Model Predictive Control — Lecture 1
Introduction and Motivation

Ling KV (ekvling@ntu.edu.sg)

January 2025

EE6225

Part 1: Model Predictive Control

References:

1. Maciejowski, Predictive Control with Constraints, Prentice-Hall 2001. [NTU Library, TJ217.6.M152]
2. Rossiter, Model-Based Predictive Control, A Practical Approach, CRC Press, 2003. [NTU Library, TJ217.6R835]
3. Camacho and Bordons, Model Predictive Control (2nd Ed.) Springer 2004. [NTU Library, TJ217.6 C172m 2004]
4. Liuping Wang, Model Predictive Control System Design and Implementation Using MATLAB, Springer, 2009. [E-book available in NTU Library]

Some newer references/textbooks

Several other textbooks exist:

1. Rawlings, Mayne and Diehl, Model Predictive Control: Theory, Computation and Design, (Nob Hill Publishing, Second edition, 2017).
2. Borrelli, Bemporad and Morari, Predictive Control for Linear and Hybrid Systems, (Cambridge University Press, 2017. Online materials: http://www.mpc.berkeley.edu/mpc-course-material).
3. Goodwin, Seron and De Dona, Constrained Control and Estimation: an Optimisation Approach, (Springer, 2005).

Recommended Readings and Warm Up Exercises

- Required Readings
  - J.M. Maciejowski, Chapters 1,2,3,4, 7 and 9
  - Liuping Wang, Chapters 1, 2, 9 and 10
  - Qin and Badgwell, A Survey of Industrial Model Predictive Control Technology, Control Engineering Practice, Vol.11, pp.733-746, 2003.
- Warm Up Exercises: Do problems 1.3, 1.4, 1.5 and 1.6 in Chapter 1 of Liuping Wang.

Review (Do you know this already?)

- Models of Dynamic Systems
  - Continuous-time vs Discrete-time
  - Transfer function vs State Space
- Basic setup of a computer controlled system
- Matlab/Simulink

Continuous Time (CT) vs Discrete Time (DT) Systems

- CT(DT) linear dynamic systems are described by linear differential(difference) equations
- CT system example: The spring-mass system

$$M\frac{d^2y(t)}{dt^2} + B\frac{dy(t)}{dt} + Ky(t) = u(t)$$

- DT system example
  - Bank savings account balance at the k-th month

$$y(k+1) = (1+\alpha)y(k) + u(k)$$

  - "Transformed" CT system

State Space Description of CT/DT LTI Systems

- CT / DT Linear Time Invariant (LTI) System:

$$\begin{aligned} \frac{dx(t)}{dt} &= A_c x(t) + B_c u(t) \\ y(t) &= C_c x(t) \end{aligned} \qquad \Bigg| \qquad \begin{aligned} x(k+1) &= A_d x(k) + B_d u(k) \\ y(k) &= C_d x(k) \end{aligned}$$

- Input vector $u(t), u(k) \in \mathcal{R}^m$
- Output vector $y(t), y(k) \in \mathcal{R}^p$
- State vector $x(t), x(k) \in \mathcal{R}^n$, n is the order of the system
- State equation: $A_c, A_d \in \mathcal{R}^{n\times n}, \quad B_c, B_d \in \mathcal{R}^{n\times m}$
- Output equation: $C_c, C_d \in \mathcal{R}^{p\times n}$

Exercise

Suppose that a continuous-time system is described by the transfer function

$$G(s) = \frac{\omega^2}{s^2 + 0.1\omega s + \omega^2}, \text{ where } \omega = 10.$$

- Represent the transfer function in `MATLAB`
- Compute the state space representation
- Compute the transfer function using the state space representation obtained in the previous step, and check that you get the same transfer function back
- Compute the ‘ZOH’ DT system using a sampling interval of $h = 0.01$. (hint: use the c2d command)

A Simple Closed-loop Simulation [MATLAB Program]

