EL2520
Control Theory and Practice

Lecture 12:
Model predictive control

Elling W. Jacobsen
School of Electrical Engineering and Computer Science
KTH, Stockholm, Sweden

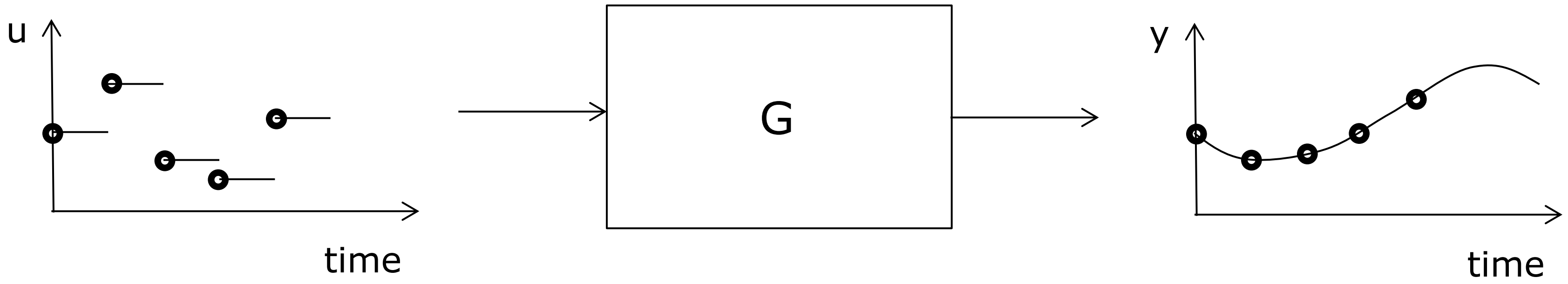
Background

- Models have predictive power, i.e., may be used to predict future
- Idea: optimize future behavior using control input
- Main problem: model uncertain and future disturbances unknown $\Rightarrow$ introduce feedback by regularly updating model states based on measurements and then repeating the optimization
- Method is known as **receding horizon control**, or more commonly as **Model Predictive Control (MPC)**
- A key point is that hard constraints can be included in the optimization
- MPC is based on discrete time models

Outline

- Sampling and discrete time systems
- The Finite horizon LQR problem (blackboard)
- Adding constraints: the MPC controller (blackboard)
- Comments on tuning and an example

Computer-controlled systems



- Input to continuous time system G changed at discrete times, kept constant between time instants
- Continuous output sampled every h seconds

How does state evolve between sampling instances?

Plant dynamics at sampling instants

Recall that

$$\dot{x}(t) = Ax(t) + Bu(t) \Rightarrow x(t+h) = e^{Ah}x(t) + \int_{s=0}^{h} e^{As}Bu(s)\,ds$$

so if u is held constant during sample interval $u(t) = u_t,\ t \in [t, t+h)$

$$x(t+h) = A_D x(t) + B_D u_t \qquad \left(A_D = e^{Ah},\ B_D = \int_{s=0}^{h} e^{As}B\,ds \right)$$

$$y(t) = Cx(t) + Du_t$$

A discrete-time linear system!

Discrete-time linear systems

For notational convenience, we drop reference to physical time and write

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{aligned}$$

where

- $\{u_0, u_1, \ldots\}$ is an **input sequence**
- $\{y_0, y_1, \ldots\}$ is the **output sequence**
- $\{x_0, x_1, \ldots\}$ is the **state evolution**

System is stable if all eigenvalues of A are less than one in magnitude

Discrete-time linear systems

Some system theory for discrete-time linear systems (Book Ch. 2.6, 3.7, 4)

System is controllable if $S(A,B) = \begin{bmatrix} B & AB & A^2B & \ldots & A^{n-1}B \end{bmatrix}$ is full rank.

