RECENT ADVANCES IN EMBEDDED MODEL PREDICTIVE CONTROL

Alberto Bemporad

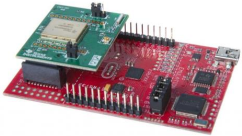
http://cse.lab.imtlucca.it/~bemporad

IMT SCHOOL FOR ADVANCED STUDIES LUCCA

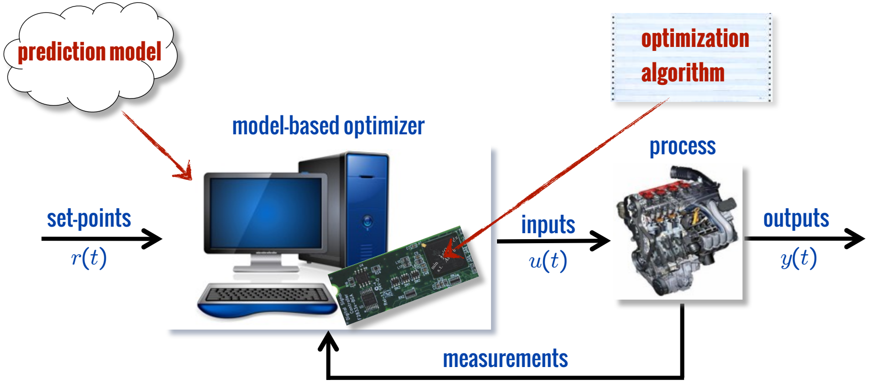
ICSTCC'18 - Sinaia, Romania - 11/10/2018

OUTLINE

- **Model Predictive Control** (MPC) (in a nutshell)
- Recent advances in **embedded quadratic programming (QP)** solvers
- **Data-driven** design of embedded MPC controllers
- Embedded MPC in **industry**



MODEL PREDICTIVE CONTROL (MPC)



Use a simplified dynamical **model** of the process to **predict** its likely future evolution and choose ~~the “best” control~~ a good action

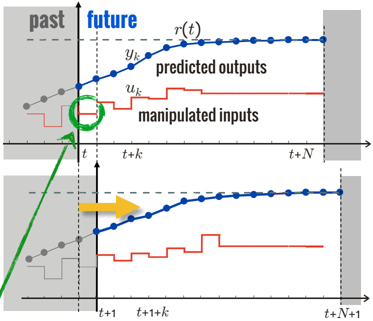
MODEL PREDICTIVE CONTROL (MPC)

- **Goal**: find the best control sequence over a future horizon of N steps

$$\min \sum_{k=0}^{N-1} \|W^y(y_k - r(t))\|_2^2 + \|W^u(u_k - u_r(t))\|_2^2$$

$$\begin{aligned}
\text{s.t.} \quad & x_{k+1} = f(x_k, u_k) && \text{prediction model}\\
& y_k = g(x_k, u_k) \\
& u_{\min} \le u_k \le u_{\max} && \text{constraints}\\
& y_{\min} \le y_k \le y_{\max} \\
& x_0 = x(t) && \text{state feedback}
\end{aligned}$$

optimization problem



- **At each time** t:
  - get new measurements to update the estimate of the current state $x(t)$
  - solve the optimization problem with respect to $\{u_0, \dots, u_{N-1}\}$
  - apply only the first optimal move $u(t) = u_0^*$, discard the remaining samples

MPC IN INDUSTRY

- The MPC concept for process control dates back to the 60's

Discrete Dynamic Optimization
Applied to On-Line Optimal Control

MARSHALL D. RAFAL and WILLIAM F. STEVENS

(Rafal, Stevens, AiChE Journal, 1968)

- MPC used in the process industries since the 80's

(Qin, Badgewell, 2003) (Bauer, Craig, 2008)

MPC is the standard for advanced control in the process industry

- Research in MPC is still very active !

MPC IN INDUSTRY

(Samad, IEEE CS Magazine, 2017)

- Impact of advanced control technologies in industry

TABLE 1 A list of the survey results in order of industry impact as perceived by the committee members.