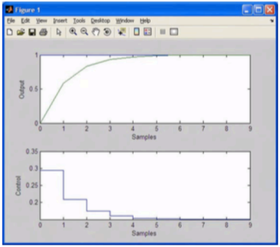
```
% A simple matlab program to closed-loop control

clear all
% define the plant and set-point
az = [1 -0.7]; bz = [2];
w = 1; %set-point

% storage for simulation
yplant=0; uplant=0; ydata=[]; udata=[]; wdata=[]; tdata=[];

for k=1:10
    %measure the current plant output
    yk = yplant;
    % calculate the MPC control
    uk = (w - 0.49*yk)/(1.4+2);
    % store the data for plotting later
    tdata = [tdata; k-1]; wdata = [wdata; w];
    ydata = [ydata; yk];  udata = [udata; uk];
    % apply the control to the plant
    uplant = uk;
    % simulate the plant
    yp_new = -az(2)*yplant + bz*uplant;
    yplant = yp_new;
end

subplot(211),plot(tdata,wdata,tdata,ydata),ylabel('Output'),xlabel('Samples')
subplot(212),stairs(tdata,udata),ylabel('Control'),xlabel('Samples')
```



Why MPC?[a]

[a]See also: J.M. Maciejowski's presentation slides, NTU/EEE, September 2004

- MPC is the only advanced control technology which has made a significant impact on industrial control engineering
- It has so far been applied mainly in the petrochemical industry, but is currently being increasingly applied in other sectors
- The main reasons for its success in these applications are:
  - It handles multivariable control problems naturally
  - It can take account of actuator limitations
  - It allows operation closer to constraints (compared with conventional control), which frequently leads to more profitable operation
  - It is intuitive, and "easy to tune"

See: Qin and Badgwell (2003), A survey of industrial model predictive control technology, Control Engineering Practice, Vol.11, pp.733-764.

Advantages of MPC

- particularly attractive to staff with only a limited knowledge of control because the concepts are very intuitive and at the same time the tuning is relatively easy.
- can be used to control a great variety of processes, from those with relatively simple dynamics to more complex ones, including systems with long delay times or non-minimum phase or unstable ones.
- multivariable case can be easily dealt with.
- intrinsically has compensation for dead times.
- introduces feed forward control in a natural way to compensate for measurable disturbances. law.
- extension to the treatment of constraints is conceptually simple, and these can be systematically included during the design process.
- very useful when future references (robotics or batch processes) are known.
- a totally open methodology based on certain basic principles which allows for future extensions.

Disadvantages of MPC

However, MPC also has it drawbacks.

- The derivation of the control law is more complex than that of classical PID controllers.
- When constraints are considered, the amount of computation required is high.
- The need for an appropriate model of the process to be available.

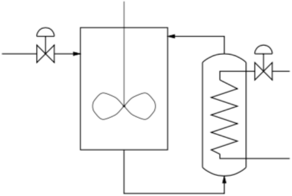
Why are Constraints Important?

- The reason cited most often for the success of MPC in the process industries is that the most profitable operation is often obtained when a process is running at a constraints, or even at more than one constraints.
- Often, these constraints are associated with direct costs, frequently energy costs:
  - if a product requires heating during its manufacture, for instance, the manufacturing cost can be minimised by keeping the heat supplied as small as possible, yet to be sufficient, and this is a constraint on the manufacturing process.
  - if a product has to be of at least a certain quality in order to be useful, the cost can usually be minimised by making its quality just sufficient; this is a constraints which has to be respected.