System is observable if

$$O(A,C) = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

has full rank

Observer-based controllers have the form

$$\begin{aligned} \hat{x}_{t+1} &= A\hat{x}_t + Bu_t + K(y_t - C\hat{x}_t) \\ u_t &= -L\hat{x}_t \end{aligned}$$

Finite-horizon LQR problem

Find control sequence

$$U = \{u_0, \ldots, u_{N-1}\}$$

that minimizes the quadratic cost function

$$J(U) = \sum_{k=0}^{N-1} (x_k^T Q_1 x_k + u_k^T Q_2 u_k) + x_N^T Q_f x_N$$

for given state cost, control cost, and final cost matrices

$$Q_1 = Q_1^T \geq 0, \quad Q_2 = Q_2^T > 0, \quad Q_f = Q_f^T \geq 0,$$

N is called the **horizon** of the problem.

Note the final state cost: mainly used to ensure stability

Finite-time LQR via least-squares

Note that $X = (x_0, \ldots, x_N)$ is a linear function of x_0 and $U = (u_0, \ldots, u_{N-1})$

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ B & 0 & \cdots & 0 \\ AB & B & 0 & \cdots \\ \vdots & \vdots & & \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix} + \begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} x_0$$

Can express as

$$X = GU + Hx_0$$

where $G \in \mathbb{R}^{Nn \times Nm}$, $H \in \mathbb{R}^{Nn \times n}$

Finite-time LQR via least-squares

Can express finite-horizon cost as

$$J(U) = X^T \underbrace{\begin{bmatrix} Q_1 & 0 & \cdots & 0 \\ 0 & \ddots & 0 & \vdots \\ \vdots & 0 & Q_1 & 0 \\ 0 & \cdots & 0 & Q_f \end{bmatrix}}_{\overline{Q}_1} X + U^T \underbrace{\begin{bmatrix} Q_2 & 0 & \cdots & 0 \\ 0 & \ddots & 0 & \vdots \\ \vdots & 0 & Q_2 & 0 \\ 0 & \cdots & 0 & Q_2 \end{bmatrix}}_{\overline{Q}_2} U =$$

$$= (GU + Hx_0)^T \overline{Q}_1 (GU + Hx_0) + U^T \overline{Q}_2 U =$$

$$= U^T (G^T \overline{Q}_1 G + \overline{Q}_2) U + 2x_0^T H^T \overline{Q}_1 G U + x_0^T H^T \overline{Q}_1 H x_0 =$$

$$:= U^T P_{LQ} U + 2q_{LQ}^T U + r_{LQ}$$

so optimal control is

$$U^\star = -P_{LQ}^{-1} q_{LQ}$$

for which

$$J(U^\star) = r_{LQ} - q_{LQ}^T P_{LQ}^{-1} q_{LQ}$$

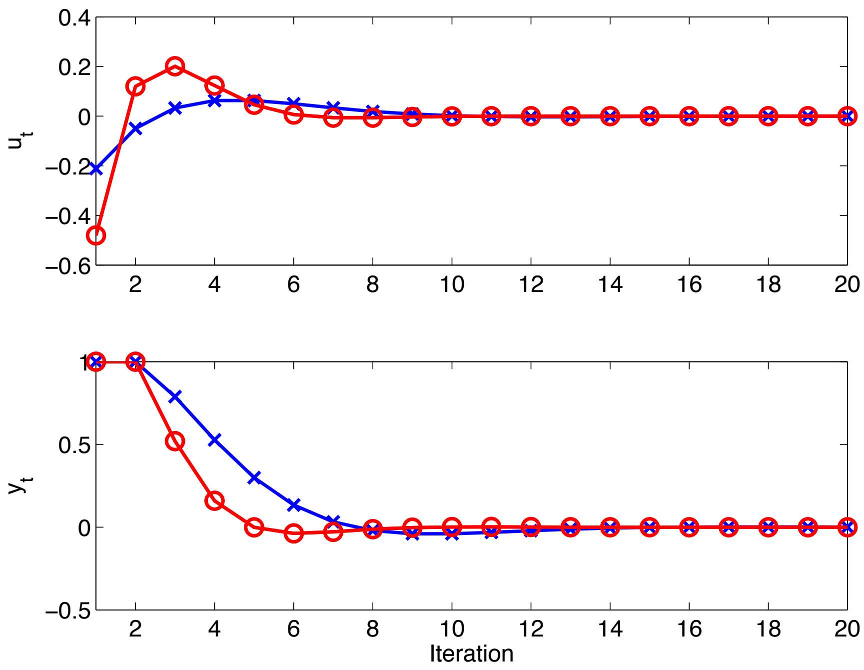
Example

LQR problem for system

$$x_{k+1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_t + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_t, \qquad C = \begin{bmatrix} 1 & 0 \end{bmatrix} x_t$$

$$Q_1 = Q_f = C^T C, \;\; Q_2 = \rho$$

with horizon length 20. Results for $\rho = 10$ (blue) and $\rho = 1$ (red)