Rank and Technology	High-Impact Ratings	Low- or No-Impact Ratings
PID control	100%	0%
Model predictive control	78%	9%
System identification	61%	9%
Process data analytics	61%	17%
Soft sensing	52%	22%
Fault detection and identification	50%	18%
Decentralized and/or coordinated control	48%	30%
Intelligent control	35%	30%
Discrete-event systems	23%	32%
Nonlinear control	22%	35%
Adaptive control	17%	43%
Robust control	13%	43%
Hybrid dynamical systems	13%	43%

AUTOMOTIVE APPLICATIONS OF MPC

(Bemporad, Bernardini, Borrelli, Cimini, Di Cairano, Esen, Giorgetti, Graf-Plessen, Hrovat, Kolmanovsky Levijoki, Ripaccioli, Trimboli, Tseng, Yanakiev, ... (2001-present))

Powertrain
engine control, magnetic actuators, robotized gearbox, power MGT in HEVs, cabin heat control, electrical motors

Vehicle dynamics
traction control, active steering, semiactive suspensions, autonomous driving

Ford Motor Company

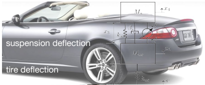
Jaguar

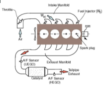
DENSO Automotive

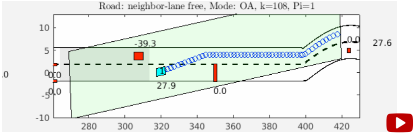
FCA

General Motors





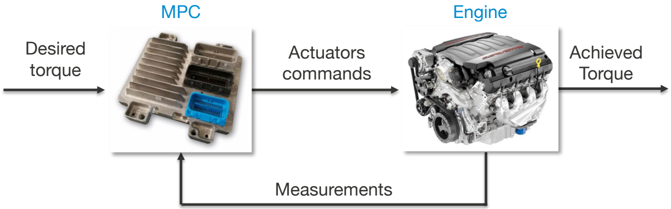
Most automotive OEMs are looking into MPC solutions today

MPC FOR AUTONOMOUS DRIVING

- Coordinate **torque request** and **steering** to achieve **safe and comfortable** autonomous driving with no collisions
- MPC combines **path planning**, **path tracking**, and **obstacle avoidance**
- **Stochastic prediction models** are used to account for uncertainty and driver's behavior

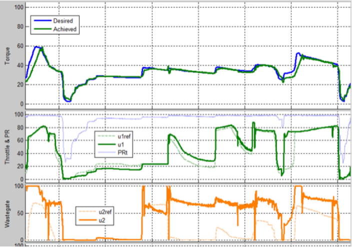
MPC OF GASOLINE TURBOCHARGED ENGINES

- Optimize **engine actuators** (throttle, wastegate, intake/exhaust cams) to make **engine torque** track set-points, maximizing efficiency and satisfying **constraints**



QP solved in real-time on ECU

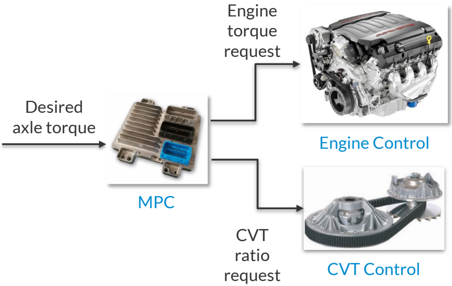
(Bemporad, Bernardini, Long, Verdejo, 2018)

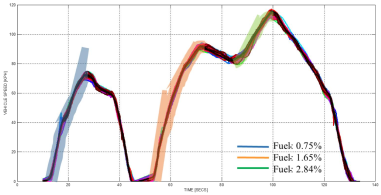


engine operating at low pressure (66 kPa)

SUPERVISORY MPC OF POWERTRAIN WITH CVT

- Coordinate **engine torque request** and continuously variable transmission (CVT) **ratio** to improve fuel economy and drivability
- Real-time MPC is able to take into account **coupled dynamics** and **constraints**, optimizing performance also during transients



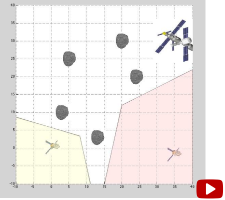


US06 Double Hill driving cycle

(Bemporad, Bernardini, Livshiz, Pattipati, 2018)

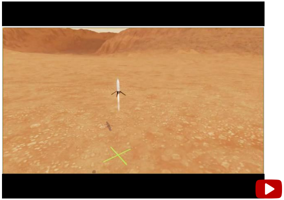
- MPC capabilities explored in new space applications

cooperating UAVs



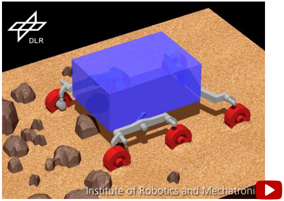
(Bemporad, Rocchi, 2011)

powered descent



(Pascucci, Bennani, Bemporad, 2016)

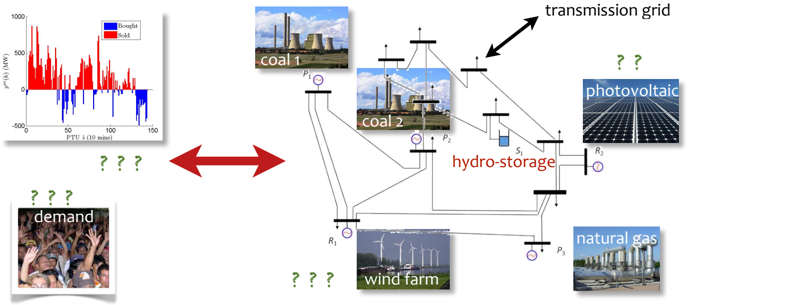
planetary rover



(Krenn et. al., 2012)

