MPC,

- MODEL IDENTIFICATION
- FROM T.F $G(z)$
- EXTENSIONS
  - ↳ STABILIZING MPC!

another important aspect of MPC...

Advanced and Multivariable Control

Model Predictive Control – Part 2

Riccardo Scattolini

next...

[until now we just used MPC → [theoretical development, guarantee of stability through MPC controller design]]

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

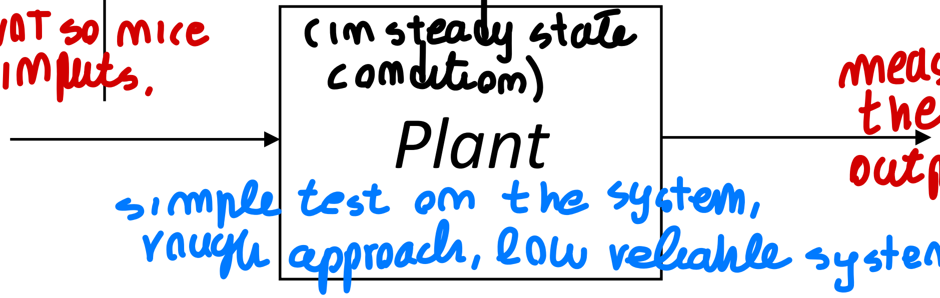
↳ when design a control for a big plant, decentralized (lots PID to tune) while by model based approach LQ controller ecc.. → effort on model development! hard to adapt the model with formal method

(impulsive input) forced

SISO

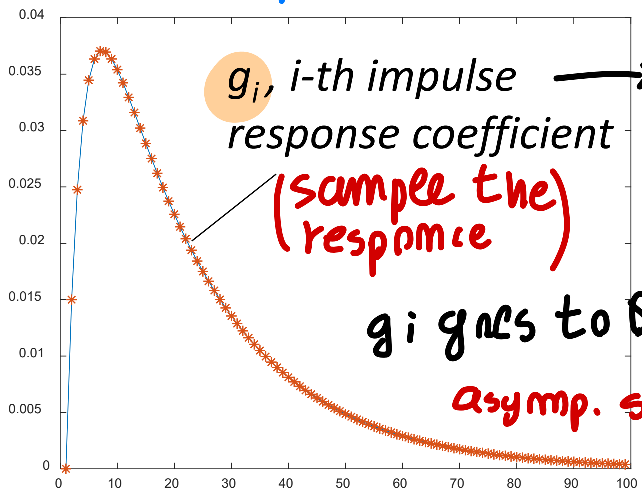
NOT so nice inputs.

(in steady state condition)



measure the output

simple test on the system, rough approach, low reliable system



g_i, i-th impulse response coefficient

(sample the response)

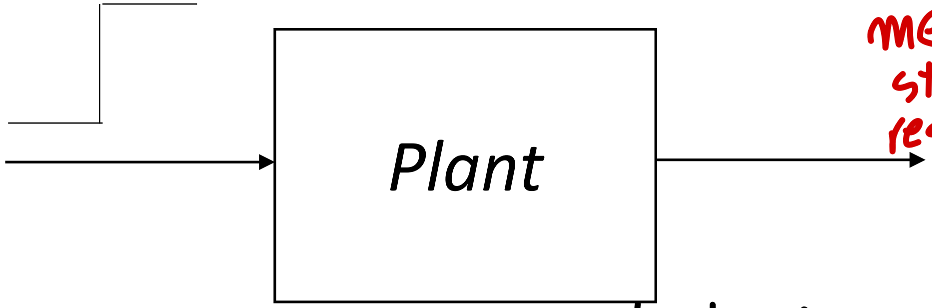
→ (IRR) to compute g_i

g_i goes to 0 @regime

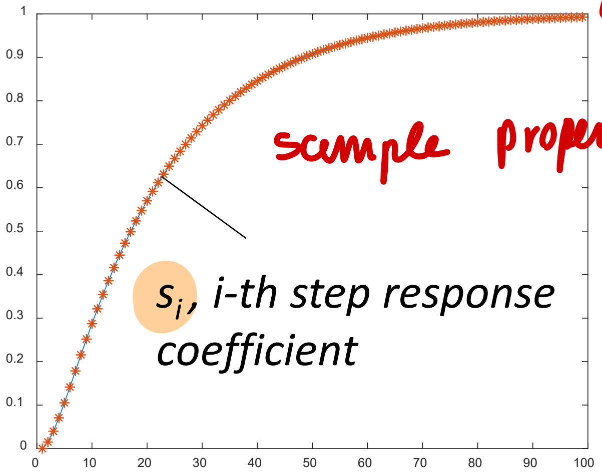
asymp. stable system

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

(step variation)



measure step responce



s_i, i-th step response coefficient

asymp. stable.. s_i goes to constant value

sample properly @ ΔT

(reduce the time to develop the model, based on expement)

⇒ you store s_i, g_i, dividing by input step size, to normalize respect gain

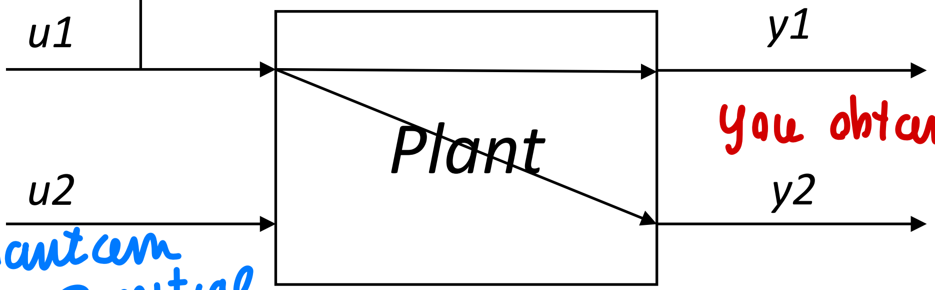
MIMO systems (2x2 case)

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

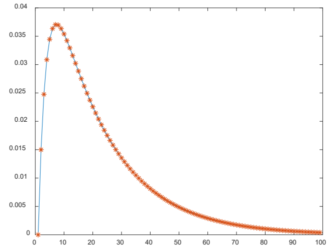
(procedure for MIMO system, collecting each var. response)

Experiment 1

apply impulse on u_1

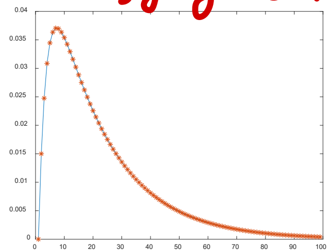


maintain u_2 @ initial equilibrium



g_i^{11}

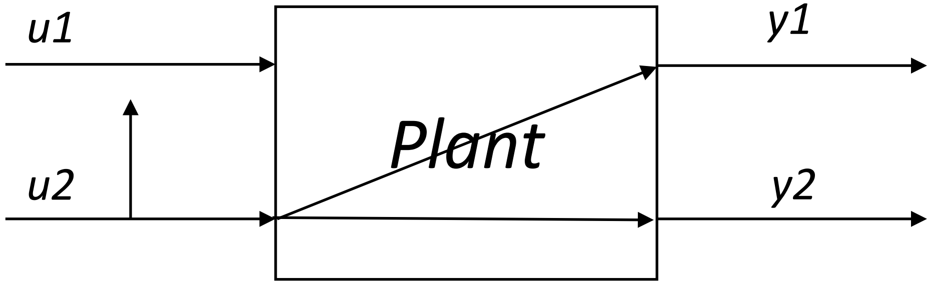
you obtain coeff of $u_1 \to y_1$ / $u_1 \to y_2$ effect

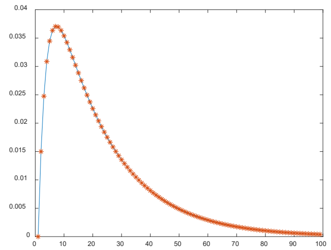


g_i^{21}

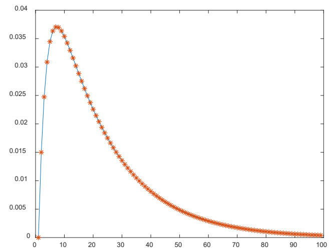
same experiment, BUT forcing the II input

Experiment 2





g_i^{12}



g_i^{22}

Valid for a generic plant!