Constrained Predictive Control

Finite-horizon LQR with hard constraints on u and y:

$$\begin{array}{ll} \text{minimize} & J(U) = \sum_{k=0}^{N-1} (x_k^T Q_1 x_k + u_k^T Q_2 u_k) + x_N^T Q_f x_N \\ \text{subject to} & u_{\min} \leq u_k \quad \leq u_{\max}, \ \ k = 0, \ldots, N-1 \\ & y_{\min} \leq C x_k \leq y_{\max}, \ \ k = 1, \ldots, N \\ & x_{k+1} = A x_k + B u_k \end{array}$$

Can be simplified by eliminating $\{x_1, \ldots, x_N\}$ (as above)

- results in a quadratic programming problem in $\{u_0, \ldots, u_{N-1}\}$

Quadratic programming (QP)

Minimizing a quadratic objective function subject to linear constraints

$$\begin{array}{ll} \text{minimize} & u^T P u + 2q^T u + r \\ \text{subject to} & Au \leq b \end{array}$$

Any u satisfying $Au \leq b$ is said to be **feasible.**

- clearly, not all quadratic programs are feasible (depends on A, b; more about this later…)

"Easy" to solve when objective function is **convex** (P positive semidefinite)

- optimal solution found in polynomial time
- commercial solvers deal with 10,000's of variables in a few seconds

Quadratic programming tricks

Example. The double inequality $u_{\min} \leq u \leq u_{\max}$ can be written as

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix} u \leq \begin{bmatrix} u_{\max} \\ -u_{\min} \end{bmatrix}$$

Example. The equality $u = u_{\mathrm{tgt}}$ can be written as $u_{\mathrm{tgt}} \leq u \leq u_{\mathrm{tgt}}$, hence

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix} u \leq \begin{bmatrix} u_{\mathrm{tgt}} \\ -u_{\mathrm{tgt}} \end{bmatrix}$$

Constrained control via QP

$$\begin{array}{ll} \text{minimize} & J(U) = \sum_{k=0}^{N-1} (x_k^T Q_1 x_k + u_k^T Q_2 u_k) + x_N^T Q_f x_N \\ \text{subject to} & u_{\min} \le u_k \quad \le u_{\max}, \quad k = 0, \ldots, N-1 \\ & y_{\min} \le C x_k \le y_{\max}, \quad k = 1, \ldots, N \\ & x_{k+1} = A x_k + B u_k \end{array}$$

As above, introducing $X = (x_0, \ldots, x_N)$, $U = (u_0, \ldots, u_{N-1})$,

$$X = GU + Hx_0$$

and the objective function can be written as

$$J(U) = U^T P_{LQ} U + 2 q_{LQ}^T U + r_{LQ}$$

Convex if $Q_2 \succ 0$ (implies that P_{LQ} is positive semi-definite)

What about the constraints?

Predictive control with constraints

Similarly, the constraints $y_{\min} \leq y_k \leq y_{\max}, \quad k = 0, \ldots, N$ can be written as

$$Y \geq y_{\min}\mathbf{1}, \quad Y \leq y_{\max}\mathbf{1}$$

where $Y = (y_0, \ldots, y_N)$. Introducing

$$\overline{C} = \begin{bmatrix} C & 0 & \ldots & 0 \\ 0 & C & 0 & 0 \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & C \end{bmatrix}$$

we can re-write these inequalities in terms of U via

$$Y \geq y_{\min}\mathbf{1} \Leftrightarrow \overline{C}(GU + Hx_0) \geq y_{\min}\mathbf{1} \Leftrightarrow \underbrace{\overline{C}G}_{A_{\underline{\mathrm{Y}}}} U \geq \underbrace{y_{\min}\mathbf{1} - \overline{C}Hx_0}_{b_{\underline{\mathrm{Y}}}}$$

$$Y \leq y_{\max}\mathbf{1} \Leftrightarrow \overline{C}(GU + Hx_0) \leq y_{\max}\mathbf{1} \Leftrightarrow \underbrace{\overline{C}G}_{A_{\overline{\mathrm{Y}}}} U \leq \underbrace{y_{\max}\mathbf{1} - \overline{C}Hx_0}_{b_{\overline{\mathrm{Y}}}}$$