MPC FOR SMART ELECTRICITY GRIDS

(Patrinos, Trimboli, Bemporad, 2011)



Dispatch power in smart distribution grids, trade energy on energy markets

Challenges: account for **dynamics**, network **topology**, physical **constraints**, and **stochasticity** (of renewable energy, demand, electricity prices)

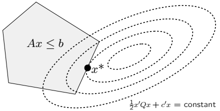
FP7-ICT project "E-PRICE - Price-based Control of Electrical Power Systems" (2010-2013)

EMBEDDED LINEAR MPC AND QUADRATIC PROGRAMMING

- Linear MPC requires solving a **Quadratic Program (QP)**

$$\begin{array}{ll} \min\limits_{z} & \frac{1}{2}z'Qz + x'(t)F'z \\ \text{s.t.} & Gz \leq W + Sx(t) \end{array}$$

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



ON MINIMIZING A CONVEX FUNCTION SUBJECT TO LINEAR INEQUALITIES

By E. M. L. BEALE

Admiralty Research Laboratory, Teddington, Middlesex

SUMMARY

THE minimization of a convex function of variables subject to linear inequalities is discussed briefly in general terms. Dantzig's Simplex Method is extended to yield finite algorithms for minimizing either a convex quadratic function or the sum of the t largest of a set of linear functions, and the solution of a generalization of the latter problem is indicated. In the last two sections a form of linear programming with random variables as coefficients is described, and shown to involve the minimization of a convex function.

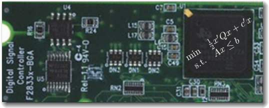
(Beale, 1955)

A rich set of good QP algorithms is available today

- Which QP algorithms are suitable for implementation in **embedded systems** ?

MPC IN A PRODUCTION ENVIRONMENT

Key requirements for deploying MPC in production:



1. **speed (throughput)**
   - **worst-case** execution time less than sampling interval
   - also fast on **average** (to free the processor to execute other tasks)

2. limited **memory and CPU power** (e.g., 150 MHz / 50 kB)

3. **numerical robustness** (single precision arithmetic)

4. **certification** of worst-case execution time

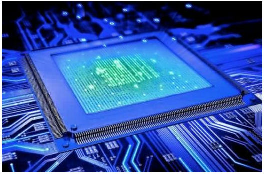
5. **code simple enough** to be validated/verified/certified
   (library-free C code, easy to check by production engineers)

EMBEDDED SOLVERS IN INDUSTRIAL PRODUCTION

- Multivariable MPC controller
- Sampling frequency = 40 Hz (= 1 QP solved every 25 ms)
- Vehicle operating $\approx$1 hr/day for $\approx$360 days/year on average
- Controller running on 10 million vehicles

~520,000,000,000,000 QP/yr

and none of them should fail.



FAST GRADIENT PROJECTION

(Nesterov, 1983) (Patrinos, Bemporad, 2014)

- Solve (dual) QP by **fast gradient method**

$$\begin{array}{ll} \min\limits_z & \frac{1}{2}z'Qz + x'F'z \\ \text{s.t.} & Gz \leq W + Sx \end{array}$$

$$\begin{array}{lcl} K & = & Q^{-1}G' \\ g & = & Q^{-1}Fx \\ L & \geq & \frac{1}{\lambda_{\max}(GQ^{-1}G')} \\ \beta_k & = & \max\{\frac{k-1}{k+2}, 0\} \end{array}$$

$$\begin{array}{rcl} w^k & = & y^k + \beta_k(y^k - y^{k-1}) \\ z^k & = & -Kw^k - g \\ s^k & = & \frac{1}{L}Gz^k - \frac{1}{L}(W + Sx) \\ y^{k+1} & = & \max\left\{w^k + s^k, 0\right\} \end{array}$$

```
while k<maxiter
  beta=max((k-1)/(k+2),0);
  w=y+beta*(y-y0);
  z=-(iMG*w+iMc);
  s=GL*z-bL;

  y0=y;

  % Termination
  if all(s<=epsGL)
    gapL=-w'*s;
    if gapL<=epsVL
      return
    end
  end

  y=w+s;
  k=k+1;
end
```

- Very **simple to code**

- Convergence rate: $f(x^k) - f(x^*) \leq \frac{2L}{(k+2)^2}\|z_0 - z^*\|_2^2$
  (Necoara, Nesterov, Glineur, 2018)