Problems of this approach

- 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)

  ↳ variables moove a bit ecc... hard, and external regulation removed

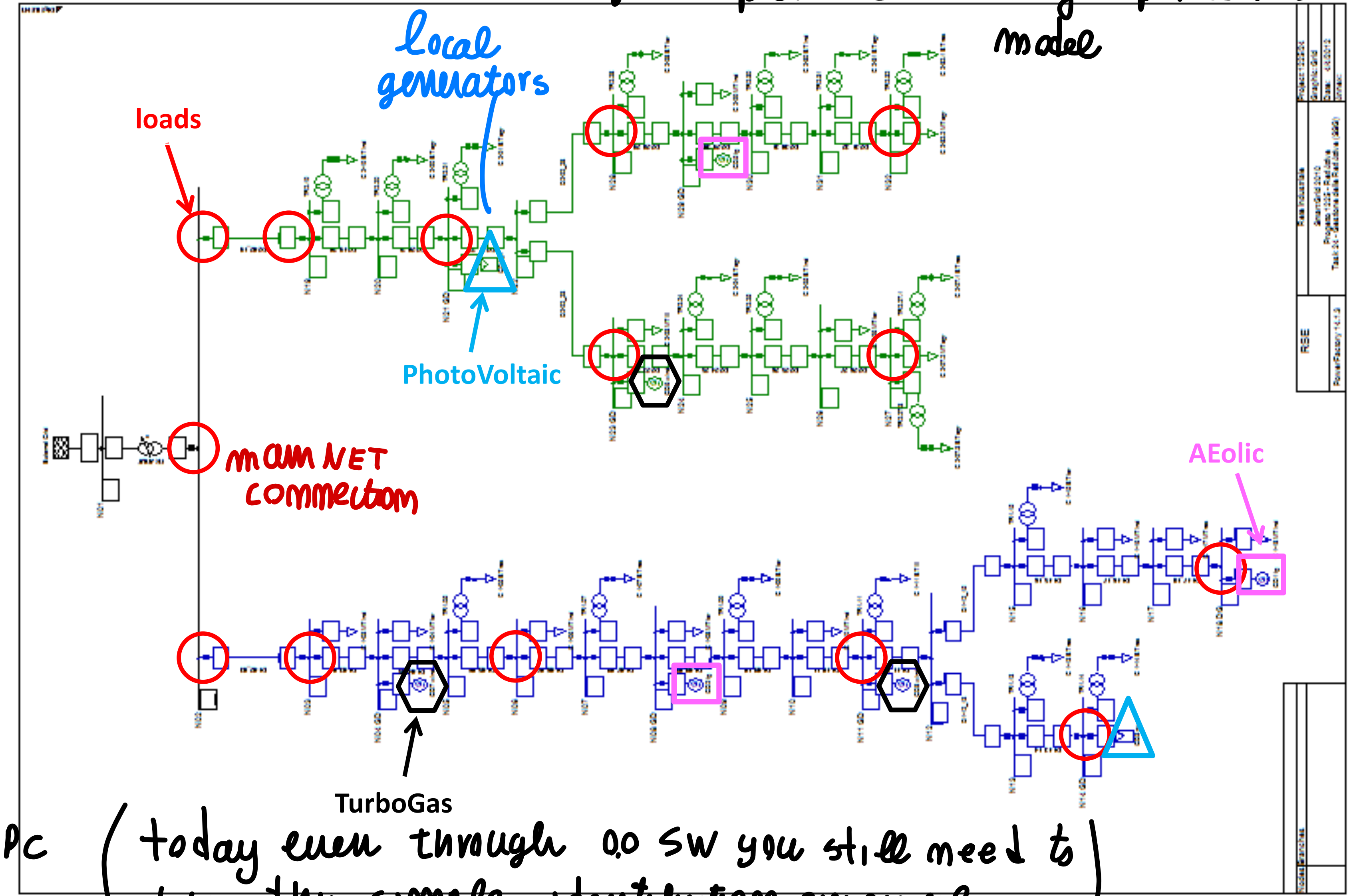
- Also during the experiment, disturbances should be avoided (how?)

- 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) approach used still today!

Voltage network control, inside distributed generators

Example: voltage control acting on the Distributed Generators power factors

we have to control this smart grid complex Net.. → using sophisticated model



Sophisticated model in DigSilent (environment for simulation of electrical systems) ↳ complex for control design!! (causality problem)

Too complex to design a regulator, but useful to compute an impulse response model ⇒ obtain nice local models to design MPC (today even through 00 SW you still need to use this simple identification empirical

Impulse/step responce ⇒ How to go from I.R to the s.s description?! T.F

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

(transform g_i to find syst. description)

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) \end{cases} \longleftrightarrow \boxed{g_i = CA^{i-1}B} \longleftrightarrow y(k) = \sum_{i=1}^{\infty} g_i u(k-i)$$

g_i related to s.s description of syst

(overall output)

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

$g_i \to 0$ because asymp. stable syst.

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

impulse response will go to 0 after large M number

↓ approximate model through finite numb. of coefficents

Therefore, the model can be approximated by

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

basic of MPC techniques: we should be able to predict output future based on input evolution by suitable model

approx model

$$\left[\; y(k) = \sum_{i=1}^{M} g_i u(k-i) \;\right]$$

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

FUTURE | PAST

$$y(k+i) = \underbrace{\sum_{j=1}^{i} g_j u(k+i-j)}_{\text{future control variable}} + \underbrace{\sum_{j=i+1}^{M} g_j u(k+i-j)}_{\text{what you have in the past}}$$

to select than the sequence of control action such that minimize cost funct

and, for $N<M$, $\Rightarrow$

(in matrix form)

$$\boxed{Y(k) = \mathcal{B}_g U(k) + \mathcal{B}_g^{old} \underbrace{U_{old}(k)}_{\text{old values of } U(k)}}$$

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

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

$y(k+1) = g_1 u(k) + g_2 u(k-2) + g_3 u(k-3) + ..$

$y(k+2) = g_1 u(k+1) + g_2 u(k) + g_3 u(k-1) + \dots$

$\dots$

$$\begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \end{bmatrix} = \dots$$

dependance on future values of u
and past values of u

by matrix B_g which multiply $U(k)$
and B_g^{oed} which multiply $U^{oed}(k)$ OF past inputs

future outputs $\equiv$ function of future input and past inputs
$\Rightarrow$ to state our probleem!

in that way we can consider the cost function easy expressed

... new formulation...

What is the problem with this formulation?

you obtain control var which does NOT depend on current value of state /action

$\|Y^o(k)-Y(k)\|^2+\|U(k)-\bar{U}(k)\|^2$ on cost function which doesn't depend on current state

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

future /past input → NOT on state /outp !

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** → min procedure does NOT depend on current time ! BAD ! ⇒ like an open loop solution based only on your model

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

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

$$\frac{\partial J}{\partial U}=0 \implies \boxed{U(k)=-(\mathcal{B}_g'\mathcal{B}_g+I)^{-1}\mathcal{B}_g'\mathcal{B}_g^{old}U_{old}(k)}$$

"open loop" control law → NOT as we desire !

no dependence on past outputs OR states..

idea: to write cost function as

$$J = \left(Y^o - B_g U(k) - B_g^{old} U^{old}(k)\right)^T Q \left(Y^o - B_g U(k) - B_g^{old} U^{old}(k)\right) + (\ldots)^T Q (\ldots)$$

I can derive $\partial J / \partial U$ → set to 0 → obtaining $\underbrace{U^{opt} = Matrix \cdot U^{old}}_{\text{state / output}}$

state / output removed in I.R ← I/O description without states ← BAD! I want a feedback control law

How to get rid of this problem?