Predictive control with constraints

Hence, the constrained predictive control problem can be cast as a QP

$$\begin{array}{ll} \text{minimize} & U^T P_{LQ} U + 2 q_{LQ}^T U + r_{LQ} \\ \text{subject to} & \begin{bmatrix} A_{\overline{Y}} \\ -A_{\underline{Y}} \\ I \\ -I \end{bmatrix} U \leq \begin{bmatrix} b_{\overline{Y}} \\ -b_{\underline{Y}} \\ u_{\max}\mathbf{1} \\ -u_{\min}\mathbf{1} \end{bmatrix} \end{array}$$

Solution gives optimal finite-horizon control subject to constraints

Model predictive control:

- apply constrained optimal control in receding horizon fashion

Model predictive control algorithm

1. Given state at time t compute ("predict") future states

$$x_{t+k}, \qquad k = 0, 1, \ldots, N$$

   as function of future control inputs

$$u_{t+k}, \qquad k = 0, 1, \ldots, N-1$$

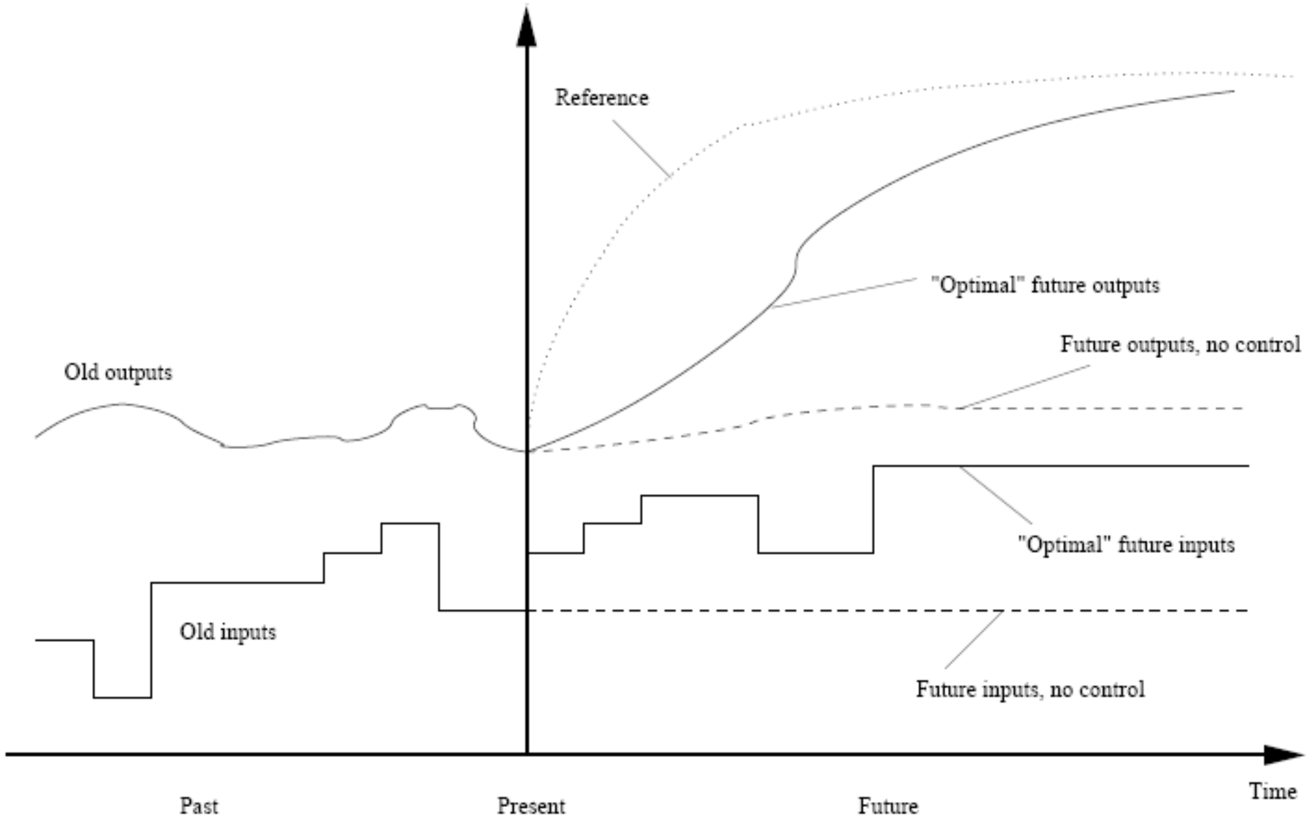
2. Find "optimal" input by minimizing constrained cost function
   - a quadratic program, efficiently solved

