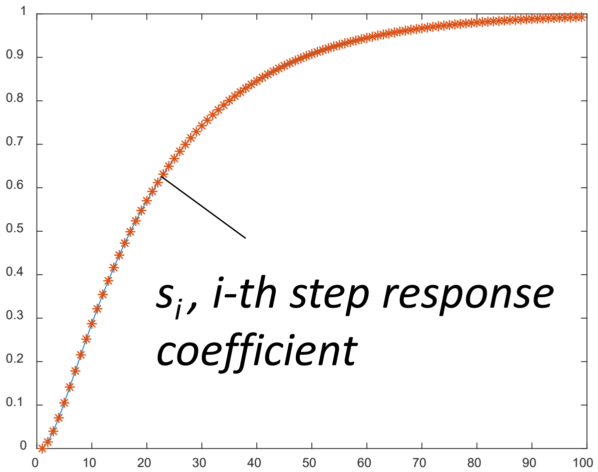
One of the main reasons of the success of MPC in the '80 was the possibility to use ***empirical models*** based on impulse or step responses. This is still a viable way today





The values of the output correspond to the impulse and step response coefficients (inputs of amplitude 1)

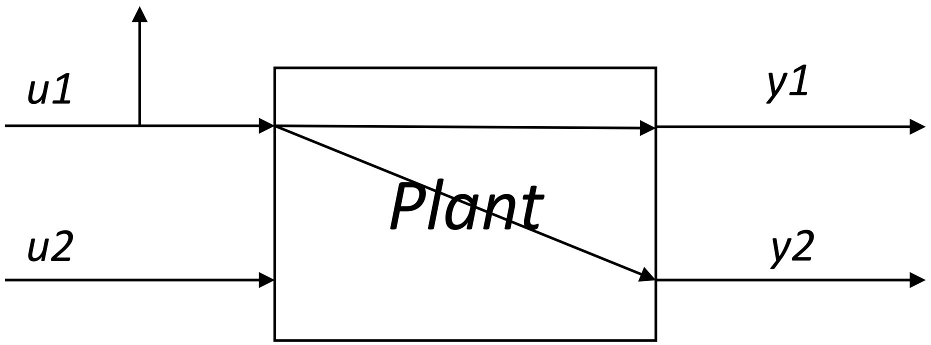


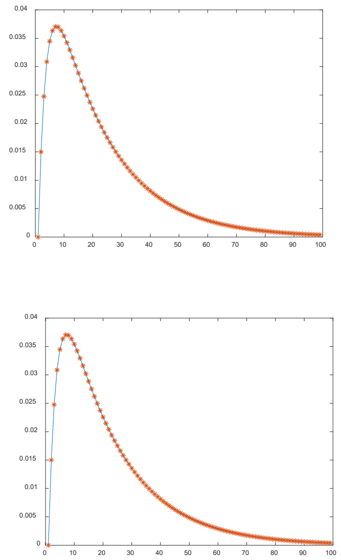


MIMO systems (2x2 case)

$$g_i = \begin{bmatrix} g_i^{11} & g_i^{12} \\ g_i^{21} & g_i^{22} \end{bmatrix}$$

Experiment 1

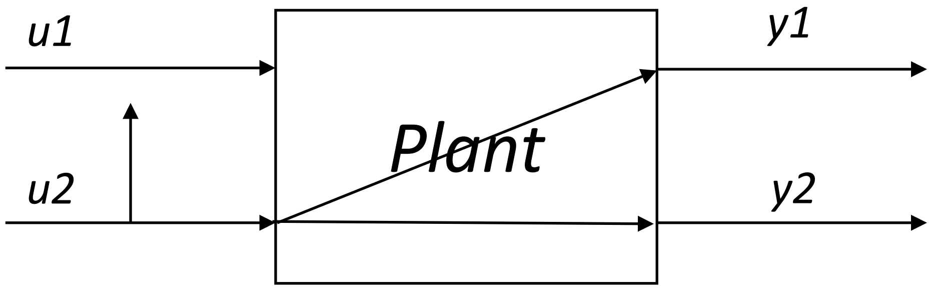


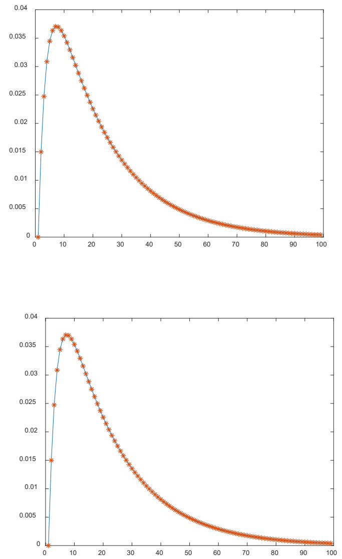


g_i^{11}

g_i^{21}

Experiment 2





g_i^{12}

g_i^{22}

Problems

The system must be brought to a stationary condition before the experiment. In large scale plants this is not so simple due to the presence of disturbances (maybe inner loops help, but be careful about what you are identifying)

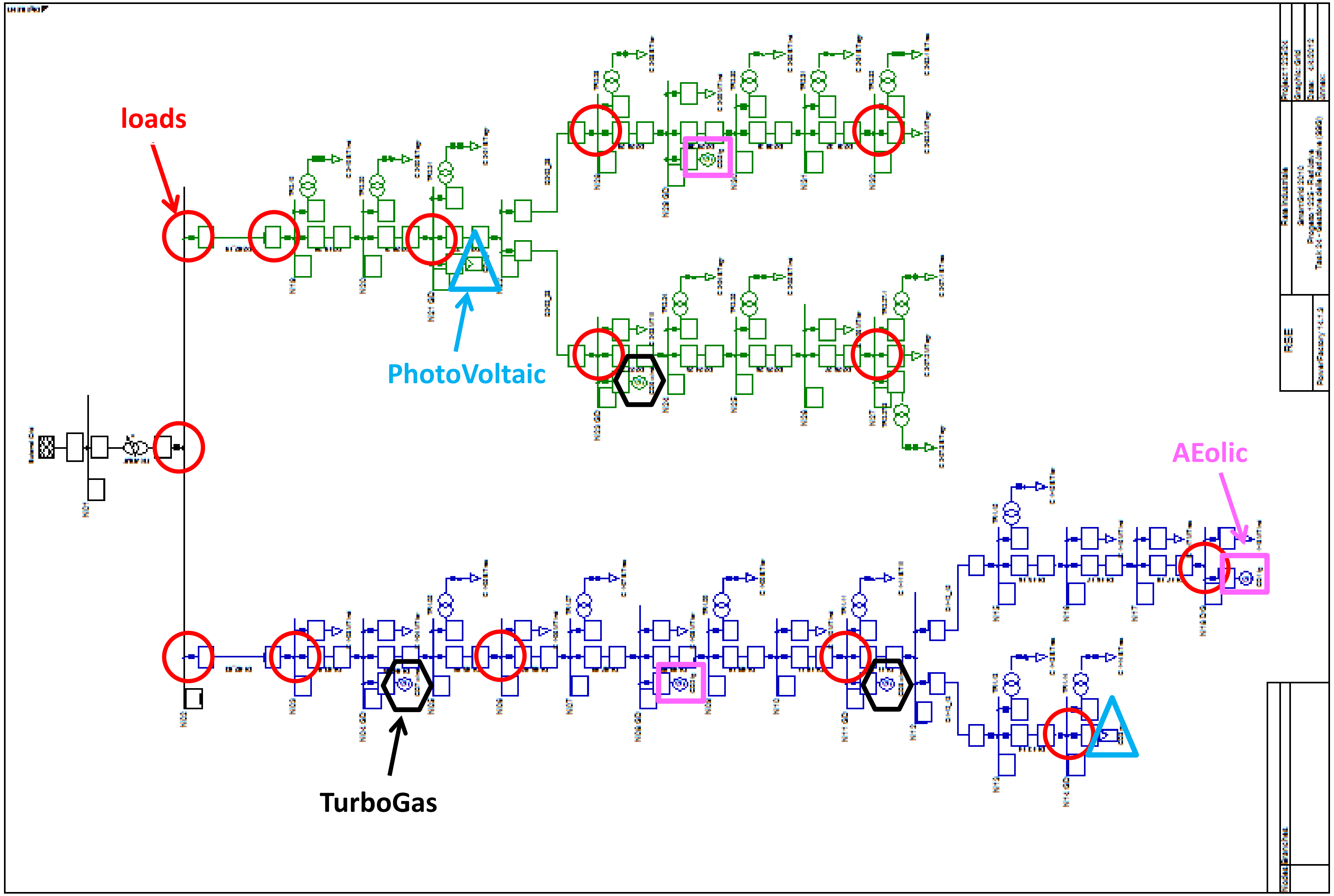
Also during the experiment, disturbances should be avoided (how?)