- Tight bounds on maximum number of iterations

- Extended to **mixed-integer quadratic programming (MIQP)** (Naik, Bemporad, 2017)

ADMM

(Gabay, Mercier, 1976) (Glowinski, Marrocco, 1975) (Douglas, Rachford, 1956) (Boyd et al., 2010)

- Alternating Directions Method of Multipliers for QP

$$\begin{array}{ll} \min & \frac{1}{2}z'Qz + c'z \\ \text{s.t.} & \ell \leq Az \leq u \end{array}$$

$$\begin{aligned} z^{k+1} &= -(Q + \rho A'A)^{-1}(\rho A'(u^k - s^k) + c) \\ s^{k+1} &= \min\{\max\{Az^{k+1} + u^k, \ell\}, u\} \\ u^{k+1} &= u^k + Ax^{k+1} - s^{k+1} \end{aligned}$$

```
while k<maxiter
  k=k+1;
  z=-iM*(c+A'*(rho*(u-s)));
  Az=A*z;
  s=max(min(Az+u,ub),lb);
  u=u+Az-s;
end
```

(7 lines EML code)
(≈40 lines of C code)

ρu = dual vector

- Matrix $(Q + \rho A'A)$ must be nonsingular
- The factorization of matrix $(Q + \rho A'A)$ can be done at start and cached
- Very **simple to code**. Sensitive to matrix **scaling** (as gradient projection)
- Used in many applications (control, signal processing, machine learning)

REGULARIZED ADMM FOR QUADRATIC PROGRAMMING

(Banjac, Stellato, Moehle, Goulart, Bemporad, Boyd, 2017)

- Robust "regularized" ADMM iterations:

$$\begin{aligned} z^{k+1} &= -(Q+\rho A^T A+\epsilon I)^{-1}(c-\epsilon z_k+\rho A^T(u^k-z^k)) \\ s^{k+1} &= \min\{\max\{Az^{k+1}+y^k,\ell\},u\} \\ u^{k+1} &= u^k+Az^{k+1}-s^{k+1} \end{aligned}$$

- Works for any $Q \succeq 0$, A, and choice of $\epsilon > 0$
- **Simple** to code, **fast**, and **robust**
- Only needs to factorize $\begin{bmatrix} Q+\epsilon I & A' \\ A & -\frac{1}{\rho}I \end{bmatrix}$ once
- Implemented in free **osQP solver**

  (Python interface: $\approx$ 20,000 downloads)

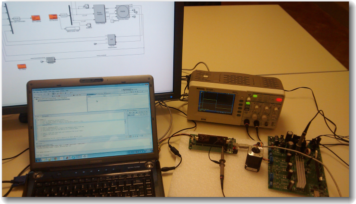
  `http://osqp.org`

- Extended to solve **mixed-integer quadratic programming** problems

  (Stellato, Naik, Bemporad, Goulart, Boyd, 2018)

QP SOLVERS - AN EXPERIMENTAL COMPARISON

- **Experimental setup**:
  - PC with MATLAB/Simulink
  - RS232 adapter
  - TMS320F28335 DSP (**150 MHz**)



vars × constr.	ODYS QP	GPAD	ADMM
4× 16	0.12 ms	0.33 ms	1.4 ms
8× 24	0.44 ms	1.1 ms	4 ms
12×32	1.2 ms	2.6 ms	8.2 ms

- Active set (AS) methods are usually the best on small/medium problems:
  - excellent quality solutions within few iterations
  - behavior is more predictable (=less sensitive to preconditioning)
  - no need for advanced linear algebra libraries

CAN WE SOLVE QP'S USING LEAST SQUARES ?

The **least squares** (LS) problem is probably the most studied problem in numerical linear algebra

Adrien-Marie Legendre (1752–1833)

Carl Friedrich Gauss (1777–1855)

$$v = \arg\min \|Av - b\|_2^2$$

In MATLAB: `>> v=A\b` (one character !)

- Nonnegative Least Squares (NNLS): (Lawson, Hanson, 1974)

$$\begin{array}{rl} \min_v & \|Av - b\|_2^2 \\ \text{s.t.} & v \geq 0 \end{array}$$

very simple to solve (750 chars in Embedded MATLAB)

SOLVING QP'S VIA NONNEGATIVE LEAST SQUARES

(Bemporad, 2016)

- Complete the squares and transform QP to **least distance problem** (LDP)

$$\begin{array}{ll}\min_z & \frac{1}{2}z'Qz + c'z \\ \text{s.t.} & Gz \leq g\end{array} \quad \xrightarrow[u \triangleq Lz + L^{-T}c]{Q = L'L} \quad \begin{array}{ll}\min_u & \frac{1}{2}\|u\|^2 \\ \text{s.t.} & Mu \leq d\end{array}$$