3. Implement u(t)

4. A next sample (t+1), return to 1.

A key is that the initial state is updated by an observer (Kalman filter) at each time step, thereby providing feedback from measurements

MPC trajectories



Example: the DC servo

Discrete-time model (sampling time 0.05 sec)

$$A = \begin{bmatrix} 1 & 0.139 \\ 0 & 0.861 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0107 \\ 0.139 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

Constrained input voltage

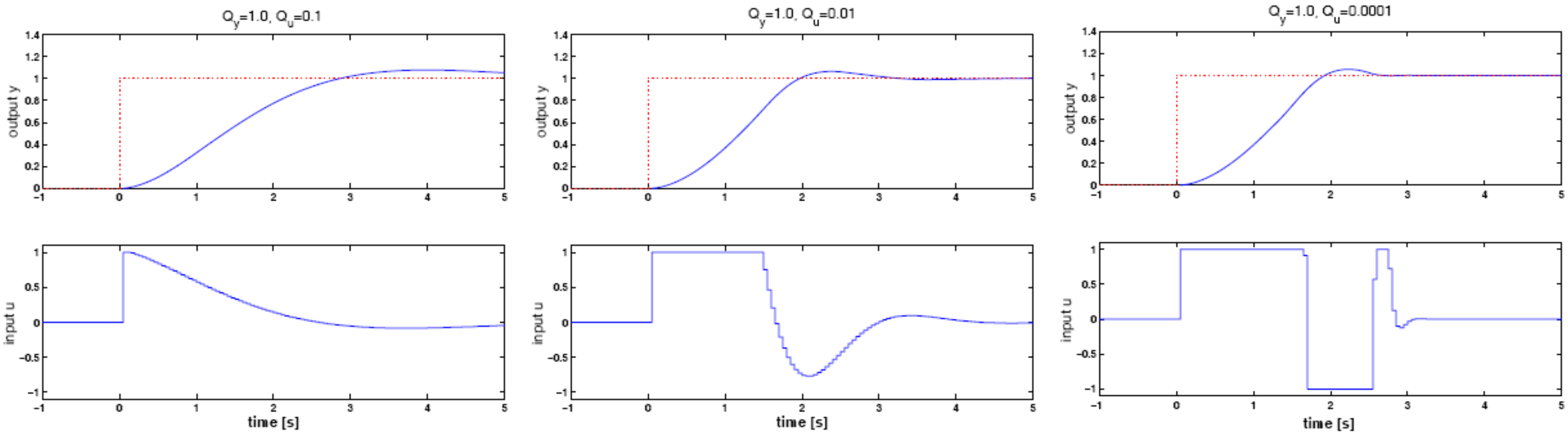
$$-1 \leq u \leq 1$$

Constrained position

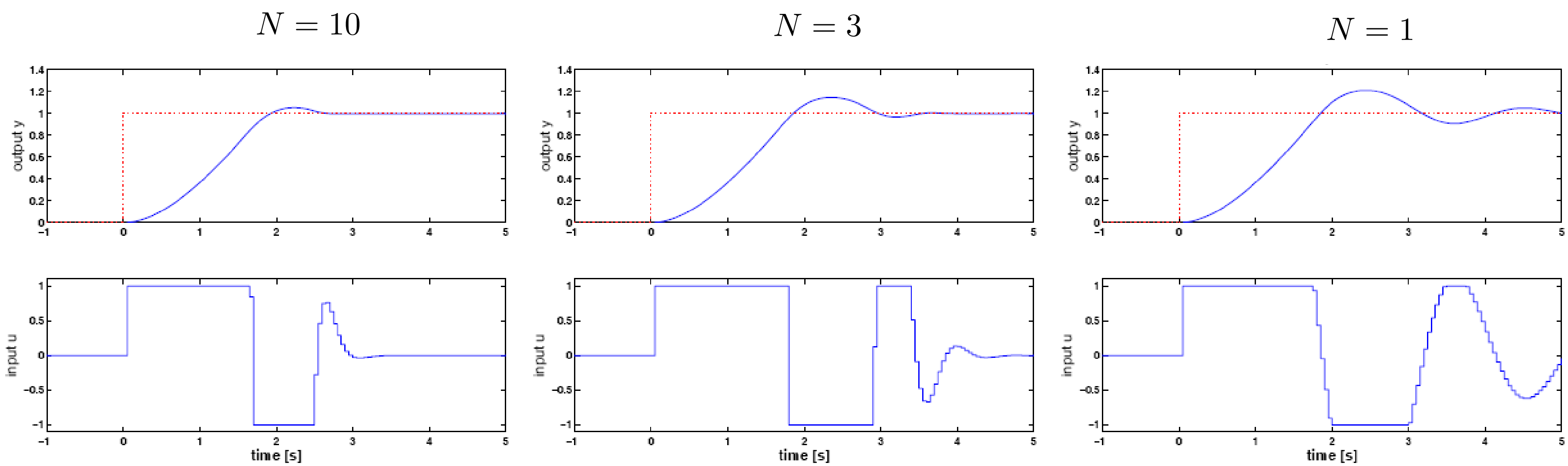
$$y_{\text{min}} \leq y_k \leq y_{\text{max}}$$

Impact of state and control weights

Prediction horizon N=10.