Impulse responses tend to excite the system too much (and the are always approximate impulses). It is preferred to use step responses (this is not a problem as we shall see)

Example: voltage control acting on the Distributed Generators power factors

Sophisticated model in DigSilent (environment for simulation of electrical systems)

Too complex to design a regulator, but useful to compute an impulse response model



Consider an asymptotically stable, SISO (just for simplicity) system

$$
\begin{aligned}
x(k+1) &= Ax(k) + Bu(k) \\
y(k) &= Cx(k)
\end{aligned}
\quad \xleftrightarrow{\;g_i = CA^{i-1}B\;} \quad
y(k) = \sum_{i=1}^{\infty} g_i u(k-i)
$$

In view of the stability assumption, $A^i \overset{i\to\infty}{\to} 0$, so that $g_i \overset{i\to\infty}{\to} 0$. In practice, after M time instants it is possible to assume

$$
g_{M+i} = 0, \quad i > 0
$$

Therefore, the model can be approximated by

$$
y(k) = \sum_{i=1}^{M} g_i u(k-i)
$$

$$y(k) = \sum_{i=1}^{M} g_i u(k-i)$$

At time k, the output prediction at time $k+i$, $i > 0$, is

$$y(k+i) = \sum_{j=1}^{i} g_j u(k+i-j) + \sum_{j=i+1}^{M} g_j u(k+i-j)$$

and, for $N < M$,

$$Y(k) = \mathcal{B}_g U(k) + \mathcal{B}_g^{old} U_{old}(k)$$

$$\mathcal{B}_g = \begin{bmatrix} g_1 & 0 & 0 & \cdots & 0 & 0 \\ g_2 & g_1 & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ g_{N-1} & g_{N-2} & g_{N-3} & \cdots & g_1 & 0 \\ g_N & g_{N-1} & g_{N-2} & \cdots & g_2 & g_1 \end{bmatrix}, \quad U_{old}(k) = \begin{bmatrix} u(k-1) \\ u(k-2) \\ \vdots \\ u(k-M+2) \\ u(k-M+1) \end{bmatrix}$$

$$\mathcal{B}_g^{old} = \begin{bmatrix} g_2 & g_3 & \cdots & \cdots & \cdots & g_{M-1} & g_M \\ g_3 & g_4 & \cdots & \cdots & \cdots & g_M & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ g_{N+1} & g_{N+2} & \cdots & g_M & 0 & 0 & 0 \end{bmatrix}$$

What is the problem with this formulation?

$$Y(k) = \mathcal{B}_g U(k) + \mathcal{B}_g^{old} U_{old}(k)$$

The output predictions do not depend on the output measurements up to time *k*. This means that the resulting control law will not depend on ***y(k-i), i=0,1,…*** so that **there is no feedback**

Let's consider *a very simple example* (*SISO* case, null reference signal, no constraints)

$$\begin{aligned} J &= \sum_{i=1}^{N} y^2(k+i) + u^2(k+i-1) \\ &= Y'(k)Y(k) + U'(k)U(k) \\ &= (\mathcal{B}_g U(k) + \mathcal{B}_g^{old} U_{old}(k))'(\mathcal{B}_g U(k) + \mathcal{B}_g^{old} U_{old}(k)) + U'(k)U(k) \end{aligned}$$

$$\frac{\partial J}{\partial U} = 0$$

$$\boxed{U(k) = -(\mathcal{B}_g' \mathcal{B}_g + I)^{-1} \mathcal{B}_g' \mathcal{B}_g^{old} U_{old}(k)}$$

no dependence on past outputs

How to get rid of this problem?

We assume that an unknown disturbance affects the system

$$y(k) = \sum_{i=1}^{M} g_i u(k-i) + d(k)$$

and the disturbance is constant

$$d(k+i) = d(k), \quad i > 0$$

Therefore

$$y(k+i) = \sum_{j=1}^{M} g_j u(k+i-j) + d(k+i) \longrightarrow y(k+i) = \sum_{j=1}^{M} g_j u(k+i-j) + \underbrace{y(k) - \sum_{j=1}^{M} g_j u(k-j)}_{d(k+i) = d(k)}$$

$$y(k+i) = \sum_{j=1}^{M} g_j u(k+i-j) + y(k) - \sum_{j=1}^{M} g_j u(k-j)$$

$$\downarrow$$

$$y(k+i) = \underbrace{\left(\sum_{j=1}^{i} g_j u(k+i-j) \right)}_{\text{depends on the future}} + \underbrace{\left(y(k) + \sum_{j=i+1}^{M} g_j u(k+i-j) - \sum_{j=1}^{M} g_j u(k-j) \right)}_{\text{depends on the past}}$$

Using this form in a standard MPC problems makes the control law to depend on the current value of the output

Exercise

Consider the first order system

$$x(k+1) = 0.5x(k) + u(k)$$
$$y(k) = x(k)$$

A. Show how to compute the first five impulse response coefficients g_i, $i=1,\dots,5$.

Problem data: $b=c=1$, $a=0.5$, so that $g_i=0.5^{i-1}$, $i=1,2,\dots$

B. Show how to compute at time *k* the predictions *y(k+1), y(k+2)* as functions of the past and future control variables

Predictions:

$$\begin{bmatrix} y(k+1) \\ y(k+2) \end{bmatrix} = \begin{bmatrix} g_1 & 0 \\ g_2 & g_1 \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \end{bmatrix} + \begin{bmatrix} g_2 & g_3 & g_4 & g_5 \\ g_3 & g_4 & g_5 & 0 \end{bmatrix} \begin{bmatrix} u(k-1) \\ u(k-2) \\ u(k-3) \\ u(k-4) \end{bmatrix}$$

C. Show how to modify the formulation of these predictions to make them to depend on the current output variable

with the dependence on *y(k)* one has

$$\begin{bmatrix} y(k+1) \\ y(k+2) \end{bmatrix} = \begin{bmatrix} g_1 & 0 \\ g_2 & g_1 \end{bmatrix} \begin{bmatrix} u(k) \\ u(k+1) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} y(k) + \begin{bmatrix} g_2 - g_1 & g_3 - g_2 & g_4 - g_3 & g_5 - g_4 & -g_5 \\ g_3 - g_1 & g_4 - g_2 & g_5 - g_3 & -g_4 & -g_5 \end{bmatrix} \begin{bmatrix} u(k-1) \\ u(k-2) \\ u(k-3) \\ u(k-4) \\ u(k-5) \end{bmatrix}$$

or, setting $I = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and with obvious definition of the other symbols

$$Y(k) = GU(k) + Iy(k) + G_{old} U_{old(k)}$$

D. Consider a cost function with prediction horizon *N=2*, with weight on the future error defined as the difference between the future output and a given set-point y^o), and formulate an MPC problem with input and output constraints and with slack variables to guarantee the feasibility of the problem at any time instant

Defining

$$Y^o = \begin{bmatrix} y^o \\ y^o \end{bmatrix}$$

The vector of the reference signal over the prediction horizon, the optimization problem becomes