We assume that an unknown disturbance affects the system

[to solve the feedback absence... we include disturb]

(disturb) approx/truncation of gi series..

$d(k) = y(k) - \sum_i g_i u(k-i)$

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

and the disturbance is constant

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

assumption

Therefore

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

sequence of u(k)

NOW y(k) influence!

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

cost function minimization

dependences on current y(k) value!
(feedback)

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

(Algorithm) of identification and control [IDCOM]

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

again feedback control law

(Exam) ↳ description in terms of I.R of the system (procedure) → troncate idea
+ equivalent disturbance procedure

Exercise

Consider the first order system in discrete time

$$\begin{cases} x(k+1) = 0.5x(k) + u(k) \\ y(k) = x(k) \end{cases}$$

- **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 $g_i=0.5^{i-1}$, $i=1,2,\dots$

(using gi formula written previously)

- **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:

future u values

past u values

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

- 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

first as before

$\pm y(k)$

including inputs which affect y @ current time subtract

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

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

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

dependence of U(k) on y(k) on algorithm

compact form → future output values as funct of future input / past inp. + current output

- **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

with linear constraints you can use Quad Prog solution
(while non lin dyn. has hard solution)

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

min respect future value U(k) and slack variable ε

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

syst. dynamic limitation
(evolution)

future error   quadratic in U   slack var coeff

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

subject to the previous dynamics and

$$U_{min} \leq U(k) \leq U_{max}$$ additional SATURATION on input

$$Y_{min} - \varepsilon(k) \leq Y(k) \leq Y_{max} + \varepsilon(k)$$ SAT on output

$$\varepsilon(k) \geq 0$$

hard constr on U(k) / soft constr on output

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.

modify y(k) constrain to have always feasible solution of MPC, necessary for each iteration

→ tends to excite too much the syst during testing

Relationships between impulse and step response coefficients

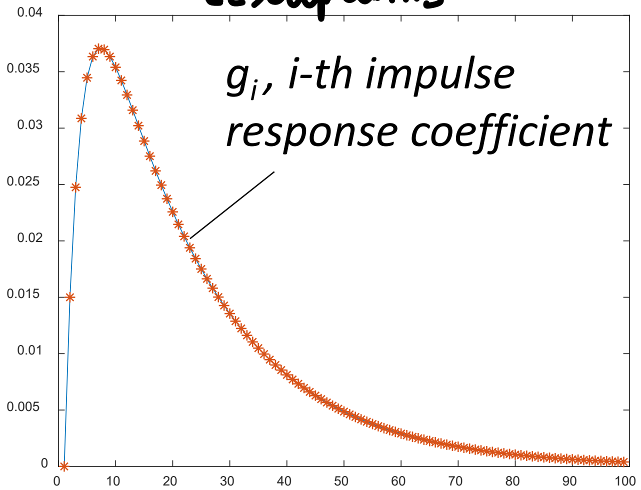
you can move between the two descriptions

↳ limited

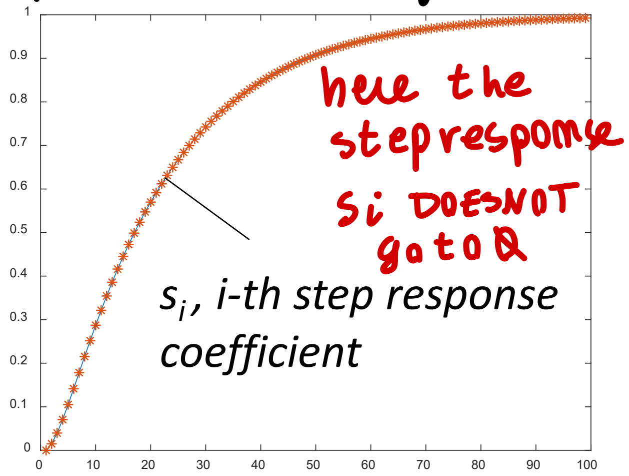
(relationship)

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

$g_i \to 0 \iff (s_i \to s_{i-1})$



(step ≃ impulse integral)



(no immediate effect)

Letting $s_0 = 0$ strictly causal syst

overall ∞ summ to take (NOT go to 0)

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

$(s_1 - s_0)u(k-1) + (s_2 - s_1)u(k-2) \ldots = s_1(\underbrace{u(k-1) - u(k-2)}_{\delta u(k-1)}) + s_2(\ldots) =$

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

$= s_1 \delta u(k-1) + s_2 \delta u(k-1)$

by this description $y(k)$ depends on δu (future control increment) over ∞ horizon

. finite summ
. use δu variables,
so dependency on current y

how to reduce ∞ summ to a finite one!?

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

$\Downarrow$ by previous trick

constant disturb

as different between output and ∞ summ

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

$\Downarrow$ OVERALL..

summ up to i, so future u value until

some terms cancel.. because $s_{M+i}=s_M$ after some time!

good formulation

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

$\delta u(k)$ variables to compute

current y value FEEDBACK

$s_{M+j}-s_j=0$ after some time..

physical actuator eval. limit : (easy to constrain) ←

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

δu MPC solut.

from δu you recompute $u(k)$ control variable

Finally

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

↑ integral action, NOT steady state to compute! good formulation

{ most famous MPC algorithm := DMC }

Dynamic Matrix Control - ***DMC*** – hundreds (thousands) of applications sampling you loose high freq components

{ model in low frequency, NOT a general representation! }

Notes

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

you can get a large M.. enormous # param. needed! → but with good SW NO Prob.

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

CONS.. lot param.

(no need to compute steady state val.)

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

↳ typically you can have complex disturb.. complex plants move steady state $\bar{x}, \bar{u}$ (you should retune / recompute $\bar{x}, \bar{u}$ each time!) NOT required with DMC!

- No need to use state observers

I.R / Step.R coefficients usage → Don't need to estimate state var.

PROS NO need of OBS → hidden inside coeff. used on identif.

Extensions of MPC..

simplest extension of basic MPC formulation

MPC with transfer function models

Consider SISO systems described by

model identification of T.F

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

how to design MPC from G(z)

possibly obtained with model identification procedures

move from I/O T.F to a s.s representation → and apply classic MPC theory!

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

you need also an observer! :⇒ system state not available

↳ or use T.F to build s.s represent. where...⇒

Solution 2: write the system in the state space realization

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

from G(s) build s.s where
x(k) is made of current/past In/Out values..

coefficents of the T.F

1st element

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

build system regression equation! easy to build matrix

↳ from here standard s.s formulation of MPC!

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

x(k+1) known ∀k: IN/OUT

and for ARX NONLIN models → non lin on the regressors you can include it ⇒ extendable for NONLIN (without asymp. value)
↳ EKF to estimate

MPC can be formulated for many optimiz/control problems related to different cases
→ lot of extensions to other issues !

Extensions

MPC is highly flexible and can cope with many different situations

(know possible extension for particular problems)

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

Lyap./stabilizing loop / variable structure approach

control design for N.L syst, small # solutions!

Extensions

in case of lin syst. use of lin models for control design is nice but not enough ... N.L techn allow good result

- ***Nonlinear MPC***: MPC for nonlinear systems (see later)

→ we consider discrete time cases... discretized cont. time → directly MPC on cont. time

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

- ***Stabilizing MPC***: MPC with stability guarantees (see later)

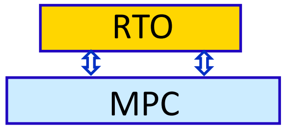
- ***Robust MPC***: the MPC problem is formulated also in presence of bounded disturbances with stability guarantees What if disturbances acting on the syst, uncertainty

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

Mix the two levels?

control main physical quantity after static optimiz. to compute set-point

- ***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

↳ how to deal with constraint with stochastic disturb !! := stocastic knowledge to deal with

(DATA driven approach!) → design C(s) from DATA

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

↓ switch / change of syst configuration..