- An LDP can be solved by the NNLS (Lawson, Hanson, 1974)

$$\begin{array}{ll}\min_y & \frac{1}{2}\left\|\begin{bmatrix} M' \\ d' \end{bmatrix} y + \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right\|_2^2 \\ \text{s.t.} & y \geq 0\end{array}$$

$$\begin{array}{rcl} M & = & GL^{-1} \\ d & = & b + GQ^{-1}c \end{array}$$

- If residual $= 0$ then QP is infeasible. Otherwise set

$$z^* = -\frac{1}{1 + d'y^*}L^{-1}M'y^* - Q^{-1}c$$

- Extended to solving mixed-integer QP's (Bemporad, NMPC, 2015)

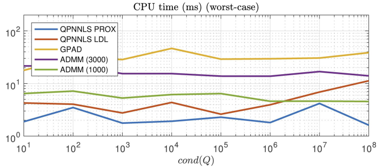
SOLVING QP'S VIA NNLS AND PROXIMAL POINT ITERATIONS

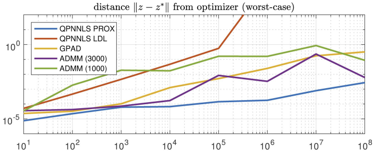
(Bemporad, 2018)

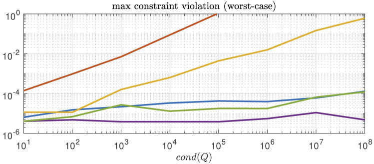
- Solve QP via NNLS within **proximal-point iterations**

$$
\begin{array}{rl}
z_{k+1} = \arg\min_z & \frac{1}{2}z'Qz + c'z + \frac{\epsilon}{2}\|z - z_k\|_2^2 \\
\text{s.t.} & Az \leq b \\
 & Gx = g
\end{array}
$$

- Advantage: **numerical robustness**, as $Q + \epsilon I$ can be arbitrarily well conditioned







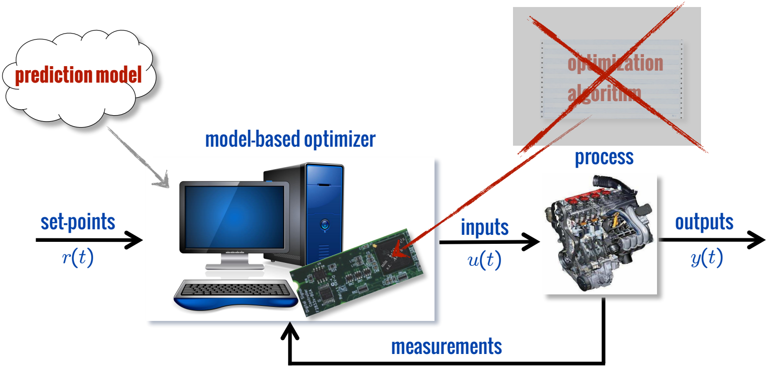
single precision arithmetic

30 vars, 100 constraints

(Macbook Pro 3 GHz Intel Core i7)

- Extended to solve **MIQP** problems (Naik, Bemporad, 2018)

MPC WITHOUT ON-LINE QP



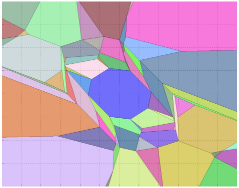
- Can we implement MPC **without an on-line solver** ?

EXPLICIT MODEL PREDICTIVE CONTROL AND MULTIPARAMETRIC QP

(Bemporad, Morari, Dua, Pistikopoulos, 2002)

- The multiparametric solution of a strictly convex QP is **continuous** and **piecewise affine**

$$\begin{array}{rl} z^*(x) = \arg\min_z & \frac{1}{2}z'Qz + x'F'z \\ \text{s.t.} & Gz \leq W + Sx \end{array}$$



- Corollary: **linear MPC is continuous & piecewise affine** !

$$z^* = \begin{bmatrix} \mathbf{u_0} \\ u_1 \\ \vdots \\ u^*_{N-1} \end{bmatrix}$$

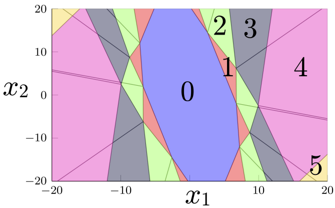
$$u_0^*(x) = \begin{cases} F_1 x + g_1 & \text{if} \quad H_1 x \leq K_1 \\ \vdots & \vdots \\ F_M x + g_M & \text{if} \quad H_M x \leq K_M \end{cases}$$

- New mpQP solver based on NNLS available (Bemporad, 2015) and included in **MPC Toolbox** since R2014b (Bemporad, Morari, Ricker, 1998-today)

Is explicit MPC better than on-line QP (=implicit MPC) ?

