MODEL PREDICTIVE CONTROL

CONCLUSIONS

Alberto Bemporad

imt.lu/ab

COURSE STRUCTURE

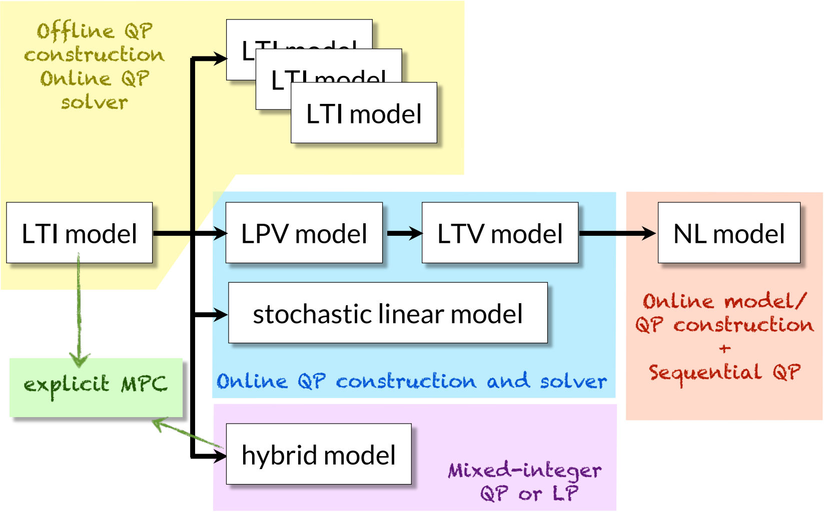
- **Basic concepts** of model predictive control (MPC) and **linear MPC**
- **Linear time-varying** and **nonlinear** MPC
- **Quadratic programming** (QP) and **explicit MPC**
- **Hybrid** MPC
- **Stochastic** MPC
- **Learning-based** MPC

Course page:

`http://cse.lab.imtlucca.it/~bemporad/mpc_course.html`

CONCLUSIONS

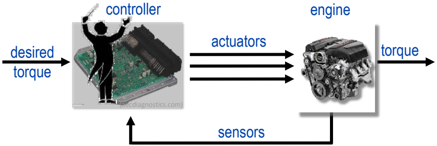
PREDICTION MODEL AND OPTIMIZATION PROBLEM



DO WE REALLY NEED ADVANCED CONTROL ?

Perspective of the automotive industry:

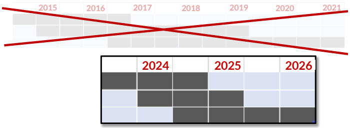
- Increasingly demanding **requirements** (emissions/consumption, passenger safety and comfort, …)



- Better control performance only achieved by better **coordination** of actuators:
  - **increasing number** of actuators (e.g., due to electrification)

  - take into account **limited range** of actuators

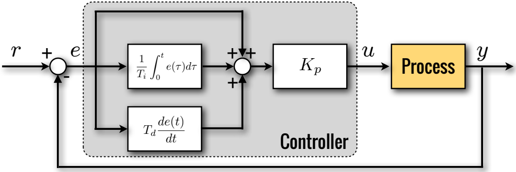
  - resilience in case of some **actuator failure**



- **Shorter development time** for control solution (market competition, changing legislation)

PROPORTIONAL INTEGRATIVE DERIVATIVE (PID) CONTROLLER

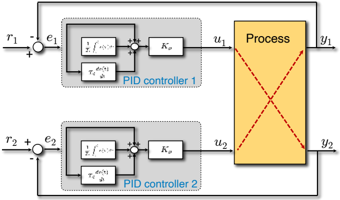
- PIDs are the most used controllers in industrial automation since the '30s



Pros:

- ✔ **Single-loops** are very easy to tune, just **3 parameters to calibrate**
- ✔ **Few lines of C code**, minimal memory and throughput requirements
- ✔ **No process model** required, just output measurements
- ✔ **Offset-free set-point tracking** thanks to integral action

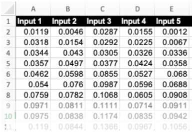
PROPORTIONAL INTEGRATIVE DERIVATIVE (PID) CONTROLLER



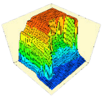
Cons: (1/2)

✖ **Multi-input/multi-output** systems: dynamical coupling requires tuning multiple PID loops together

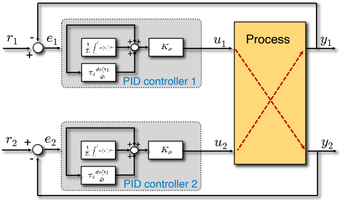
- ☹ Surgically changing a PID loop tuning may have bad consequences on other loops, due to dynamical **interactions**
- ☹ Lookup-table complexity increases **exponentially** (e.g.: 5 inputs, 10 values each $\rightarrow 10^5$ entries)
- ☹ Hard to coordinate multiple actuators **optimally**
- ☹ The calibration might need to be completely redone for a new model