⇒ *... a very long list* lot of extensions !!

MPC of nonlinear systems

typically NON LIN syst.. formulate directly on nom lin.

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)

discretiz. of cont. time plant model

System

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

Algorithm which rely on J gradient to find optimum

Cost function

⟹ many sw tools to solve this!
How to formulate formally and solve by sw toolbox

[NLMPC formulation]

$$\min_{U(k)} \; J = \sum_{i=0}^{N-1} l(x(k+i), u(k+i))$$

$l(x,u)$ positive definite at the origin

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

constrain on state, inputs

$\mathcal{X}, \mathcal{U}$ 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

↳ not easy algorithm of optimization

- 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

↑

SW tool "casadi"

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

optimization of non lin system!

⇓ possible simplification of the optimization problem!

Problem statement

nonlinear system

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

model

quadratic cost function

quadratic on state / control variation respect x^o, u^o desired reference trajectory

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

↳ linearize around this traj the model and use linearized model to formulate MPC

linear or nonlinear constraints

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

Solution 1 – linearization along the reference trajectories

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

$\Delta x, \Delta u$

Set

define deviation var. respect x^o, u^o desired ref.

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

and you know:

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

By means of linearization,

A, B matrix functions of time → lin. along traj which vary over t, make

(lin. model)

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

where

$$\left[\, r(k+j) = f(x^o(k+j), u^o(k+j)) - x^o(k+j+1) \,\right]$$

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

r = 0 if eq. holds, so state traj satisfy dym eq.

BUT with this approach you must know a priori x^o, u^o desired traj (which better if satisfy dyn.)

↳ NOT enough to specify x^o, you should have knowledge about u^o... cont.!

Linearize also the constraints

⇓

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

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

and finally formulate the QP problem

simple approach of N.L solution!

all respect $\Delta x, \Delta u$

linear cost funct + linear constr.

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

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

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

QP function (matlab) allow to evaluate the optimum!

Solution 2 – linearization along the predicted trajectories (not along desired one!)

@k-1 solved by min J

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

use this as initial guest sol.

"WARM START" to initialize optimiz.

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

computed!

applied

only first value applied

the others are similar to the optimal one... can I reuse it?

$$\Downarrow$$

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

use remaining seq.

NOT computed

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

reasonable future control seq

here I can repeat last value respect the one removed...

It is possible to compute the predicted state trajectory

obtained by apply past seq.

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

then linearize respect $\hat{x}$ predicted seq → similar to EKF! where you linearize around..

This predicted trajectory is used for linearization variation variable

Define
$$\begin{cases} \tilde{\delta}\hat{x}(k+i|k) = x(k+i) - \hat{x}(k+i|k) \quad , \quad i = 0, ..., N-1 \\ \tilde{\delta}u(k+i|k-1) = u(k+i) - u^o(k+i|k-1) \quad , \quad i = 0, ..., N-1 \end{cases}$$

Then

linearize your model! (standard lineariz.)

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

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

Letting

computable @ k when all this values are known!

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

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

(classical setting)

Final QP problem

min optimiz. problem

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

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

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

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

 (NOT asked on EXAM)

Solution 3 – nonlinear optimization … preliminaries

standard non lin. optimiz. approach → used on Matlab toolbox

Consider the nonlinear problem

(K.K.T conditions!)

$$\begin{cases} \min\limits_x f(x) \\ \quad g(x) = 0 \\ \quad h(x) \geq 0 \end{cases}$$

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

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 → based on gradient estimation and hessian, formulate sequence of QP to solve !

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

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

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

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

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

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

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

field of research → practical application (engine control USA)

- **Explicit MPC**

we see that on lin const, ∞ horiz → you don't have explicit sol NOT Real

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

linear dyn system

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

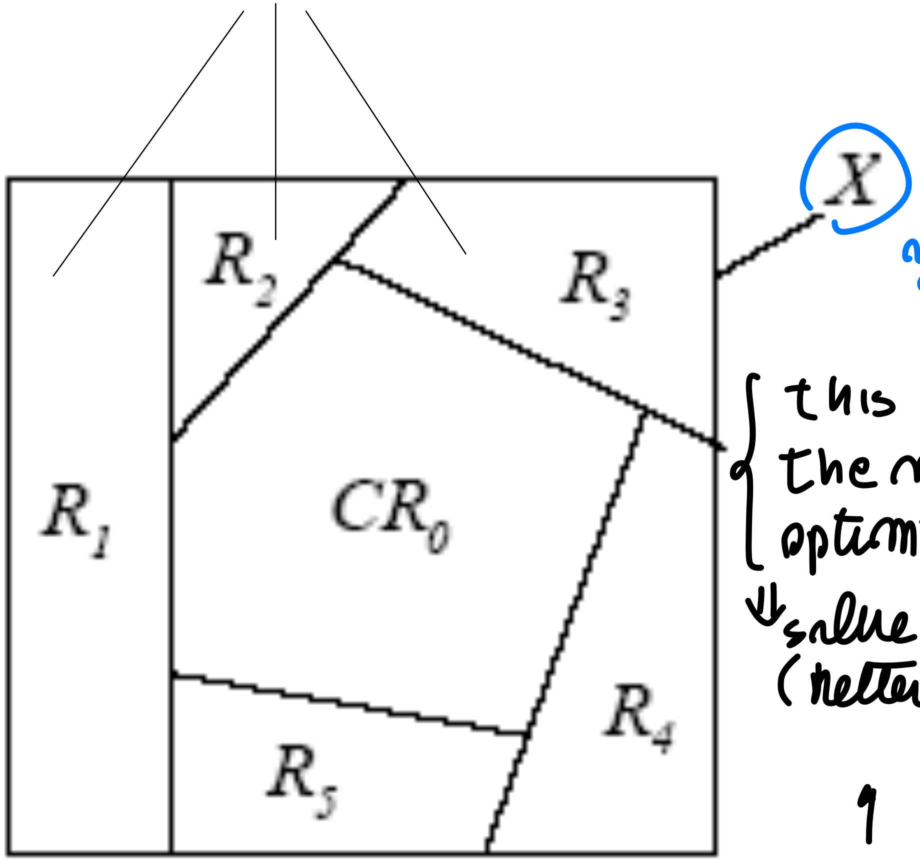
it is proven that can be solved...

in specific regions of the state space

iterative procedure,
divide region in zone... on each zone: the optical control law can be computed as linear + const. term

[constrained ∞ horiz prob. solution!]

compute all Regions Offline and correspondly control law



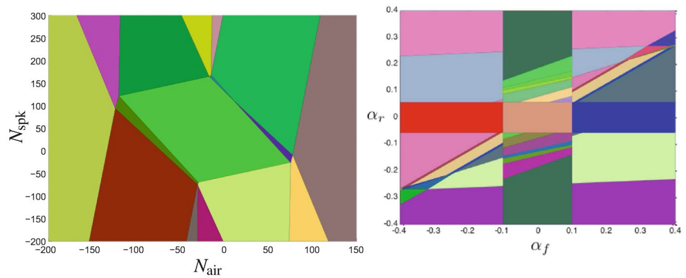
s.s region of your problem

this Explicit MPC avoid the need to solve online optimiz. problems

solve ∞ horiz problem (better than finite horiz.)