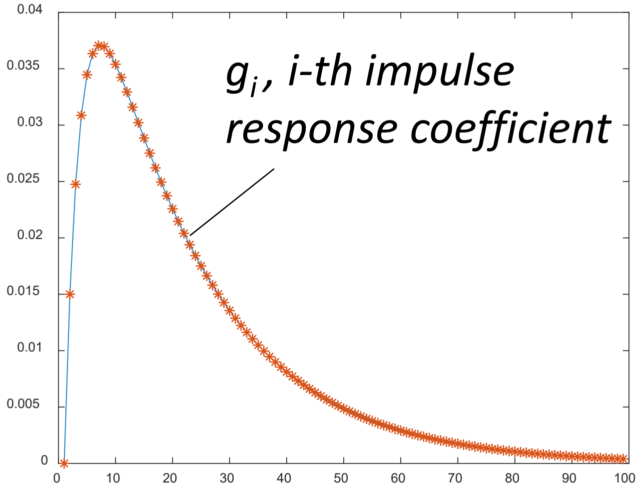
$$\min_{U(k),\varepsilon} J = (Y^o - Y(k))'Q(Y^o - Y(k)) + U'(k)RU(k) + \lambda\varepsilon(k)$$

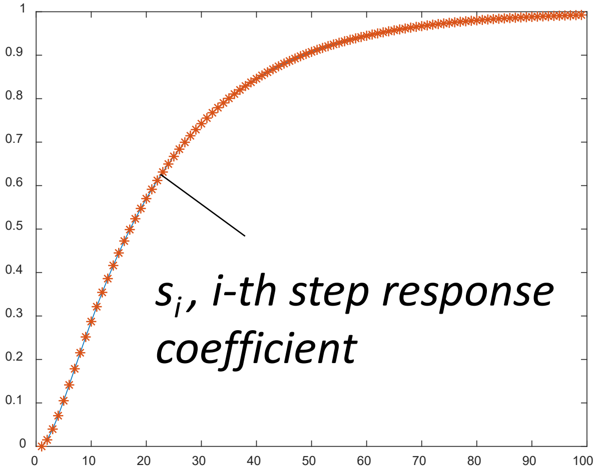
subject to the previous dynamics and

$$U_{min} \leq U(k) \leq U_{max}$$
$$Y_{min} - \varepsilon(k) \leq Y(k) \leq Y_{max} + \varepsilon(k)$$
$$\varepsilon(k) \geq 0$$

where ε are the slack variables.
Once the optimal *U(k)* has been computed, only *u(k)* is used and the overall procedure is repeated at the next time instant.

Relationships between impulse and step response coefficients





$$s_i = \sum_{j=1}^{i} g_j \quad \longleftrightarrow \quad g_i = s_i - s_{i-1}$$

Letting $s_0 = 0$

$$y(k) = \sum_{i=1}^{\infty} g_i u(k-i) = \sum_{i=1}^{\infty} (s_i - s_{i-1}) u(k-i) = \sum_{i=1}^{\infty} s_i \delta u(k-i)$$

For asymptotically stable SISO systems $g_{M+i} = 0 \quad \longleftrightarrow \quad s_{M+i} = s_M$

$$y(k) = \sum_{i=1}^{\infty} g_i u(k-i) = \sum_{i=1}^{\infty} (s_i - s_{i-1}) u(k-i) = \sum_{i=1}^{\infty} s_i \delta u(k-i)$$

$$y(k+i) = \sum_{j=1}^{\infty} s_j \delta u(k+i-j) + d(k+i) \qquad \longleftrightarrow \qquad d(k+i) = d(k) = y(k) - \sum_{j=1}^{\infty} s_j \delta u(k-j)$$

$$y(k+i) = \left(\sum_{j=1}^{i} s_j \delta u(k+i-j) \right) + \left(y(k) + \sum_{j=1}^{M-1} (s_{i+j} - s_j) \delta u(k-j) \right)$$

By properly formulating the MPC problem one computes the optimal sequence $\delta u(k+i),\ i = 0, ..., N-1$

Finally

$$u(k) = u(k-1) + \delta u^o(k)$$

Dynamic Matrix Control - ***DMC*** – hundreds (thousands) of applications

Notes

Empirical models are easy to obtain (it is a simplified form of model identification)

The obtained models are largely parametrized even for simple (I order) systems. The number of parameters also depends on the adopted sampling period

With DMC there is no need to compute steady state values of $\boldsymbol{x}$ and $\boldsymbol{u}$, since the cost function can be formulated in terms of future output errors and control variations

No need to use state observers

MPC with transfer function models

Consider SISO systems described by

$$G(z) = \frac{B(z)}{A(z)} = \frac{b_{n-1}z^{n-1} + b_{n-2}z^{n-2} + \ldots + b_0}{z^n + a_{n-1}z^{n-1} + a_{n-2}z^{n-2} + \ldots + a_0}$$

possibly obtained with model identification procedures

Solution 1: obtain a state space realization in terms of a control or observer canonical form

Solution 2: write the system in the state space realization

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

$$x(k+1) = \begin{bmatrix} y(k+1) \\ y(k) \\ \vdots \\ y(k-n+2) \\ u(k) \\ u(k-1) \\ \vdots \\ u(k-n+2) \end{bmatrix} \quad A = \begin{bmatrix} -a_{n-1} & -a_{n-2} & \cdots & -a_0 & b_{n-2} & b_{n-3} & \cdots & b_0 \\ 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} b_{n-1} \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} C = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$$

Non minimal form, but the state is measurable, no need of an observer

Extensions

MPC is highly flexible and can cope with many different situations

Before looking at the numerical algorithms and stability properties, let's have a look at some of these extensions

Extensions

Nonlinear MPC: MPC for nonlinear systems (see later)

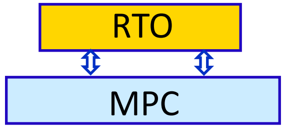
Continuous MPC: MPC for continuous time systems (see later)

Stabilizing MPC: MPC with stability guarantees (see later)

Robust MPC: the MPC problem is formulated also in presence of bounded disturbances with stability guarantees

Explicit MPC: MPC with explicit solution of the optimization problem (see later)

Economic MPC: the idea is to mix the static optimization problem used to compute the optimal operating conditions with the dynamic MPC regulator



Stochastic MPC: the system under control is affected by stochastic signals, the fulfillment of constraints must be guaranteed in stochastic terms

Hybrid MPC: the system under control hybrid, with continuous dynamics and asyncronous discrete events which modify its structure and configuration

… a very long list

MPC of nonlinear systems

Many solutions available, besides the standard one of formulating the problem for the linearized system

We consider here the formulation of the regulation problem at the origin (many extensions and weaker conditions could be used)

System

$$x(k+1) = f(x(k), u(k)) \qquad f(0,0) = 0$$

Cost function

$$\min_{U(k)} \quad J = \sum_{i=0}^{N-1} l(x(k+i), u(k+i)) \qquad l(x,u) \text{ positive definite at the origin}$$

$$x \in \mathcal{X}, \quad u \in \mathcal{U}$$

$$\mathcal{X}, \mathcal{U} \text{ compact sets containing the origin}$$

When the system is nonlinear, it is not possible to find explicit formulas representing the state evolution as a linear function of the future control variables, i.e. the state dynamics imposes nonlinear constraints. The state evolution can be computed through simulation given prescribed inputs.