COMPLEXITY CERTIFICATION FOR ACTIVE-SET QP SOLVERS

- **Result**: The **number of iterations** to solve the QP via a dual active-set method is a **piecewise constant function** of the parameter x



(Cimini, Bemporad, 2017)

We can **exactly** quantify how many iterations (flops) the QP solver takes in the worst-case !

- Examples (from MPC Toolbox):

	inverted pendulum	DC motor	nonlinear demo	AFTI F16
Explicit MPC				
max flops	3382	1689	9184	16434
max memory (kB)	55	30	297	430
Implicit MPC				
max flops	3809	2082	7747	7807
sqrt	27	9	37	33
max memory (kB)	15	13	20	16

- QP certification algorithm currently used in production

ODYS Advanced Controls & Optimization

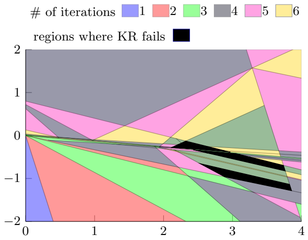
CERTIFICATION OF KR SOLVER

- The **KR algorithm** is a **very simple** and **effective** solver for **box-constrained QP**. All violated/active constraints form the new active set at the next iteration

  (Kunisch, Rendl, 2003) (Hungerländer, Rendl, 2015)
- Assumptions for convergence are quite conservative, and indeed **KR can cycle**

> We can **exactly** map how many iterations KR takes to converge (or cycle)

(Cimini, Bemporad, 2019)



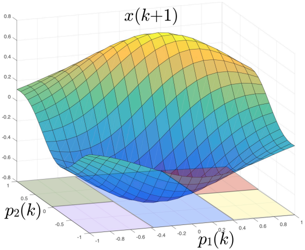
	Example 1	Example 2	Example 3
Explicit MPC			
max flops	324	1830	5401
max memory [kB]	3.97	15.9	89.69
Dual active-set			
max flops + sqrt	580 + 5	1922+ 13	3622+ 24
max memory [kB]	8.21	8.63	8.90
KR algorithm			
max flops	489	1454	2961
max memory [kB]	3.19	3.39	3.51

DATA-DRIVEN MPC

MPC AND MACHINE LEARNING

- **Model** predictive control requires a **model** of the process
- Models are usually obtained **from data** (parameter estimation or black-box modeling)

In industrial MPC most effort is spent in **identifying open-loop process models**

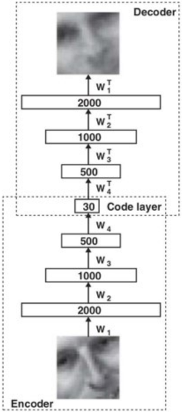


- Many techniques and tools available from **systems identification** and **machine learning** literature
- Chosen model structure must be tailored to MPC design and optimization (linear/switching liner/nonlinear)

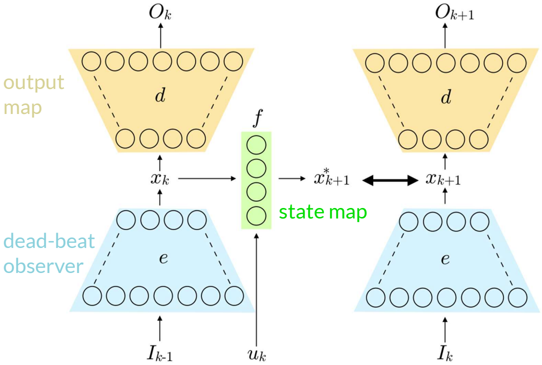
LEARNING NONLINEAR MODELS FOR MPC

(Masti, Bemporad, CDC 2018)

- **Idea**: use **autoencoders** and artificial neural networks to learn a **nonlinear state-space model** of **desired order** from input/output data



(Hinton, Salakhutdinov, 2006)

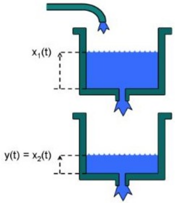


$$
\begin{aligned}
O_k &= [y_k' \ \dots \ y_{k-m}']' \\
I_k &= [y_k' \ \dots \ y_{k-n_a+1}' \ u_k' \ \dots \ u_{k-n_b+1}']'
\end{aligned}
$$

LEARNING NONLINEAR MODELS FOR MPC - AN EXAMPLE

(Masti, Bemporad, CDC 2018)