↳ then online I check where I am and apply correspondly control law

Possible approach

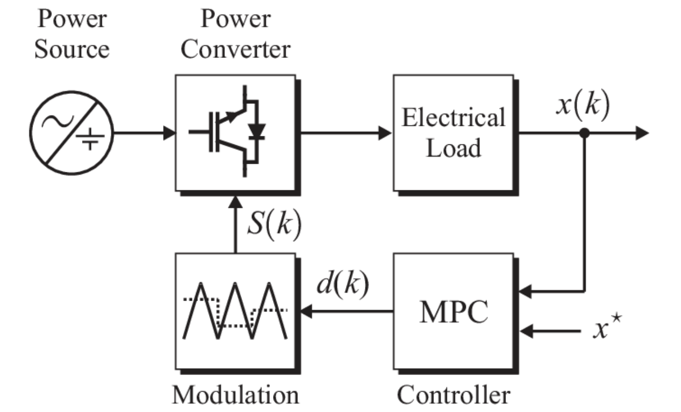


PROBLEMS...

1. Compute off-line the regions and the corresponding control laws
   ↳ # Regions can be almost ∞ !
2. Compute on-line the region where is the state and apply the corresponding control law
   ↳ not always easy check where I am! → critical computation! even harder than solve optimiz.

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)
  Good for low order mech syst.
- Already used in many industrial fields (automotive, power electronics,…)



- **Systems with on-off actuators**

in many industrial plants, you have actuators with ON/OFF values, so $u \in \{0,1\}$ →

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 $u(k) \in \{0, 1, 2, ..., n\}$

formulation of control problem as usual

BUT consider u constraint!

you have variable Re, Bol on optimiz. problem

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

Branch & Bound approach!

Binary choices, you try to follow optimal path

→ technique to solve the problem hard!

sw solver not always allow this problems

- **Hybrid MPC**

in many control problems you don't deal with only cont. variables...
sometimes you have satisfy logical constraint (for example power plant)
NOT JUST ON/OFF, you need to include a timer + logical constraint to set variables!

MPC can handle logical relations and constraints among the process variables

or system working on different config. switching valves/other logic var.!
hybrid systems (cont. time evolut + discrete event evolut)

Procedure

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

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

⇒ [important theory, a lot used in many control problems!]

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

logical var.
↳ reformulate logical relation among var. and constraints as

to state logic constr as MPC problem

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

propositional logical equation
logical condition by truth table

↳ reformulate logic constraint sequence as constr among boolean variable → algebraic constr among δi

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

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

⇒ formulate MPC and iterate procedure

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

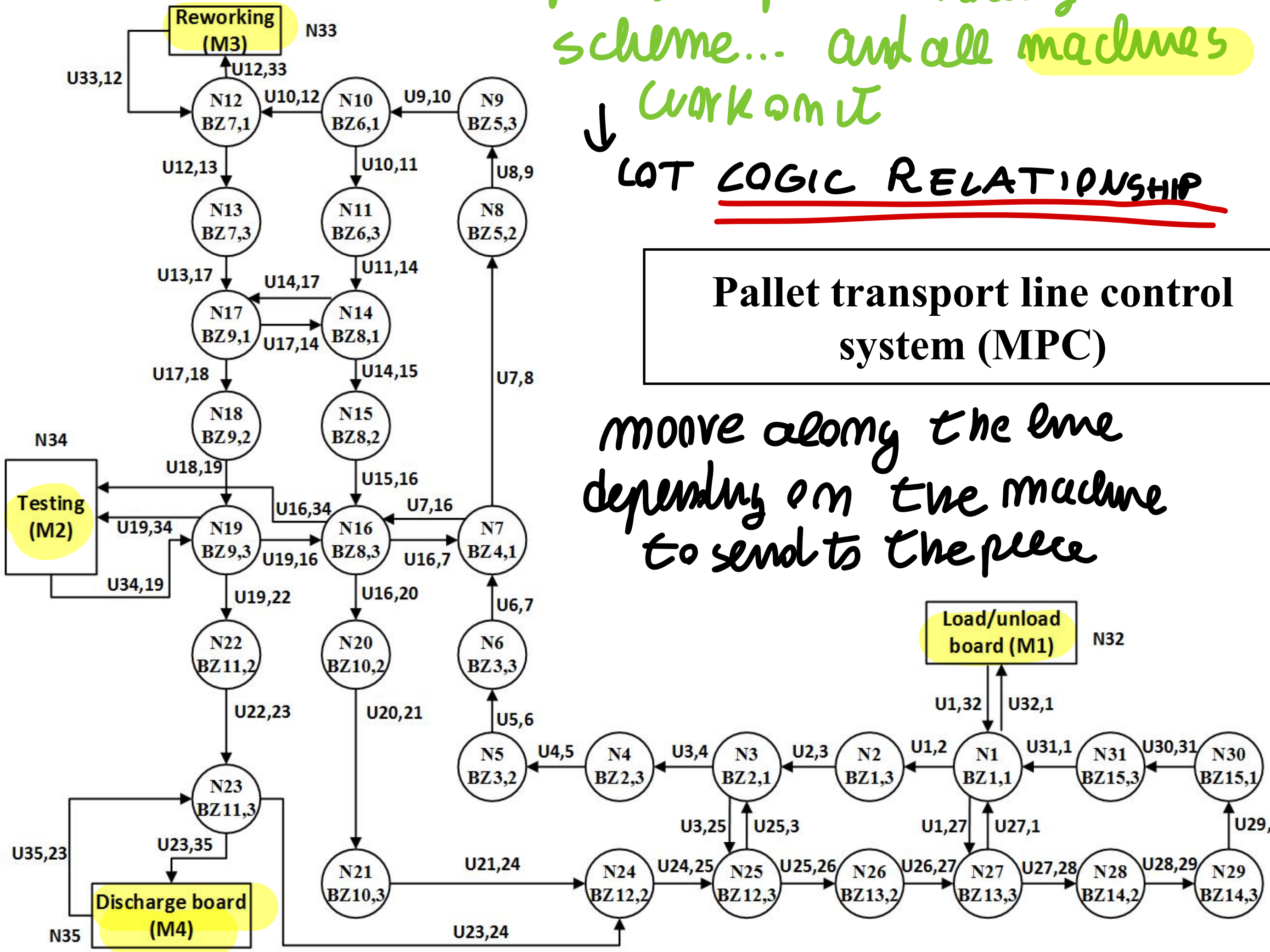
↳ overall model made out of cont. part + logical var. and this algebraic constraint

Example: MPC of a transportation line → seminar argument!

De manufacturing plant ~>

electronic board object are placed on pallets routing scheme... and all machines work on it

↓ LOT LOGIC RELATIONSHIP



Pallet transport line control system (MPC)

moove along the line depending on the machine to send to the piece

logic occupancy constraints...

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

to deal with MPC stability... We need an additional result!

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

by Def

@ k, in next k+1

set such that..

next $f(x(k))$ strictly inside the set X^o

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

assuming to know this X^o, and consider neighborhood of the origin

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

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

(valid for some non continuous lyap funct)

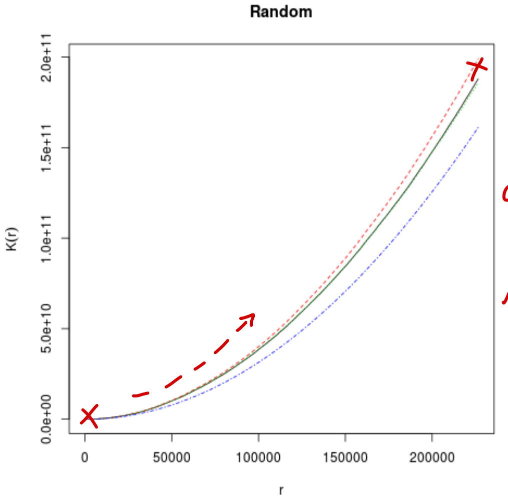
$$V(x) \geq w(\|x\|), \quad \forall x \in X^o$$

$$V(x) \leq \psi(\|x\|), \quad \forall x \in \mathcal{N}$$

V increment smaller than r

$$\Delta V(x) \leq -r(\|x\|), \quad \forall x \in X^o$$

$\Downarrow$



to move you take a non lin behaviour, non cont function

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 $\varphi(0) = 0$

 General addition result

Preliminaries – an extension of the Lyapunov theorem

assume to have an autonomous system

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

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

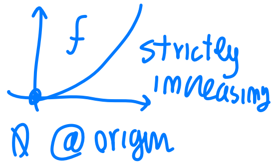
> Positively invariant set:
> if $x(k) \in X^o$, also $x(k+1) = f(x(k)) \in X^o$

consider the case:

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

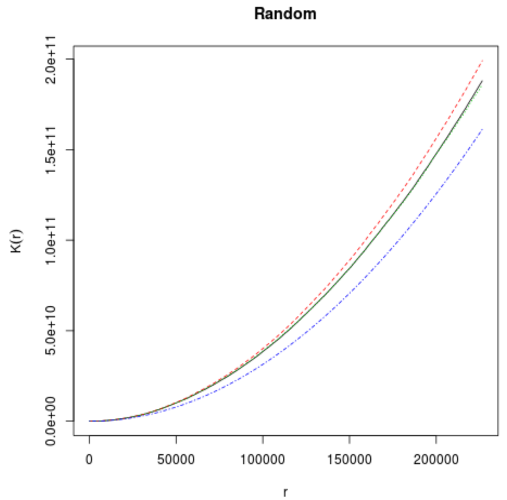
functions of the state



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

increment of V

(Extension of Lyap theory) required for non continuous Lyap funct



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 $\varphi(0) = 0$

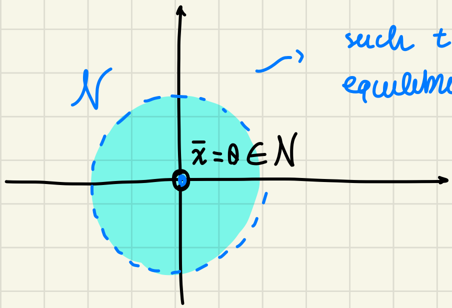
"POSITIVE INVARIANT SET"

for $k \uparrow$ you stay there

you remain here

you know a X_0 positive inv. set

and an N neighborhood of the origin



such that contains equilibrium $\underline{x}=0$

and there is a Lyap funct V

such that in X^0

$$V(x) \geq \omega(\|x\|)$$

ω is a class K function: so a function such that

$\downarrow$

$\omega(0)=0$ and strictly increasing

$\uparrow$

in MPC usually when going between two points...

go through ε on direction $\rightarrow$ NON CONTINUOUS task

$\downarrow$

to prove stability property we must check those conditions defined a pos. def function → another stability property relevant is...

A nother relevant idea for stability of MPC

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

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

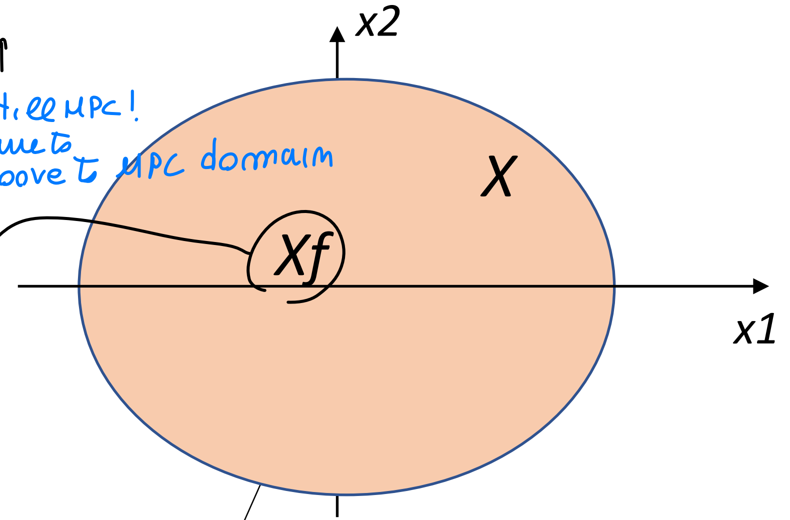
Just control law, NOT enough...

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

NOT still MPC! you have to moove to MPC domain

one has

apply $\kappa_a(x)$ inside X_f you remain inside..



$$x(k) \in X_f \quad , \quad k \geq \bar{k}$$

$$u(k) = \kappa_a(x(k)) \in U \quad , \quad k \geq \bar{k}$$

you have to know this auxiliary control law which stabilize → NOT best MPC solution!

↳ we wanna enlarge it

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

assume that your system $\boxed{x(k+1) = f(x(k), u(k))}$

↓

assume you know a stabilizing control law for this system

$u(k) = K_a(x(k))$ auxiliary C.L.

↓

such that, for this $u(k)$... you know a pos. inv. set $X_f \subset X$ such that for the closed-loop syst you obtain

$\hookrightarrow x(k+1) = f(x(k), K_a(x(k)))$

you have good result $\Leftarrow$ and $\forall$ state $\in X_f$

|| consider closed loop syst with auxiliary control law $K_a(x)$ ||

↓

for this syst you know X_f (pos. inv. set) where you remain here (because pos. inv), and in addition control law such that u satisfy its constraints.. $(u(k) \in U)$

⇓ $\left\{ \begin{array}{l} \text{we remain inside the set} \\ \text{and satisfy constrain} \end{array} \right\}$

$\hookrightarrow$ mean that START near X_f, using $u(x)$ we remain inside set and control var. constraint are guaranteed

near the origin $u(k)$ remain near...

$u = K_a(x)$ satisfy constrain...

goal of MPC is to enlarge this region of stability

↑

simple by linearizing near origin, set up LQ_∞ control law on lin. system → remaining near origin $u(k) \in U$ (respect contraints!)

(you can guarantee that constraints are satisfied...)

FORMULATION

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

and apply only first one (RH)

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

computing future control var sequence

by quadratic cost

modify cost function

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

[Quadratic cost function]

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

add an additional weight! Terminal, on final prediction horizon

subject to

@ end of prediction horiz. we are in X_f

+ constraint that final state should be inside X_f

KNOWN constraint

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

final state inside X_f constraint

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

additional elements of MPC, to design a stabilizing control.. MPC

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

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

(solving $\forall$ time)

@ end of prediction horizon $x \in X_f$!!

(constraints can limit the space where solution exist (depending on N))
I take the set of x where solution exist

Theorem

where auxiliary c.L. is defined holds

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 (∀ state on Xf)

QK

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

terminal cost evaluated first on x(k+1) then on x(k)

|| condition for recursive feas + stability

is fulfilled and

$$V_f(x) \le \alpha_f(\|x\|)$$

$\Downarrow$

where $\alpha_f(\|x\|)$ 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 $\alpha_f(\|x\|) = b\|x\|^2$ and $X_f = X^{RH}(N)$~~

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

and also $X(N) \supseteq X_f$

IF you choose V_f of the J that satisfy this (*) constraint... you can guarantee A.S of the origin! suitable Vf to define!

→ in a practical P.O.V I am to be able to define it (NOT the proof of that Theorem)
suitable Xf, Vf to choose !?

check main MPC issues...

Proof

Recursive feasibility (PROBLEM 1)

for Recursive feas. should be feasible to solve also @ k+1...

assume @ k solve optimiz (FEASIBLE)

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

optimiz. problem solved @ k

sequence over prediction horizon...

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

check feasibility over prediction horizon

feasible use of auxiliary CL inside X_f!

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

take all sequence optimal @ k... with additional term

recursive feas. if @ k+1 we have solution of PROBL.

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

(shift the previous sequence) NOT optimal @ k+1

term as terminal

(auxiliary control law)

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

I'm on my set X_f I remain here! feasible

⟹ Moreover,

optimal cost function

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

$w(\|x\|)$

greater than $\|\cdot\|_Q$ of first element of

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

K function class!!

satisfied here!