When the performance index is nonlinear or non quadratic, and/or the state/input/output constraints are nonlinear, even in the case of linear systems, it is not possible to use linear or quadratic programming techniques for the solution of the optimization problem

In all these cases, the mathematical programming problem is much more difficult to solve (on-line!) and specific algorithms and SW environments have been developed

A short overview of possible numerical solutions of *Nonlinear* MPC

Problem statement

nonlinear system

$$x(k+1) = f(x(k), u(k))$$

quadratic cost function

$$\min_{u(k),\ldots,u(k+N-1),x(k+1),\ldots,x(k+N)} \quad J = \sum_{j=0}^{N-1} \frac{1}{2} \begin{bmatrix} x(k+j) - x^o(k+j) \\ u(k+j) - u^o(k+j) \end{bmatrix}' \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} x(k+j) - x^o(k+j) \\ u(k+j) - u^o(k+j) \end{bmatrix}$$

linear or nonlinear constraints

$$h(x(k+j), u(k+j)) \leq 0 \quad , \quad j = 0, \ldots, N-1$$

Solution 1 – linearization along the reference trajectories

Assume to know in advance reference trajectories x^o and u^o

Set

$$\Delta x(k+j) = x(k+j) - x^o(k+j) \quad , \quad \Delta u(k+j) = u(k+j) - u^o(k+j) \quad , \quad j = 0, ..., N-1$$

and

$$A(k+j) = \left.\frac{\partial f}{\partial x}\right|_{x^o(k+j), u^o(k+j)}, B(k+j) = \left.\frac{\partial f}{\partial u}\right|_{x^o(k+j), u^o(k+j)}$$

By means of linearization,

$$\Delta x(k+j+1) = A(k+j)\Delta x(k+j) + B(k+j)\Delta u(k+j) + r(k+j)$$

where

$$r(k+j) = f(x^o(k+j), u^o(k+j)) - x^o(k+j+1)$$

which can be zero if the reference trajectory satisfies the system dynamics

Linearize also the constraints

$$h(x(k),u(k)) = h(x^o(k),u^o(k)) + C(k)\Delta x(k) + D(k)\Delta u(k)$$

$$C(k) = \left.\frac{\partial h}{\partial x}\right|_{x^o(k),u^o(k)}, D(k) = \left.\frac{\partial h}{\partial u}\right|_{x^o(k),u^o(k)}$$

and finally formulate the QP problem

$$\min_{\Delta u(k),\dots,\Delta u(k+N-1),\Delta x(k+1),\dots,\Delta x(k+N)} \quad J = \sum_{j=0}^{N-1} \frac{1}{2} \begin{bmatrix} \Delta x(k+j) \\ \Delta u(k+j) \end{bmatrix}' \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} \Delta x(k+j) \\ \Delta u(k+j) \end{bmatrix}$$

$$\Delta x(k+i+1) = A(k+i)\Delta x(k+i) + B(k+i)\Delta u(k+i) + r(k+i)$$

$$h(x^o(k),u^o(k)) + C(k)\Delta x(k) + D(k)\Delta u(k) \leq 0$$

Solution 2 – linearization along the predicted trajectories

Assume that at time ***k-1*** the *optimal control sequence* has been computed

$$\bar{u}^o(k-1 : k+N-2|k-1) = \left[\; \underbrace{u^o(k-1|k-1)}_{\text{applied}} \quad u^o(k|k-1) \quad \dots \quad u^o(k+N-2|k-1) \; \right]$$

Consider the *future control sequence* (better choices could be done)

$$\bar{u}^o(k : k+N-1|k-1)) = \left[\; u^o(k|k-1) \quad \dots. \quad u^o(k+N-2|k-1) \quad u^o(k+N-2|k-1) \; \right]$$

It is possible to compute the predicted state trajectory

$$\hat{x}(k+i+1|k) = f(\hat{x}(k+i|k), u^o(k+i|k-1)) \quad , \quad i = 1, ..., N-1$$

This predicted trajectory is used for linearization

Define

$$\tilde{\delta}\hat{x}(k+i|k) = x(k+i) - \hat{x}(k+i|k) \quad , \quad i = 0, ..., N-1$$

$$\tilde{\delta}u(k+i|k-1) = u(k+i) - u^o(k+i|k-1) \quad , \quad i = 0, ..., N-1$$

Then

$$x(k+i+1) = f(\hat{x}(k+i|k), u^o(k+i|k-1)) + A(k+i)\tilde{\delta}\hat{x}(k+i|k) + B(k+i)\tilde{\delta}u(k+i|k-1)$$

$$A(k+i) = \left.\frac{\partial f}{\partial x}\right|_{\hat{x}(k+i|k), u^o(k+i|k-1)}, B(k+i) = \left.\frac{\partial f}{\partial u}\right|_{\hat{x}(k+i|k), u^o(k+i|k-1)}$$

Letting

$$H(k+i) = f(\hat{x}(k+i|k), u^o(k+i|k-1)) - (A_{k+i}\hat{x}(k+i|k) + B_{k+i}u(k+i|k-1))$$

One finally obtains

$$x(k+i+1) = A(k+i)x(k+i) + B(k+i)u(k+i) + H(k+i)$$

computable at k
linearization error

Final QP problem

$$\min_{u(k),\dots,u(k+N-1),x(k+1),\dots,x(k+N)} \quad J=\sum_{j=0}^{N-1}\frac{1}{2}\begin{bmatrix} x(k+j)-x^o(k+j) \\ u(k+j)-u^o(k+j)\end{bmatrix}'\begin{bmatrix} Q & 0 \\ 0 & R\end{bmatrix}\begin{bmatrix} x(k+j)-x^o(k+j) \\ u(k+j)-u^o(k+j)\end{bmatrix}$$

$$x(k+i+1)=A(k+i)x(k+i)+B(k+i)u(k+i)+H(k+i) \qquad (1)$$

$$h(\hat{x}(k+i),u^o(k+i|k-1))+C(k+i)(x(k+i)-\hat{x}(k+i))+D(k)(u(k+i)-u^o(k+i|k-1))\leq 0$$

$$C(k+i)=\left.\frac{\partial h}{\partial x}\right|_{\hat{x}(k+i),u^o(k+i|k-1)},\; D(k+i)=\left.\frac{\partial h}{\partial u}\right|_{\hat{x}(k+i),u^o(k+i|k-1)}$$

Solution 3 – nonlinear optimization … preliminaries

Consider the nonlinear problem

$$\min_x f(x)$$
$$g(x) = 0$$
$$h(x) \geq 0$$

$$f, g, h \in C^2$$

Define the **Lagrangian** $L(x, \lambda, \mu) = f(x) - \lambda' g(x) - \mu' h(x)$

KKT necessary conditions

$$\frac{\partial L}{\partial x}\Big|_{x^*,\lambda^*,\mu^*} = \frac{\partial f}{\partial x}\Big|_{x^*} - \frac{\partial g}{\partial x}\Big|_{x^*}\lambda^* - \frac{\partial h}{\partial x}\Big|_{x^*}\mu^* = 0$$
$$g(x^*) = 0$$
$$h(x^*) \geq 0$$
$$\mu^* \geq 0$$
$$\mu_i^* h_i(x^*) = 0 \quad i = 1, ..., q (n. of constraints)$$

Problem: it can be not easy to solve this set of equations with respect to (x^*, λ^*, μ^*)

Sequential Quadratic Programming (SQP) approach

Given a solution x_i iteratively compute

$$x_{i+1} = x_i + \Delta x_i$$

solving the quadratic optimization problem

$$\min_{\Delta x_i} f(x_i) + \frac{\partial f(x)'}{\partial x}\Big|_{x_i} \Delta x_i + \frac{1}{2}\Delta x_i' \frac{\partial^2 L(x_i, \lambda_i, \mu_i)}{\partial x^2}\Big|_{x_i} \Delta x_i$$

$$h(x_i) + \frac{\partial h(x)'}{\partial x}\Big|_{x_i} \Delta x_i \geq 0$$

$$g(x_i) + \frac{\partial g(x)'}{\partial x}\Big|_{x_i} \Delta x_i = 0$$

until convergence is reached. **Problem**: $\frac{\partial^2 L(x_i, \lambda_i, \mu_i)}{\partial x^2}$ is difficult to compute, many approximations have been proposed guaranteeing convergence