See also Examples 1.1 and 1.2 in [JMM2001]

Example 1.1 of [JMM2001] Semi-batch Reacter

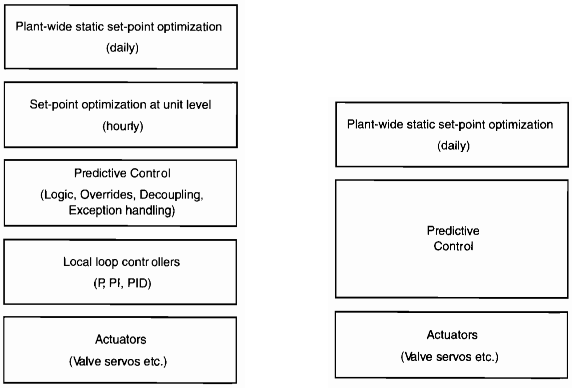
- Keep temperature close to target
- Keep water flow rate between constraints
- Keep reactant flow rate between constraints
- Minimise time to finish the batch — ie maximise reactant flow rate



What is MPC?

- MPC = Practical Optimal Control (Multivariable and can deal with process constraints)
- At each time instant, MPC
  - uses a current measurement of the process output, and
  - an internal model of the process, to
  - compute and implement a new control input, which
  - minimises some cost function, while
  - guaranteeing that constraints are satisfied
- Usually the control input is implemented in a receding-horizon fashion
- We have feedback because a new control input is computed for each new measurement

Use of MPC in Process Control, Current and Future Trends?



For a historical insights of the motivation of MPC, see: Richalet, et al, Model Predictive Heuristic Control: Applications to Industrial Processes, Automatica, Vol.14, pp.413-428, 1978.

MPC ‘de-skills’ control system design

Problem	Usual solution	MPC solution
Delay in plant	Phase advance	Long horizon
Unstable plant	Encirclements by Nyquist plot Bandwidth high enough	Long horizon Fast reference trajectory
Non-minimum phase	Bandwidth low enough	Long horizon Slow reference trajectory

MPC gives easier ‘parametrisation’ of controllers

(intuitive statement — not theorem)

Prescription for using MPC

- If you can do it with a PI or PID controller, then do it.
- If you need anything more complicated (eg delay longer than time constant), then use MPC.
- This assumes skill level of typical system engineer in industry.
- If you have a PhD in Control and if you love LQG or H_∞ then you can use MPC without losing anything!

A little history of MPC

First industial developments: About 1970.

- Shell Texas: Cutler et al. (Proc. JACC 1980)
- Adersa France: Richalet. (Automatica 1978)

Patents:

- Martin-Sanchez (Spain), 1976.
- Prett, Ramaker, Cutler (Shell), 1982.

Academics:

- Propoi (1963)
- Kleinman (1970)
- Kwon and Pearson (1975)
- Rouhani and Mehra (1982)
- Clarke et al (1987)

Some terminology

Commercial and academic ‘brand’ names:

- Dynamic Matrix Control (DMC),
- Extended Prediction Self Adaptive Control (EPSAC),
- Generalised Predictive Control (GPC),
- Model Algorithmic Control (MAC),
- Predictive Functional Control (PFC),
- Quadratic Dynamic Matrix Control (QDMC),
- Sequential Open Loop Optimization (SOLO),

Generic names:

- Model Predictive Control (MPC)
- Model Based Predictive Control (MBPC)
- Receding Horizon Control (RHC)
- Advanced Process Control (APC)

Software for MPC

- MPC Toolbox for Matlab — not free!
- MultiParametric Toolbox (for Matlab), or *'MPT'*.
  `http://people.ee.ethz.ch/ mpt/3/`
- ACADO (KU Leuven)
  `http://acadotoolkit.org/`
- MPCtools (Lund University)
  Uses notation etc consistent with my book and lectures.
  `www.control.lth.se/Research/tools/mpctools.html`
- ICLOCS (Imperial College London)
  `www.ee.ic.ac.uk/ICLOCS`
- Write your own (good exercise!).

Summary of MPC

The essential components

- An internal model capable of simulating the plant behaviour faster than real time.
- A reference trajectory which defines the desired closed-loop behaviour — optional.
- The receding horizon principle.
- Characterising the assumed future input trajectory by a finite number of ‘moves’, or other parameters.
- On-line optimization, possibly constrained, to determine the future control strategy.

End of Lecture 1