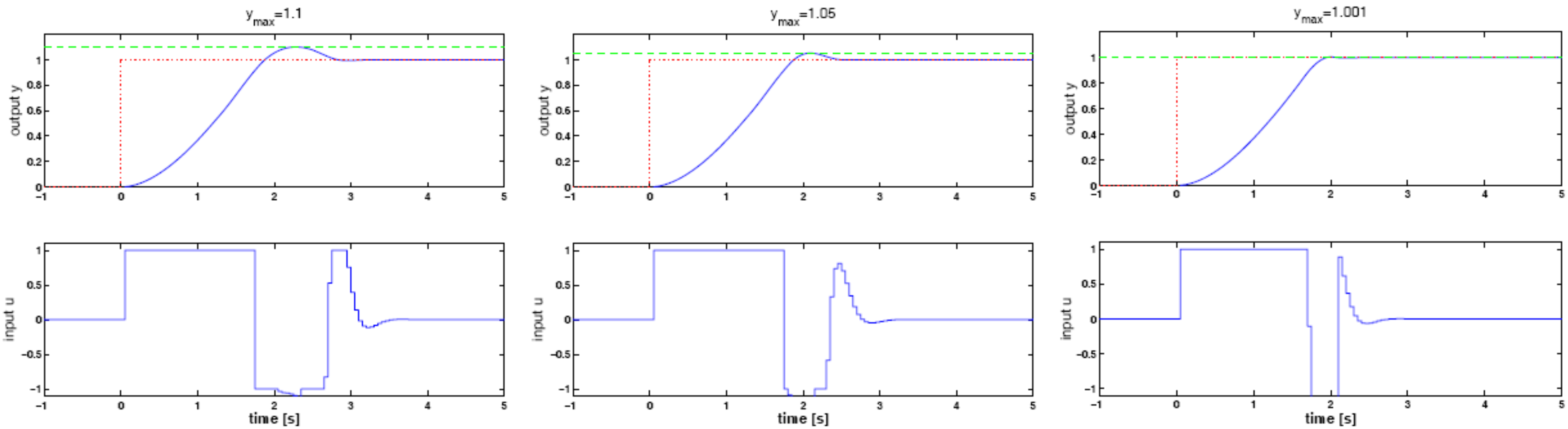


Impact of horizon



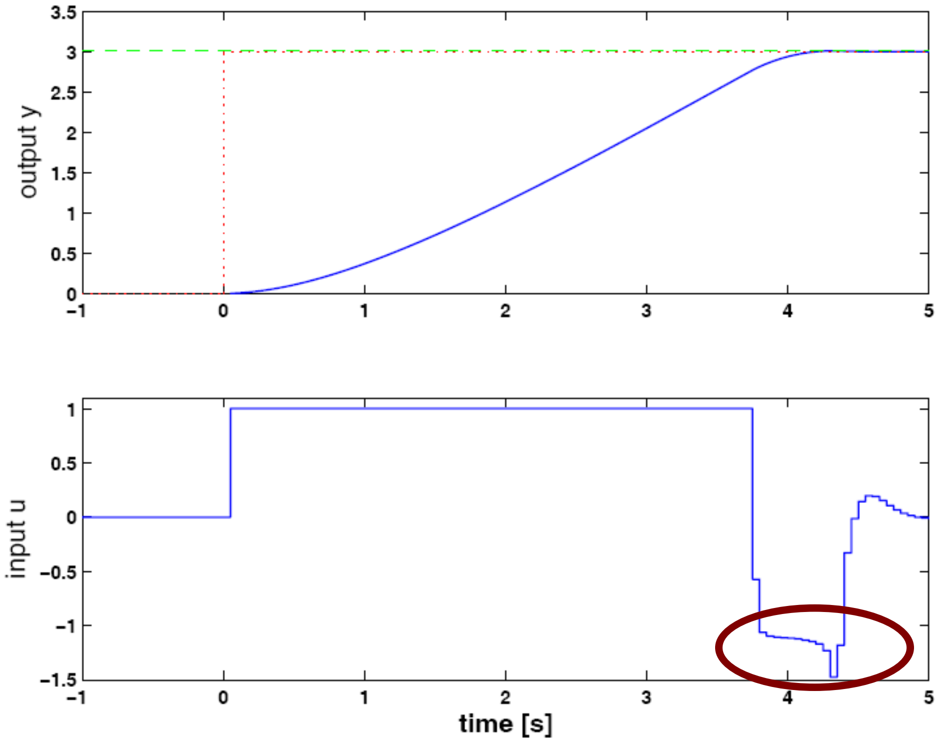
Too short horizon➔inaccurate predictions➔poor performance

Adding output constraints



Infeasibility

What happens when there is no solution to the QP?



Not clear what control to apply!

Ensuring feasibility

One way to ensure feasibility:

- introduce slack variables $s_{ck} \geq 0$
- "soften" constraints

$$u_k \leq u_{\max} \Rightarrow u_k \leq u_{\max} + s_{ck}$$

- add term in quadratic programming objective to minimize slacks

$$\underset{U}{\text{minimize}} \quad U^T P_{LQ} U + 2q_{LQ}^T U + r_{LQ}$$

$$\Downarrow$$

$$\underset{U,S}{\text{minimize}} \quad U^T P_{LQ} U + 2q_{LQ}^T U + r_{LQ} + \kappa S^T S$$

Notes:

- still QP, but more variables; can also use penalty κS (also QP)
- better to soften "physically soft" constraints (e.g. output constraints)

Reference tracking

Would like z to track a reference sequence $\{r_1, \ldots, r_N\}$, i.e. to keep

$$\sum_{k=0}^{N-1} \left((z_k - r_k)^T Q_1 (z_k - r_k) + u_k^T Q_2 u_k \right) + (z_N - r_N)^T Q_f (z_N - r_N)$$

small.