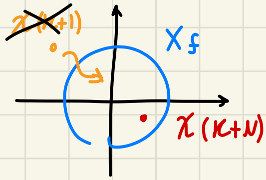
Idea... @K I compute optimal sequence

$u(k)$ $u(k+1)$.... with constraint $u(k+N)$ such

$x(k+N) \in X_f$

↑ additional constraints to guarantee

With optimal I remain inside, @end using auxiliary C.L.



@ k+1
I can be outside..

using previous sequence I guarantee @ end reach $x(k+N)$

I use inside X_f the $u_a(k)$ and guarantee to remain on X_f

We should check V is decreasing ... by a different P.O.V

At time k, the sequence

✓ taking opt. sequence + auxiliary

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

{solve optimization with horizon (N)}

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

I remain in Xf due to constraint..

optimal @ N - removing... terminal cost + add N+1 cost

$$J(x, \tilde{U}(k, N+1), N+1) = V(x, N) - V_f(x(k+N)) + V_f(x(k+N+1))$$
$$+ \|x(k+N)\|_Q^2 + \|\kappa_a(x(k+N))\|_R^2 \leq 0$$
$$\leq V(x, N)$$

take N sol, add terminal N+1 cost

+ values quadratic..

IF condition of hyp. SATISFIED! on inequality

to get the optimal value..

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

without predict. horiz!

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

decreasing property of optimal cost function, progressively decrease..

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.

{satisfied condition for V inside Xf}

We already find $V(x) \geq \|w(x)\|$ in $X_0 = X_f$ here!

Let's check that $\Delta V(x) \downarrow$ (other condition)

At time k the sequence

in a different P.O.V

feasible still!

IF using N+1

previous optimal sequence

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

I can have feasible sol

add auxiliary C.L

solve optimiz with horizon N (ensure @ end you stay in X_f)

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

see the sequence which effect has...

we look to horizon N

remove terminal cost, NOT part of form.

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

add termal cost

from hyp condition

from

$V_f(\cdot\cdot) - V_f(x) + (\cdot\cdot\cdot) \leq 0$ theorem condition

$(*) \rightarrow$

$$\leq V(x, N)$$

this is all $\leq V(x, N)$! $(\leq *)$

this J is NOT the optimal one

$\Rightarrow$ so that we have the monotonicity property (with respect to N)

$N = 0$

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

so you have a decreasing property of optimal cost function!

optimal cost progressively decrease of $N\uparrow$

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

you remove initial

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

cost just maintain

$V(x)$ is smaller than $V_f(x) \leq \alpha_f(\cdot)$

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

final value

(2nd condition on initial)

N of theorem (using κ_a in X_f)

Finally

we can rewrite V in this way...

cost function from k+1 to N-1

$$V(x,N) = \|x\|_Q^2 + \|\kappa_{RH}(x)\|_R^2 + J(f(x,\kappa_{RH}(x)), u^o(k+1, N-1), N-1) \ldots$$

$$= \|x\|_Q^2 + \|\kappa_{RH}(x)\|_R^2 + V(f(x,\kappa_{RH}(x)), N-1)$$

$$\geq \|x\|_Q^2 + \|\kappa_{RH}(x)\|_R^2 + V(f(x,\kappa_{RH}(x)), N)$$ ← bigger by horizon N

$$\geq \|x\|_Q^2 + V(f(x,\kappa_{RH}(x)), N), \quad \forall x \in X^{RH}(N)$$

$\Downarrow$

$V(x,N) - V(f(x,..)) \geq \|x\|_Q^2$ decreasing cond satisfied!

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. (stability!) satisfy all our request!!

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

↳ IMPORTANT Remarks! ⟹

Difficult task $\to$ how to chose X_f, V_f , $K_a(x)$

Remarks to properly choose the problem terms!

Remark 1

basic idea: use optimal cost funct J as Lyapunov...

you can guarantee it's decreasing! by MPC you hardly find optimal solution @ any instant!

in general you don't need optimal MPC just guarantee decreasing function..

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

$$\bar{U}(k) = \left[\bar{u}_k(k) \quad \bar{u}_k(k+1) \quad \cdots \quad \bar{u}_k(k+N-2) \quad \bar{u}_k(k+N-1) \right]$$

sequence such that cost funct decreases...

such that

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

find a sequence reducing J (NOT the optimal)

Remark 2

It is possible to conclude that

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

set of states where optimal sol exist, becomes larger as the horizon enlarges!

greater possibility to satisfy requirements

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

Using a large prediction horizon brings advantages!

Remark 1

basic idea: use optimal cost function as lyap...
you need to guarantee that it is decreasing J↓

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

⇒ numerically issues on optimal solution ∀t (HARD even by good sw)

$$\bar{U}(k) = \begin{bmatrix} \bar{u}_k(k) & \bar{u}_k(k+1) & \cdots & \bar{u}_k(k+N-2) & \bar{u}_k(k+N-1) \end{bmatrix}$$

such that

→ generally: you don't need optimal MPC solution to guarantee stability :⇒ you need that J↓ decrease

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

IF you find seq. such that J↓ ok! decreasing property → NO optimality, find sequence u which reduces J (ROBUSTNESS)

Remark 2

It is possible to conclude that

iterate lot of time, it converges to ∞ ! allow to approx ∞ horizon C.L

set of state where opt. sol exist becomes larger as N↑

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

↳ means that IF N↑ you have more DOF ! := great possibility of sol

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

LARGE prediction horizon → better MPC behaviour! better results than auxiliary control law..

you can prove that $\forall N$
X_f included on RH set...

Remark 3

set to know where constrain
satisfied...

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

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

by MPC approach you enlarge your problem feasibility!

Remark 4

enlarging sufficently N the set of states where feasible, include
enlarged set.

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?

large horizon on MPC by iterate, better solution!
↳ Iterate riccati equation converging on the best solution $\simeq \infty$ horizon control law

why NOT using JUST auxiliary c.L ?! instead of solve MPC...

Remark 3

$\forall N$ can be proven sol. include X_f where Ka exist..
so with RH MPC you enlarge feasibility region of problem

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

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

Remark 4

enlarging enough N, set of state where N is feasible include a larger $\bar{X}_f$ where Ka exist!

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.

take large N!

usually LQ more used in those field...

But, how to select the terminal cost and the terminal set? ? ⇒ how apply this Theory

find (V_f, X_f ?)

how to find this parameter, even IF NOW to guarantee MPC it is still lot used quadratic cost J

how to apply this theory in practice?...

⇒ **Algorithm 1: zero terminal constraint** approach

This is the first algorithm proposed, defined by

We have to find out V_f and X_f and auxiliary c.l.

$$\kappa_a(x) = 0$$
$$X_f = \{0\}$$
$$V_f = 0$$

$$x(k+N) = 0$$

It can be used as it is for NON linear systems!

LINEAR/NONLIN

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

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

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

$$U(k+1) = \left[\; u_k^o(k+1) \quad u_k^o(k+2) \quad \cdots \quad u_k^o(k+N-1) \quad 0 \;\right]$$

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)

Focus on linear, even if holds for NON LIN...

NOW FOCUS

Consider the linear system

we can choose ∞ horizon control law LQ

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

NOT applyed on system!

LQ∞ as auxiliary, we know it stabilize and small x bring to remain on U!

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

because it stabilize the system with small u (satisfy constraints)

this is used for proof, NOT for control

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

auxiliary control law (use LQ∞ control law)

where I just use MPC!! (just to formulate) My proof, then I use u_{RH}!! Never used!!