SQP: a basic approach for MPC

At iteration i, given $x_i(k)=x(k), x_i(k+1), ..., x_i(k+N-1), u_i(k), ..., u_i(k+N-1)$
solve the QP problem

$$\min_{\Delta u_i(k),\dots,\Delta x_i(k+N)} \quad J=\sum_{j=0}^{N-1}\frac{1}{2}\begin{bmatrix}\Delta x_i(k+j)\\ \Delta u_i(k+j)\end{bmatrix}'\begin{bmatrix}Q & 0\\ 0 & R\end{bmatrix}\begin{bmatrix}\Delta x_i(k+j)\\ \Delta u_i(k+j)\end{bmatrix}+\begin{bmatrix}x_i(k+j)-x^o(k+j)\\ u_i(k+j)-u^o(k+j)\end{bmatrix}'\begin{bmatrix}Q & 0\\ 0 & R\end{bmatrix}\begin{bmatrix}\Delta x_i(k+j)\\ \Delta u_i(k+j)\end{bmatrix}$$

$$\Delta x_i(k+j+1)=A_i(k+j)\Delta x_i(k+j)+B_i(k+j)\Delta u_i(k+j)+r(k+j)$$

$$r_i(k+j)=f(x_i(k+j),u_i(k+j))-x_i(k+j+1)$$

$$h(x_i(k+j),u_i(k+j))+C_i(k+j)\Delta x_i(k+j)+D_i(k+j)\Delta u_i(k+j)\leq 0$$

$$A_i(k+j)=\left.\frac{\partial f}{\partial x}\right|_{x_i(k+j),u_i(k+j)}, B_i(k+j)=\left.\frac{\partial f}{\partial u}\right|_{x_i(k+j),u_i(k+j)}, C_i(k+j)=\left.\frac{\partial h}{\partial x}\right|_{x_i(k+j),u_i(k+j)}, D_i(k+j)=\left.\frac{\partial h}{\partial u}\right|_{x_i(k+j),u_i(k+j)}$$

Then set

$$\begin{bmatrix}x_i(k)\\ u_i(k)\end{bmatrix}+\alpha\begin{bmatrix}\Delta x_i(k)\\ \Delta u_i(k)\end{bmatrix}\rightarrow\begin{bmatrix}x_{i+1}(k)\\ u_{i+1}(k)\end{bmatrix} \quad , \quad \alpha\in(0,1]$$

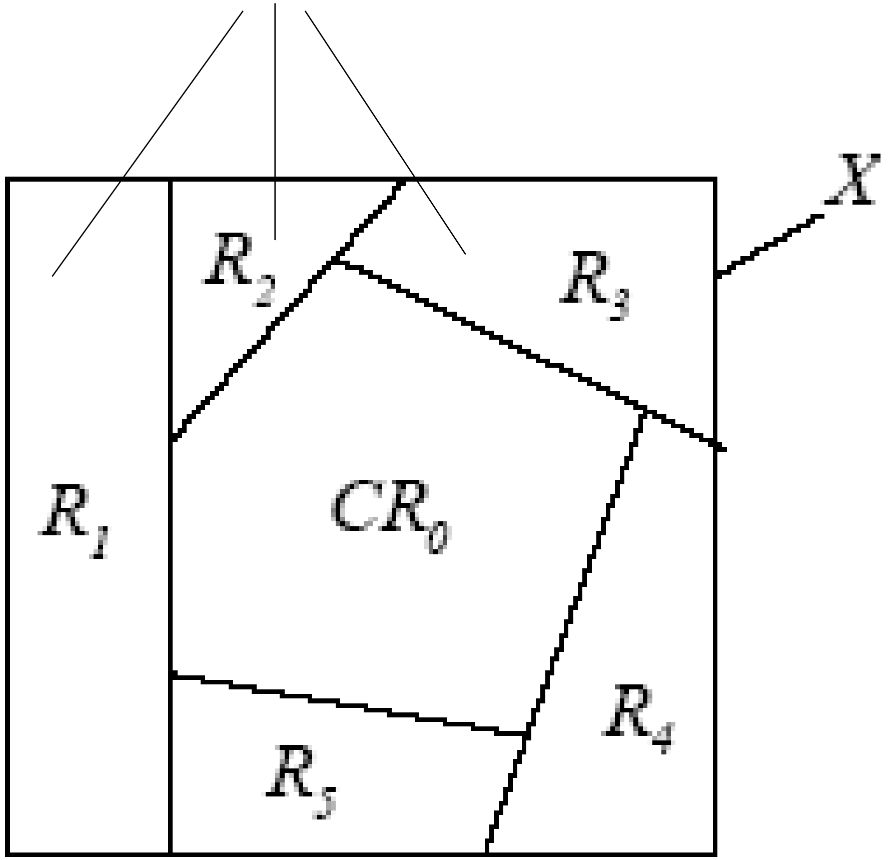
until a stopping condition is verified

Explicit MPC

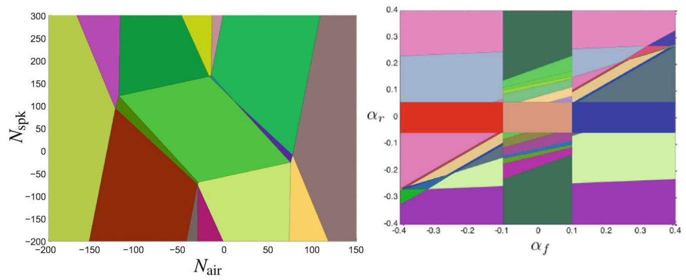
For linear constrained systems it has been proven that the solution of the infinite horizon LQ problem takes the form

$$u = K_{CR_i}x + \gamma_{CR_i}$$

in specific regions of the state space



Possible approach



1. Compute off-line the regions and the corresponding control laws

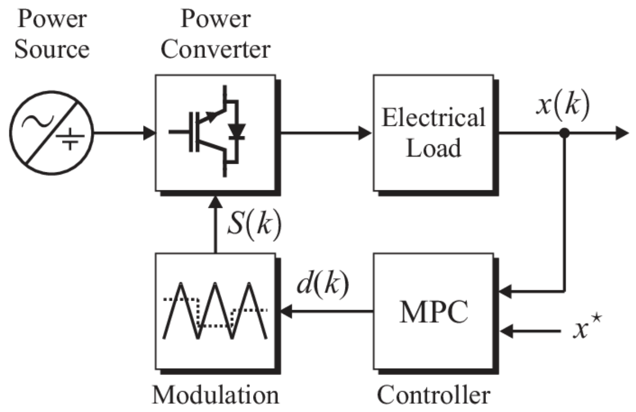
2. Compute on-line the region where is the state and apply the corresponding control law

Pros & cons

Step 2 can be computationally demanding due to a very large number of regions, even more demanding than computing the implicit control law with a standard MPC formulation

Interesting approach for fast systems of small size (order 2-3)

Already used in many industrial fields (automotive, power electronics,…)

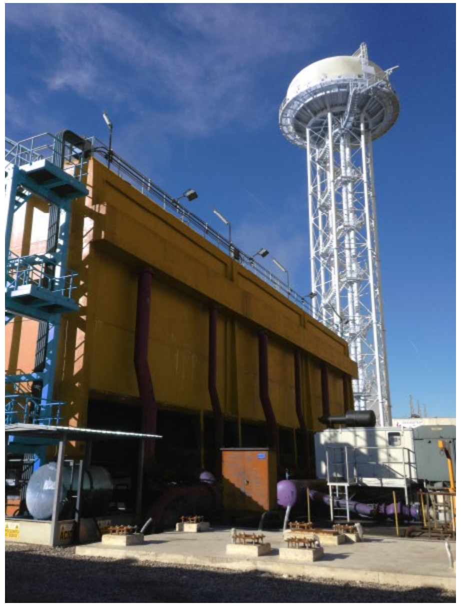


Systems with on-off actuators

Imagine to have a system with *n* actuators which can be switched on/off, for example a set of on/off pumps

In this case, the MPC optimization problem will contain the constraint $u(k) \in \{0, 1, 2, ..., n\}$

The resulting problem is more difficult to solve (typically it is a Mixed Integer problem), but still affordable if the adopted sampling period is not too small



Hybrid MPC

MPC can handle logical relations and constraints among the process variables

Procedure

$$and(\wedge), \qquad or(\vee), \qquad not(\neg), \qquad imply(\rightarrow),\ iff(\leftrightarrow),\ exclusive\ or(\oplus)$$

Truth table (X_i are statements which can be true or false)