- System generating the data = nonlinear 2-tank benchmark

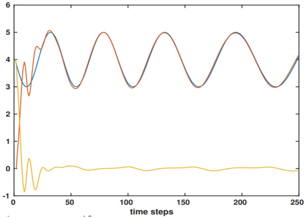


www.mathworks.com

$$\begin{cases} x_1(k+1) = x_1(k) - k_1\sqrt{x_1(k)} + k_2(u(k) + w(k)) \\ x_2(k+1) = x_2(k) + k_3\sqrt{x_1(k)} - k_4\sqrt{x_2(k)} \\ y(k) = x_2(k) + v(k) \end{cases}$$

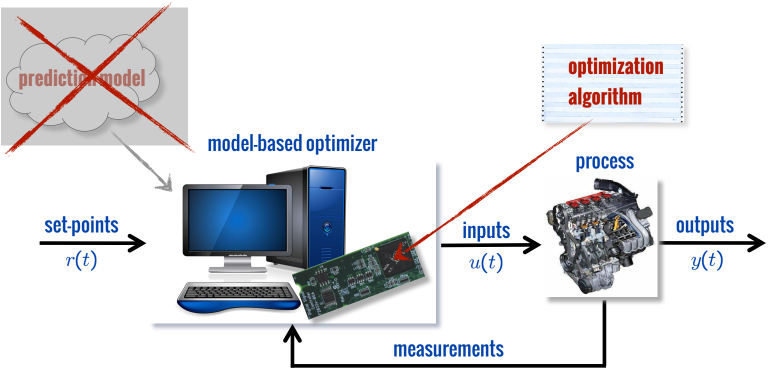
Model is totally unknown to learning algorithm

- Artificial neural network (ANN): 3 hidden layers 60 exponential linear unit (ELU) neurons
- For given number of model parameters, **autoencoder approach is superior to NNARX**
- **Jacobians** directly obtained from ANN structure for Kalman filtering & MPC problem construction



LTV-MPC results

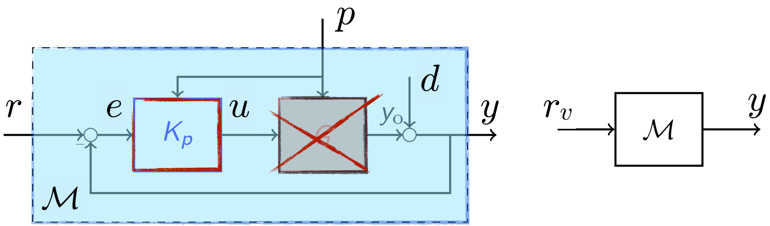
DATA-DRIVEN MPC



- Can we design an MPC controller **without** first identifying a model of the **open-loop process** ?

DATA-DRIVEN DIRECT CONTROLLER SYNTHESIS

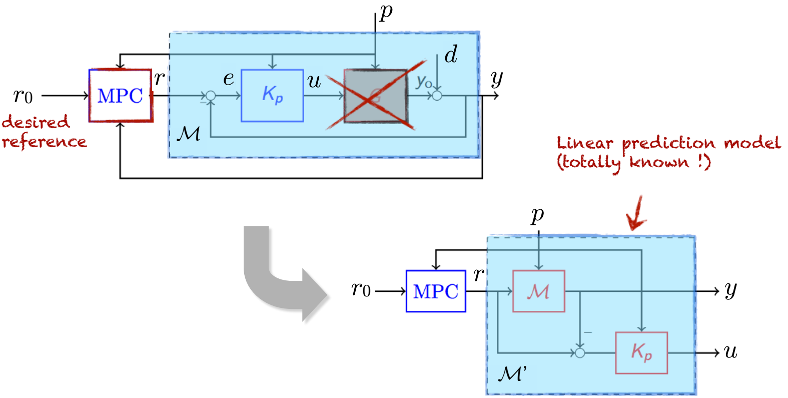
(Campi, Lecchini, Savaresi, 2002) (Formentin et al., 2015)



- Collect a set of **data** $\{u(t), y(t), p(t)\}, t = 1, \ldots, N$
- Specify a **desired closed-loop linear model** $\mathcal{M}$ from r to y
- Compute $r_v(t) = \mathcal{M}^{\#} y(t)$ from **pseudo-inverse model** $\mathcal{M}^{\#}$ of $\mathcal{M}$
- **Identify** linear (LPV) model K_p from $e_v = r_v - y$ (virtual tracking error) to u

- Design a linear MPC (**reference governor**) to generate the reference r

  (Bemporad, Mosca, 1994) (Gilbert, Kolmanovsky, Tan, 1994)