	A	B	C	D	E
1	Input 1	Input 2	Input 3	Input 4	Input 5
2	0.0119	0.0046	0.0287	0.0155	0.0012
3	0.0318	0.0154	0.0292	0.0225	0.0067
4	0.0344	0.043	0.0305	0.0326	0.0336
5	0.0357	0.0497	0.0377	0.0424	0.0358
6	0.0462	0.0598	0.0855	0.0527	0.068
7	0.054	0.076	0.0987	0.0596	0.0688
8	0.0759	0.0782	0.1068	0.0605	0.0908
9	0.0971	0.0811	0.1111	0.0714	0.0911
10	0.0975	0.0838	0.1174	0.0835	0.0942
11	0.119	0.0844	0.1366	0.0967	0.1056



PROPORTIONAL INTEGRATIVE DERIVATIVE (PID) CONTROLLER

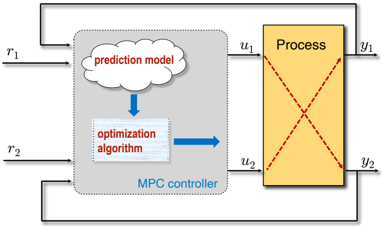


Cons: (2/2)

- ✖ Handling **input constraints** require additional **anti-windup** design
- ✖ **Output constraints** are much harder to handle
- ✖ Limited **preview** (derivative term =1st order extrapolation of future output)
- ✖ No explicit performance index optimized at runtime
- ✖ Resilience to **actuator faults** requires further design effort

Multivariable PID control design & calibration might be time consuming

MODEL PREDICTIVE CONTROL (MPC)

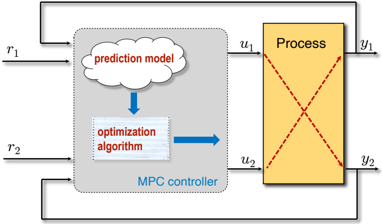


$$
\begin{aligned}
\min \quad & \sum_{k=0}^{N-1} \|y_k - r_{t+k}\|_2^2 + \rho \|u_k - u_{\mathrm{r},t+k}\|_2^2 \\
\text{s.t.} \quad & x_{k+1} = Ax_k + Bu_k \\
& y_k = Cx_k \\
& u_{\min} \le u_k \le u_{\max} \\
& y_{\min} \le y_k \le y_{\max}
\end{aligned}
$$

Pros:

- ✔ Naturally coordinates **multiple inputs and outputs**
- ✔ Naturally handles **input and output constraints**
- ✔ Very easily includes **preview** on references/measured disturbances
- ✔ **Performance index** optimized at runtime

MODEL PREDICTIVE CONTROL (MPC)

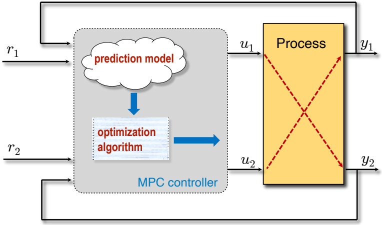


$$
\begin{aligned}
\min \quad & \sum_{k=0}^{N-1} \|y_k - r_{t+k}\|_2^2 + \rho \|u_k - u_{r,t+k}\|_2^2 \\
\text{s.t.} \quad & x_{k+1} = Ax_k + Bu_k \\
& y_k = Cx_k \\
& u_{\min} \le u_k \le u_{\max} \\
& y_{\min} \le y_k \le y_{\max}
\end{aligned}
$$

Pros:

- ✔ **Offset-free set-point tracking** thanks to **disturbance models** and observers
- ✔ Design easy to **transfer** to new models (**no lookup tables**)
- ✔ Controller easily reconfigurable online to **handle faults** (resilience)

MODEL PREDICTIVE CONTROL (MPC)



$$
\begin{aligned}
\min \quad & \sum_{k=0}^{N-1} \|y_k - r_{t+k}\|_2^2 + \rho\|u_k - u_{r,t+k}\|_2^2 \\
\text{s.t.} \quad & x_{k+1} = Ax_k + Bu_k \\
& y_k = Cx_k \\
& u_{\min} \leq u_k \leq u_{\max} \\
& y_{\min} \leq y_k \leq y_{\max}
\end{aligned}
$$

Cons:

- ✖ Multiple parameters to calibrate (models, weights, solver tolerances, ...)
- ✖ **Nontrivial C code** (QP solver), need to consider **memory** and **throughput** issues
- ✖ Requires a **process model** (physical modeling and/or system identification) (similar to all **model-based control-design** methods)

CONCLUSIONS

- MPC is a **universal control methodology**, same approach used for different
  - **models** (linear, nonlinear, hybrid, stochastic, ...)
  - **performance indices** (quadratic, convex, nonlinear, stochastic)
  - **constraints** (linear, nonlinear, robust, in probability)
- **MPC research**:
  1. Linear, uncertain, explicit, hybrid, nonlinear MPC: **mature theory**
  2. Stochastic MPC, economic MPC: **still open issues**
  3. Embedded optimization methods for MPC: **still room for many new ideas**
  4. System identification for MPC: there is **a lot to "learn"** from machine learning
  5. Data-driven MPC: still **a lot of open issues**
- **MPC technology**: rather mature, widely spread in many industrial sectors