X_1	X_2	$\neg X_1$	$X_1 \vee X_2$	$X_1 \wedge X_2$	$X_1 \rightarrow X_2$	$X_1 \leftrightarrow X_2$	$X_1 \oplus X_2$
F	F	T	F	F	T	T	F
F	T	T	T	F	T	F	T
T	F	F	T	F	F	F	T
T	T	F	T	T	T	T	F

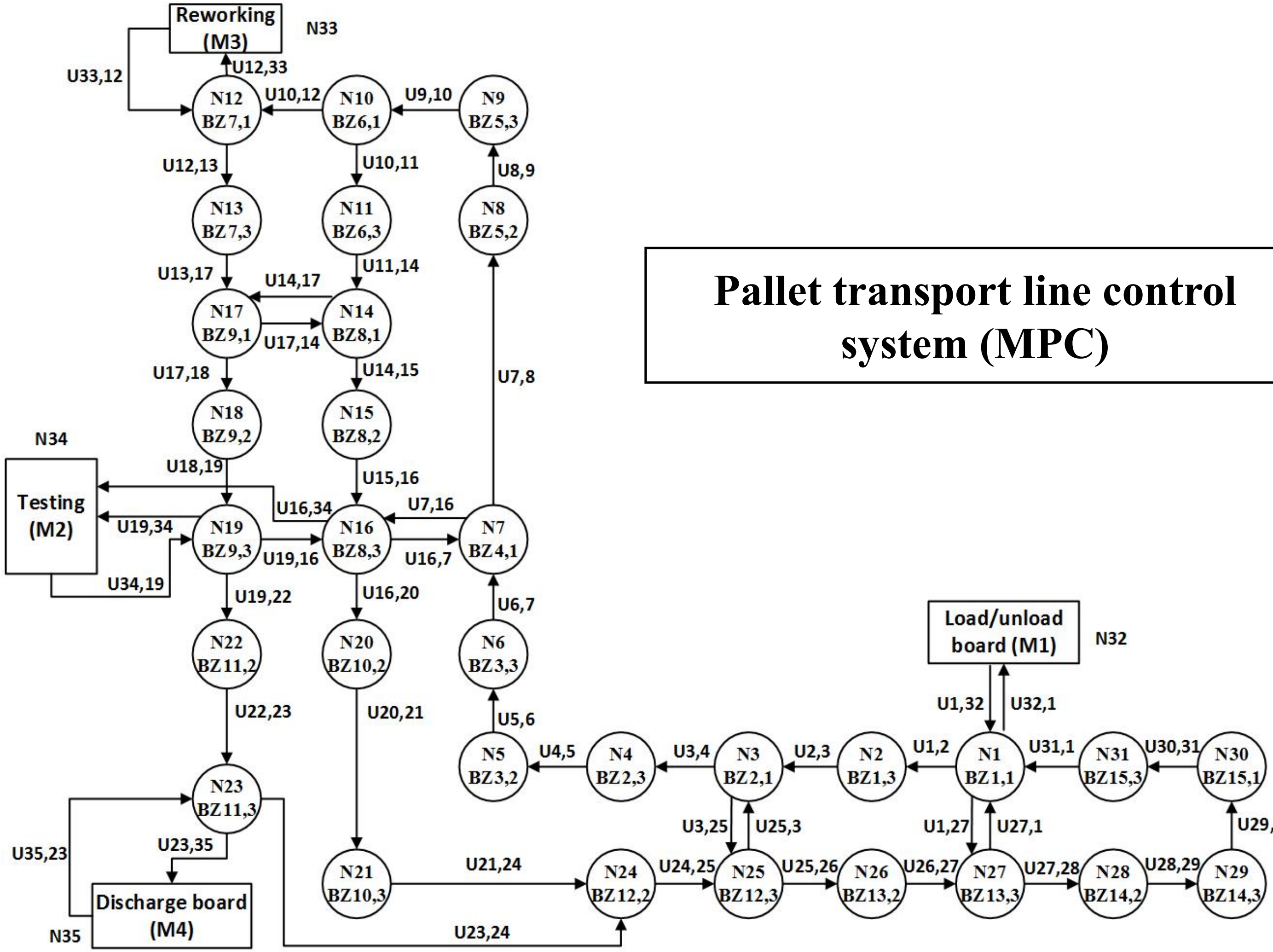
to each statement X_i is associated a boolean variable $\delta_i \in \{0,1\}$ such that $\delta_i = 1\ iff\ \ X_i = True$

With propositional calculus, logical relations can be transformed into linear inequalities

$$
\begin{array}{ccc}
X_1 \vee X_2 & \Leftrightarrow & \delta_1 + \delta_2 \geq 1 \\
X_1 \wedge X_2 & \Leftrightarrow & \delta_1 = 1, \delta_2 = 1 \\
\neg X_1 & \Leftrightarrow & \delta_1 = 0 \\
X_1 \rightarrow X_2 & \Leftrightarrow & \delta_1 - \delta_2 \leq 0 \\
X_1 \leftrightarrow X_2 & \Leftrightarrow & \delta_1 - \delta_2 = 0 \\
X_1 \oplus X_2 & \Leftrightarrow & \delta_1 + \delta_2 = 1
\end{array}
$$

These relations can be used to describe logical constraints as inequalities among integer (boolean) variables to be included in the optimization problem. This leads to the definition of Mixed Logical Dynamical (MLD) systems

Example: MPC of a transportation line



Pallet transport line control system (MPC)

Stability of MPC

system

$$x(k+1) = f(x(k), u(k)) \qquad f \in C^1 \text{ and } f(0,0) = 0$$

constraints

$$x \in X \quad , \quad u \in U$$

X, U compact containing the origin

How to design MPC to guarantee the stability of the origin?

MPC is based on the solution of a suitable optimization problem.

When studying its properties, two main issues have to be considered:

Recursive feasibility
if at a given time *k* the optimization problem is feasible, it will be feasible also at k+1.
This can be guaranteed using slack variables, which however modify the cost function. The goal is be to use other approaches for theoretical analysis.

Stability (equilibrium)
Stability of the equilibrium must be guaranteed by properly using the (modified) cost function as a Lyapunov function of the closed-loop system with MPC

Preliminaries – an extension of the Lyapunov theorem

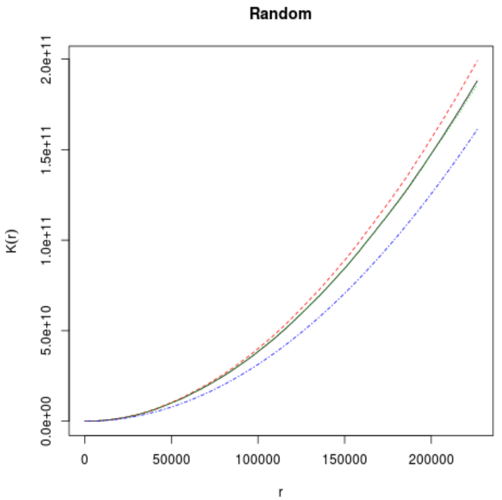
Let $X^o \subseteq R^n$ be a positively invariant set for the system

$$x(k+1) = f(x(k))$$

Positively invariant set:
if $x(k) \in X^o$, also $x(k+1) = f(x(k)) \in X^o$

containing a neighborhood $\mathcal{N}$ of the equilibrium $\bar{x} = 0$
Let w, ψ, r be class *K* functions and assume that there exists a nonnegative scalar function $V : X^o \to R_+, V(0) = 0$ such that

$$V(x) \geq w(\|x\|), \quad \forall x \in X^o$$
$$V(x) \leq \psi(\|x\|), \quad \forall x \in \mathcal{N}$$
$$\Delta V(x) \leq -r(\|x\|), \quad \forall x \in X^o$$



$\Downarrow$

then the origin is an asymptotically stable equilibrium in X^o. Moreover, if

$$w(\|x\|) := a\|x\|^\sigma, \psi(\|x\|) := b\|x\|^\sigma, r(\|x\|) := c\|x\|^\sigma$$

for some $a, b, c, \sigma > 0$ and $\mathcal{N} = X^o$ then the origin is exponentially stable in X^o