Define the matrix *P* solution of

steady state sol of Riccati eq

to choose X_f, take R.E solution of LQ control law

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

take steady state sol

as terminal set

and the terminal set

take this quadratic form with small α ✓

you stay inside this set thanks to u(k) →

$$X_f = \{x \mid x'Px \le \alpha\} \subset X$$

where α is a sufficiently small value.

guarantee to stay inside set..!

even in the unconstrained case → the optimal J of LQ control can take

$\sqrt{x^T P x}$ level line (ellipsoid) form... as lyap function!

↓

- for a small region around the origin $\bar{x} = 0$ you don't cross the limits

$x^T P x \leq \alpha$ α small

↑ you remain in your initial statespace set

⇓

positive invariance property

⇐ all x, u remain on region

x, u inside constraints!

corresponding to different α

α hard to select...

$\Rightarrow$ we have to decide the terminal cost V_f last term to chose

Consider also the terminal weight

taken in this way

is like writing ∞ horiz. cost function approx. $\leftarrow$

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

With this choice, we guarantee R.E @ end Q

These choices fulfill the stability condition

that guarantee the condition as RICCATI EQ .. = 0

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

In fact

$$\Gamma(x(k)) := V_f\left(f\left(x(k), -K_{LQ}x(k)\right)\right) - V_f(x(k)) + \left(\|x(k)\|_Q^2 + \|\kappa_a(x(k))\|_R^2\right)$$

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

$$= 0$$

Result can be interpreted as: $V_f(x(k+N))$ with x^TPx @ $k+N$ $\rightarrow$ ∞ horiz cost function using LQ without constraint

Simple: solve R.E, find P, set $V_f = x^TPx$ on MPC using $u = -K_{LQ}x$ with those X_f

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

$$\Downarrow$$

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

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

to deal with N.L syst in quasi ∞ horiz approach

Quasi infinite horizon for nonlinear systems

same approach (1) used without modifications, while (2) quasi ∞ horiz..

First assume that the system is linearizable at the origin

algorithm 2: linearize syst..

It can be prooven as before

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

LINEARIZE

where

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

Use linear control law as auxiliar c.l

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

enlarges to deal with NON LINEARITY

solve Riccati equation with more coefficents

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

and consider again the terminal cost

⇓ use again final cost as quadratic

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

In the neighborhood of the origin,

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

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

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

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

regulating syst when we wanna trak a given constant value x_s, u_s

FINAL POINT: to regulate the system tracking a given costant x_s, u_s values ($\neq (0,0)$)

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

we can moove all to new origin x_s, u_s reformulate all!!

moove all to new origin
reformulate all as variation respect equilibrium

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

as function of final diff

the condition to be fulfilled in X_f is

in the same way → condition to satisfy with different approaches

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

MOOVE to new eq

(ALG 1)

zero terminal constraint

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

NO weight @ end of prediction horizon
With X_f = ORIGIN (here $\{x_s\}$) to regulate @ x_s

here κ_a as desired $u(k)$

desired equilibr.

to regulate equilibrium @ x_s

again solution of R.E

(ALG. 2)

quasi infinite horizon

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

QUADRATIC

weight also $(X - X_s)$

LQ solution

relate to new eq

(Just moove to new equilibrium point)


The end

CONTROL and OPTIMIZATION OF an INDUSTRIAL PLANT → design an MPC

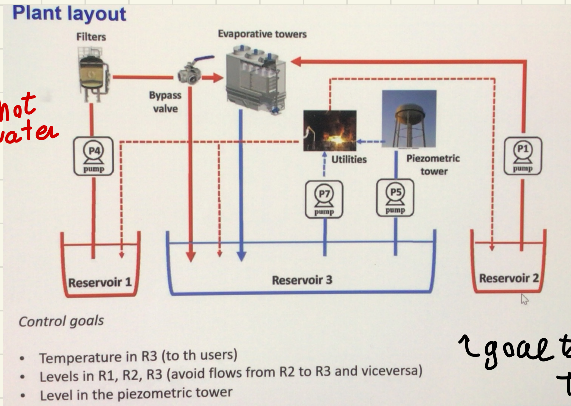
Water cooling of a large steel plant, heat to remove

large energy consumptions, by ON/OFF pumps and fans

cooling system

↓

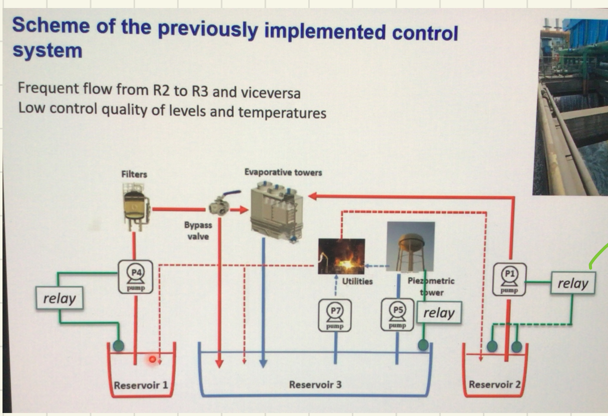
hot water



group of pumps mooves water where needed

goal to mantain levels and temperature $\leq 30°$ OR impossible to cold..

control action such that you have overflow of energy → wasted heat!



relay control on the pump to control the LEVEL

ON/OFF control, limited effort, with lots examples

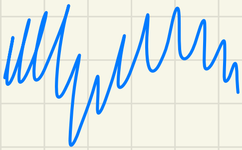
developing a phisical model of the system → complex plant,
approx model..

we need a phisical based → where testing
to check if it works properly!

partition water to check if as expected

\+ noises sinusoidal,
different behavior daily

due to pumps 1/0



design a non lin observer, MPC controller, simulate all!

↓ Dynamic model found through mass/energy balance

values estimated by spline function..

↓

Formal description

$$\dot{x}(t) = f_c(x(t), u(t), d(t))$$

$$x = [\ h_1 \quad h_2 \quad h_3 \quad T_1 \quad T_2 \quad T_3 \quad h_p\ ]'$$

$$u = [\ n_{wp1} \quad n_{wp4} \quad n_{wp5} \quad f_p\ ]'$$

$$d = [\ n_{wp7} \quad T_{ex} \quad \varphi_{ex} \quad w_{h1} \quad T_{h1} \quad w_{h2} \quad T_{h2} \quad w_{h3} \quad T_{h3} \quad \alpha\ ]'$$

$$d_m = [\ n_{wp7} \quad T_{ex} \quad \varphi_{ex}\ ]'$$

$$d_{nm} = [\ w_{h1} \quad T_{h1} \quad w_{h2} \quad T_{h2} \quad w_{h3} \quad T_{h3} \quad \alpha\ ]' \rightarrow$$ NON measured disturbances

assume $\dot{d}_{nm} = 0$

$\chi = [x' \quad d_{nm}']$ enlarged state

using different sampling time..

made estimator like an EKF

taking all equation of the model...

L_i tuned by linearized model

+ disturbances estimation → good effective practical solution

we linearize around predicted traj...

repeating.. standard MPC

↓

standard J definition ⤳ $\min\limits_{\Delta U(k), E(k)} J$ ...

+ constraint on level

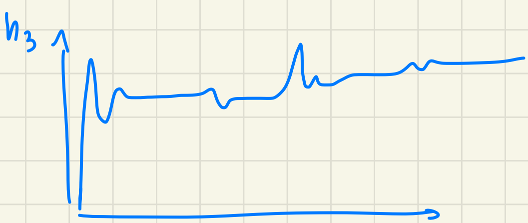
with commands not continuous → mixed problem!

transient cannot be avoided due to ON/OFF, obtaining $h_i < 100\% \ \forall t$ (good level!!)

↓ compary MPC vs Relay control

you gain 5% of energy (good enough gain!!)

+ check what if using modulating pump (expensive solution)



remove oscillatory behaviour

significant energy saving!

Testing on the system → technical problems of overwrite!

MPC engeneery application for real control cases!

by PI, Relay is complex → sometimes better to include constraints and centralized control