General references on MPC

- D.Q. Mayne, "Model predictive control: Recent developments and future promise," Automatica, vol. 50, n.12, p. 2967-2986, 2014
- D.Q. Mayne, J.B. Rawlings, M.M. Diehl, "Model Predictive Control: Theory and Design," 2nd Ed., 2018
- A. Bemporad, M. Morari, and N. L. Ricker, Model Predictive Control Toolbox for Matlab – User's Guide, The Mathworks, Inc., 2004, `http://www.mathworks.com/access/helpdesk/help/toolbox/mpc/`
- F. Borrelli, A. Bemporad, M. Morari, "Predictive control for linear and hybrid systems," Cambridge University Press, 2017
- A. Bemporad, "Model-based predictive control design: New trends and tools," in Proc. 45th IEEE Conf. on Decision and Control, San Diego, CA, 2006

Hybrid systems

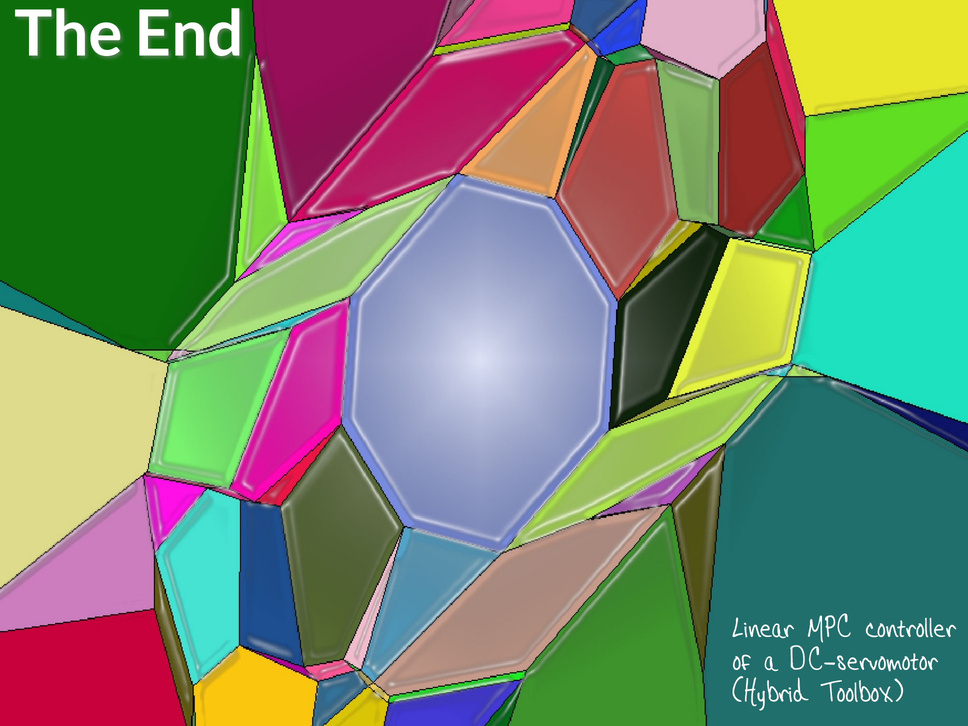
- A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," Automatica, 35(3), pp. 407-427, 1999
- F.D. Torrisi and A. Bemporad, "HYSDEL - A tool for generating computational hybrid models," IEEE Trans. Cont. Syst. Technology, 12(2), pp. 235-249, 2004
- A. Bemporad, "Hybrid Toolbox – User's Guide," Dec. 2003, `http://cse.lab.imtlucca.it/~bemporad/hybrid/toolbox`
- W.P.H.M Heemels, B. de Schutter, and A. Bemporad, "Equivalence of hybrid dynamical models," Automatica, 37(7), pp. 1085-1091, 2001
- A. Bemporad, G. Ferrari-Trecate, and M. Morari, "Observability and controllability of piecewise affine and hybrid systems," IEEE Trans. Autom. Cont., 45(10), pp. 1864-1876, 2000.

- V. Breschi, D. Piga, and A. Bemporad, "Piecewise affine regression via recursive multiple least squares and multicategory discrimination," Automatica, 73, pp. 155–162, 2016

Explicit MPC

- A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," Automatica, 38(1), pp. 3-20, 2002
- A. Bemporad, "A multiparametric quadratic programming algorithm with polyhedral computations based on nonnegative least squares," IEEE Trans. Autom. Cont., 60(11), pp. 2892–2903, 2015
- F. Borrelli, M. Baotic, A. Bemporad, and M. Morari, "Dynamic programming for constrained optimal control of discrete-time linear hybrid systems," Automatica, 41(10), 2005

The End



Linear MPC controller of a DC-servomotor (Hybrid Toolbox)