$\varphi : R_+ \to R$ is a ***K* function** if it is continuous, strictly increasing with $\varphi(0) = 0$

The auxiliary control law

Assume to know a stabilizing auxiliary control law $u = \kappa_a(x)$

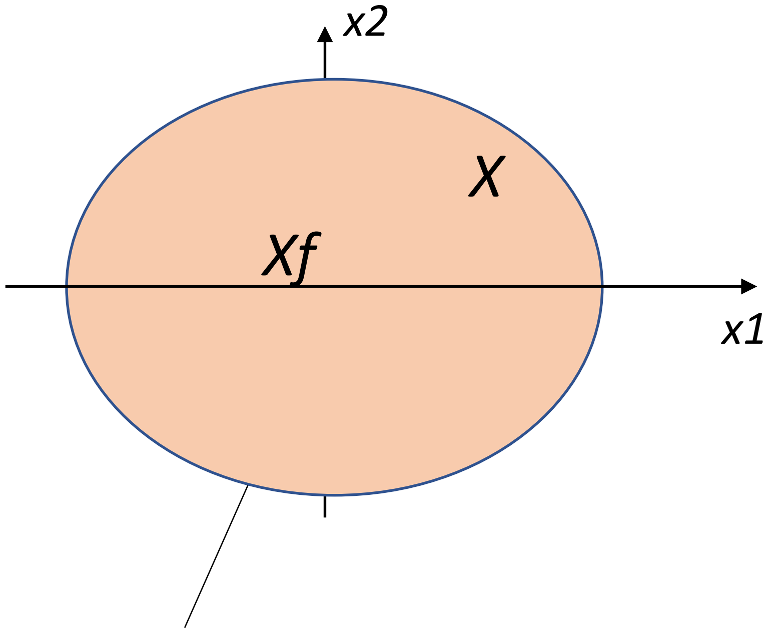
and a positively invariant set $X_f \subset X$ containing the origin such that, for the closed-loop system

$$x(k+1) = f(x(k), \kappa_a(x(k)))$$

and for any $x(\bar{k}) \in X_f$

one has

$$\begin{aligned} &x(k) \in X_f \quad , \quad k \geq \bar{k} \\ &u(k) = \kappa_a(x(k)) \in U \quad , \quad k \geq \bar{k} \end{aligned}$$



If you are here and apply the auxiliary control law, you remain here and the control constraints are satisfied

MPC problem: at any time *k* find the sequence

$$u(k), u(k+1), \ldots, u(k+N-1)$$

minimizing the cost function (*Q>0 (for simplicity), R>0*)

$$J(x(k), u(\cdot), N) = \sum_{i=0}^{N-1} \left(\|x(k+i)\|_Q^2 + \|u(k+i)\|_R^2 \right) + V_f(x(k+N))$$

subject to

$$x(k+i) \in X \quad , \quad u(k+i) \in U$$

$$x(k+N) \in X_f$$

The RH solution implicitly defines the MPC time-invariant control law

$$u = \kappa_{RH}(x)$$

Theorem

Let $X^{RH}(N)$ be the set of states where a solution of the optimization problem exists.
If, for any $x \in X_f$ the condition

$$V_f\left(f\left(x(k),\kappa_a(x(k))\right)\right) - V_f(x(k)) + \left(\|x(k)\|_Q^2 + \|\kappa_a(x(k))\|_R^2\right) \leq 0$$

is fulfilled and

$$V_f(x) \leq \alpha_f\left(\|x\|\right)$$

where $\alpha_f\left(\|x\|\right)$ is a class *K* function, then the origin of the closed-loop system with the MPC-RH control law is an asymptotically stable equilibrium point with region of attraction $X^{RH}(N)$

Moreover, if $\alpha_f\left(\|x\|\right) = b\,\|x\|^2$ and $X_f = X^{RH}(N)$

then the origin is exponentially stable in $X^{RH}(N)$

Proof
Recursive feasibility

Let $x(k) \in X^{RH}(N)$ and

$$U^o(k,N) = \left[\; u_k^o(k) \quad u_k^o(k+1) \quad \cdots \quad u_k^o(k+N-2) \quad u_k^o(k+N-1) \;\right]$$

be the optimal solution at *k* with prediction horizon *N*. Then, at time *k+1*

$$\tilde{U}(k+1,N) = [u_k^o(k+1) \cdots \; u_k^o(k+N-2) u_k^o(k+N-1) \;\; \kappa_a(x(k+N))]$$

is a feasible solution, so that $x(k+1) \in X^{RH}(N)$

Moreover,

$$V(x,N) := J(x, \kappa_{RH}(x), N) \geq \|x\|_Q^2$$

so that the condition $V(x,N) \geq w(\|x\|)$ is verified in $X^{RH}(N)$

At time *k*, the sequence

$$\tilde{U}(k, N+1) = [U^o(k, N), \kappa_a(x(k+N))]$$

is feasible for the MPC problem ***with horizon N+1*** and

$$\begin{aligned} J(x, \tilde{U}(k, N+1), N+1) &= V(x, N) - V_f(x(k+N)) + V_f(x(k+N+1)) \\ &+ \|x(k+N)\|_Q^2 + \|\kappa_a(x(k+N))\|_R^2 \\ &\leq V(x, N) \end{aligned}$$

so that we have the monotonicity property (with respect to *N)*

$$V(x, N+1) \leq V(x, N), \ \forall x \in X^{RH}(N)$$

with $V(x, 0) = V_f(x), \ \forall x \in X_f$

Then $V(x, N+1) \leq V(x, N) \leq \ldots \leq V_f(x) \leq \alpha_f(\|x(k)\|) \quad \forall x \in X_f$

and the condition $V(x, N) \leq \psi(\|x\|), \forall x \in X_f$ is satisfied.

Finally

$$\begin{aligned} V(x,N) &= \|x\|_Q^2 + \|\kappa_{RH}(x)\|_R^2 + J(f(x,\kappa_{RH}(x)), u^o(k+1,N-1), N-1) \\ &= \|x\|_Q^2 + \|\kappa_{RH}(x)\|_R^2 + V(f(x,\kappa_{RH}(x)), N-1) \\ &\geq \|x\|_Q^2 + \|\kappa_{RH}(x)\|_R^2 + V(f(x,\kappa_{RH}(x)), N) \\ &\geq \|x\|_Q^2 + V(f(x,\kappa_{RH}(x)), N), \quad \forall x \in X^{RH}(N) \end{aligned}$$

and also the condition $\Delta V(x) \leq -r(\|x\|), \forall x \in X^{RH}(N)$ is satisfied.

In conclusion, *V(x,N)* is a Lyapunov function.

Moreover, if $\alpha_f(\|x\|) = b\|x\|^2, X_f = X^{RH}(N)$ the origin is exponentially stable

Remark 1

The main point is to prove that the cost function is decreasing. For this, it is not necessary to find the optimum, but just a sequence

$$\bar{U}(k) = \left[\begin{array}{ccccc} \bar{u}_k(k) & \bar{u}_k(k+1) & \cdots & \bar{u}_k(k+N-2) & \bar{u}_k(k+N-1) \end{array} \right]$$

such that

$$\bar{J}(x(k), \bar{U}(k), k) < \tilde{J}(x(k), \tilde{U}(k, N), k)$$

Remark 2

It is possible to conclude that

$$X^{RH}(N+1) \supseteq X^{RH}(N)$$

In fact, with longer horizons one has more degrees of freedom.

Remark 3

$$X^{RH}(N) \supseteq X_f$$

In fact, the auxiliary control law $u = \kappa_a(x)$ can be used by the optimization algorithm

Remark 4

There exists a value $\bar{N}$ such that $X^{RH}(\bar{N}) \supseteq \bar{X}_f$ where $\bar{X}_f$ is the maximum (unknown) positively invariant set associated to the auxiliary control law.

But, how to select the terminal cost and the terminal set?