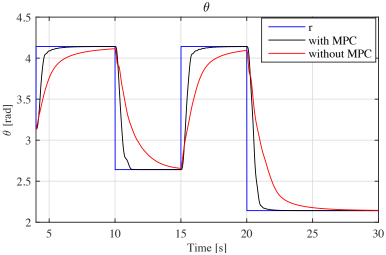


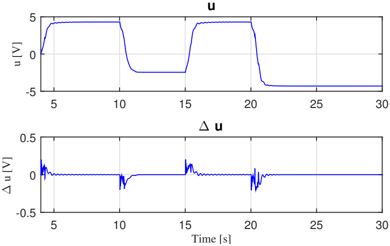
- MPC designed to handle input/output **constraints** and improve **performance**

  (Piga, Formentin, Bemporad, 2017)

DATA-DRIVEN MPC - AN EXAMPLE

- Experimental results: MPC handles soft constraints on u, Δu and y





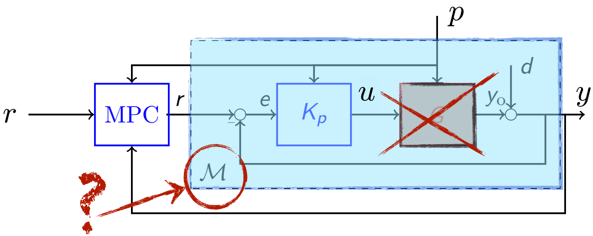
desired tracking performance achieved

constraints on input increments satisfied

No open-loop process model is identified to design the MPC controller!

OPTIMAL DATA-DRIVEN MPC

- **Question**: How to choose the reference model $\mathcal{M}$?



- Can we choose $\mathcal{M}$ from data so that K_p is an **optimal controller** ?

OPTIMAL DATA-DRIVEN MPC

(Selvi, Piga, Bemporad, 2018)

- **Idea**: parameterize desired closed-loop model $\mathcal{M}(\theta)$ and optimize

$$\min_{\theta} J(\theta) = \frac{1}{N}\sum_{t=0}^{N-1} \underbrace{W_y(r(t)-y_p(\theta,t))^2 + W_{\Delta u}\Delta u_p^2(\theta,t)}_{\text{performance index}} + \underbrace{W_{\text{fit}}(u(t)-u_v(\theta,t))^2}_{\text{identification error}}$$

- Evaluating $J(\theta)$ requires synthesizing $K_p(\theta)$ from data and simulating the nominal model and control law

$$y_p(\theta,t) = \mathcal{M}(\theta)r(t) \qquad u_p(\theta,t) = K_p(\theta)(r(t)-y_p(\theta,t))$$

$$\Delta u_p(\theta,t) = u_p(\theta,t) - u_p(\theta,t-1)$$

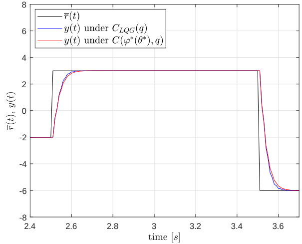
- Optimal θ obtained by solving a **(non-convex) nonlinear programming** problem

(Selvi, Piga, Bemporad, 2018)

- **Results**: linear process

$$G(z) = \frac{z - 0.4}{z^2 + 0.15z - 0.325}$$

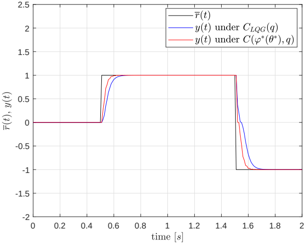
The data-driven controller is **only 1.3% worse** than model-based LQR



- **Results**: nonlinear (Wiener) process

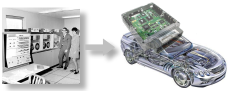
$$\begin{aligned} y_L(t) &= G(z)u(t) \\ y(t) &= |y_L(t)| \arctan(y_L(t)) \end{aligned}$$

The data-driven controller is **24% better** than LQR based on identified open-loop model !



CONCLUSIONS

- Long history of success of MPC in the **process industries**, now spreading to **automotive** (and many others)



- **Key enablers** for MPC to be successful in industry:

    1. Fast, robust, and simple to code **QP solvers**, with proved execution time
    2. Good **system identification / machine learning** methods to deal with data
    3. **Production managers** that are willing to adopt new advanced control technologies

Is MPC a mature technology for the automotive industry ?

MPC GOES TO AUTOMOTIVE PRODUCTION NOW !

General Motors and ODYS have developed a multivariable constrained MPC system for torque tracking in turbocharged gasoline engines. The control system is scheduled for production by GM in fall 2018.

- Multivariable system, 4 inputs, 4 outputs. QP solved in real time on ECU
  (Bemporad, Bernardini, Long, Verdejo, 2018)

- Supervisory MPC for powertrain control also in production in 2018
  (Bemporad, Bernardini, Livshiz, Pattipati, 2018)

First known mass production of MPC in the automotive industry

http://www.odys.it/odys-and-gm-bring-online-mpc-to-production

ODYS
Advanced Controls & Optimization