Algorithm 1: zero terminal constraint

This is the first algorithm proposed, defined by

$$\kappa_a(x) = 0 \qquad X_f = \{0\} \qquad V_f = 0 \qquad \longrightarrow \qquad x(k+N) = 0$$

In fact, since *f(0,0)=0*, if at time *k* the optimal sequence is

$$U^o(k) = \left[\begin{array}{cccccc} u_k^o(k) & u_k^o(k+1) & \cdots & u_k^o(k+N-2) & u_k^o(k+N-1) \end{array}\right]$$

leading to $x^o(k+N) = 0$, at time *k+1* the sequence

$$U(k+1) = \left[\begin{array}{ccccc} u_k^o(k+1) & u_k^o(k+2) & \cdots & u_k^o(k+N-1) & 0 \end{array}\right]$$

is such that

$$x^o(k+N+1) = x^o(k+N) = 0$$

and the condition

$$V_f\left(f\left(x(k), \kappa_a(x(k))\right)\right) - V_f(x(k)) + \left(\|x(k)\|_Q^2 + \|\kappa_a(x(k))\|_R^2\right) \leq 0$$

is satisfied (all the terms are null).

Algorithm 2: quasi infinite horizon (linear systems, for simplicity)

Consider the linear system

$$x(k+1) = Ax(k) + Bu(k)$$

and the LQ control law (computed with the same *Q, R* matrices of the MPC cost function)

$$u(k) = -K_{LQ}x(k)$$

auxiliary control law

Define the matrix *P* solution of

$$\left(A - BK_{LQ}\right)' P \left(A - BK_{LQ}\right) - P = -\left(Q + K_{LQ}' R K_{LQ}\right)$$

and the terminal set

$$X_f = \left\{x \middle| \, x'Px \le \alpha\right\} \subset X$$

where α is a sufficiently small value.

Consider also the terminal weight

$$V_f(x) = x'Px$$

These choices fulfill the stability condition

$$V_f\left(f\left(x(k), \kappa_a(x(k))\right)\right) - V_f(x(k)) + \left(\|x(k)\|_Q^2 + \|\kappa_a(x(k))\|_R^2\right) \leq 0$$

In fact

$$\begin{aligned}
\Gamma(x(k)) :=& \quad V_f\left(f\left(x(k), -K_{LQ}x(k)\right)\right) - V_f\left(x(k)\right) + \left(\|x(k)\|_Q^2 + \|\kappa_a(x(k))\|_R^2\right) \\
=& \quad x'(k)\left\{\left(A - BK_{LQ}\right)' P \left(A - BK_{LQ}\right) - P + \left(Q + K_{LQ}' R K_{LQ}\right)\right\} x(k) \\
=& \ 0
\end{aligned}$$

Moreover, X_f is positively invariant for the auxiliary control law, since its boundary coincides with a level line of the Lyapunov function associated to the closed-loop system.

Finally, with continuity arguments it can be concluded that in the neighborhood of the origin (i,.e. for a sufficiently small α) one has

$$u = -K_{LQ}x \in U$$

Note that the terminal cost can be interpreted as the “cost to go” of a classical LQ-IH approach.

Quasi infinite horizon for nonlinear systems

First assume that the system is linearizable at the origin

$$\begin{aligned} f(x,u) &= \left.\frac{\partial f}{\partial x}\right|_{x=u=0} \delta x + \left.\frac{\partial f}{\partial u}\right|_{x=u=0} \delta u + \phi(\delta x, \delta u) \\ &= A\delta x + B\delta u + \phi(\delta x, \delta u) \end{aligned}$$

where

$$\lim_{\|(\delta x,\delta u)\|\to 0} \sup \frac{\|\phi(\delta x,\delta u)\|}{\|(\delta x,\delta u)\|} = 0$$

For the linearized system compute with the same *Q, R* matrices the LQ control law

$$\delta u(k) = -K_{LQ}\delta x(k)$$

which will be used as the auxiliary control law.

For the corresponding nonlinear controlled system one has

$$f(x,u) = (A - BK_{LQ})\delta x + \phi(\delta x, -K_{LQ}\delta x)$$

and

$$\lim_{\|\delta x\| \to 0} \sup \frac{\left\|\phi(\delta x, -K_{LQ}\delta x)\right\|}{\|(\delta x, \delta u)\|} = 0$$

Now solve the Lyapunov equation

$$\left(A - BK_{LQ}\right)' P \left(A - BK_{LQ}\right) - P = -\beta\left(Q + K_{LQ}' R K_{LQ}\right) \qquad \beta > 1$$

and consider again the terminal cost

$$V_f(x) = x' P x$$

In the neighborhood of the origin,

$$\begin{aligned}\Gamma(x(k)) &:= V_f\left(f\left(x(k), -K_{LQ}x(k)\right)\right) - V_f\left(x(k)\right) + \left(\|x(k)\|_Q^2 + \|\kappa_a(x(k))\|_R^2\right) \\ &= f\left(x(k), -K_{LQ}x(k)\right)' P f\left(x(k), -K_{LQ}x(k)\right) \\ &\quad - x'(k)Px(k) + x'(k)Qx(k) + x'(k)K_{LQ}'RK_{LQ}x(k)\end{aligned}$$

$$\bar{\phi}(x) = \phi(x, -K_{LQ}x)$$

$$\begin{aligned}\Gamma(x(k)) = \; & x'(k)\left(A - BK_{LQ}\right)' P\left(A - BK_{LQ}\right)x(k) + 2x'(k)P\bar{\phi}(x(k)) + \\ & -x'(k)Px(k) + \bar{\phi}'(x)P\bar{\phi}(x) + x'(k)\left(Q + K_{LQ}'RK_{LQ}\right)x(k) \\ = \; & x'(k)\left\{\left(A - BK_{LQ}\right)' P\left(A - BK_{LQ}\right) - P + \left(Q + K_{LQ}'RK_{LQ}\right)\right\}x(k) + \\ & +2x'(k)P\bar{\phi}(x(k)) + \bar{\phi}'(x)P\bar{\phi}(x) \\ = \; & x'(k)\left(1 - \beta\right)\left(Q + K_{LQ}'RK_{LQ}\right)x(k) + 2x'(k)P\bar{\phi}(x(k)) + \bar{\phi}'(x)P\bar{\phi}(x)\end{aligned}$$

Letting $$L_{\bar{\phi}} = \sup \frac{\left\|\bar{\phi}(x)\right\|}{\|x\|}$$

one has $$2x'(k)P\bar{\phi}(x(k)) \leq 2\|P\| L_{\bar{\phi}} \|x(k)\|^2$$

and $$\bar{\phi}'(x)P\bar{\phi}(x) \leq \|P\| L_{\bar{\phi}}^2 \|x(k)\|^2$$

Since $L_{\bar{\phi}} \to 0 for \|x(k)\| \to 0$ then $\Gamma(x(k)) \leq 0$ in a sufficiently small neighborhood of the origin, so that the decreasing condition is satisfied.

Finally note that $X_f = \left\{x|\, x'Px \leq \alpha\right\} \subset X$ is positively invariant for the auxiliary LQ control law, as it coincides with a level line of the Lyapunov function associated to the linearized system. Moreover, in a neighborhood of the origin $u = -K_{LQ}x \in U$

Tracking of constant references

If the goal is to stabilize the system at the equilibrium (x_s, u_s),
With $x_s \in X_f(x_s)$ the performance index can be chosen as

$$J = \sum_{i=0}^{N-1} \left\| x(k+i) - x_s \right\|_Q^2 + \left\| u(k+i) - u_s \right\|_R^2 + V_f(x(k+N) - x_s)$$

the condition to be fulfilled in X_f is

$$V_f\big(f(x, \kappa_a(x))\big) - V_f(x) + \left\| x - x_s \right\|_Q^2 + \left\| \kappa_a(x) - u_s \right\|_R^2 \le 0$$

zero terminal constraint	$V_f = 0$ $X_f = \{x_s\}$ $\kappa_a(x_s) = u_s$

quasi infinite horizon	$V_f = (x - x_s)' P (x - x_s)$ $\kappa_a(x) = u_s + K(x - x_s)$ $x(k+N) \in X_f(x_s)$

The end