Problem: making $z_k = r_k$ typically requires $u_k \neq 0$

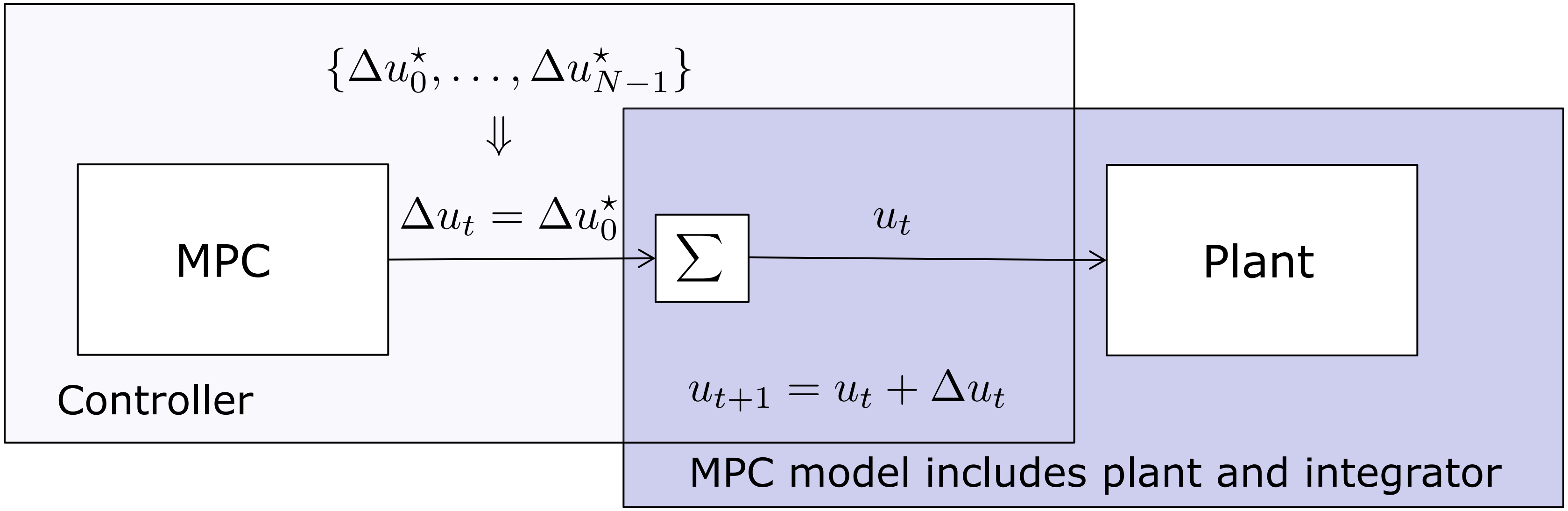
- a trade-off between zero tracking errors and using zero control
- often results in steady-state tracking error

DC motor simulations used MPC with integral action.

Including integral action

Integral action often included by a change in free variables

- Use $\Delta u_i = u_i - u_{i-1}$ as variables in the optimization
- Actual input obtained by summing up MPC outputs



Including integral action cont'd

Form augmented model with state $\bar{x}_k = (x_k,\ u_k)$ and input Δu_k:

$$\begin{bmatrix} x_{k+1} \\ u_{k+1} \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} \Delta u_k$$

$$y_k = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix}$$

Consider finite-horizon cost

$$\sum_{k=0}^{N-1} \left((z_k - r_k)^T Q_1 (z_k - r_k) + \Delta u_k^T Q_2 \Delta u_k \right) + (z_N - r_N)^T Q_f (z_N - r_N)$$

Now, all terms can go to zero (at least when unconstrained, infinite horizon)

Apply control $u_t = u_{t-1} + \Delta u_{t-1}$

MPC controller tuning

MPC has a large number of “tuning” parameters.

The prediction model:

- we need to decide sampling interval
  (rule of thumb: sample 10 times desired closed-loop bandwidth)
- obtain discrete-time state-space model

Finite-horizon optimal control:

- set prediction horizon
  (rule of thumb: equal to closed-loop rise time; could be smaller)
- decide weight matrices (as for continuous-time LQG)
- decide final state penalty
  (guideline: stationary Riccati solution for given weight matrices)

MPC controller tuning

Finite-horizon optimal control, advanced:

- control horizon
  (try to set small, rule-of-thumb: use 1-10)
- inner-loop control
  (guideline: stationary LQR controller for given weight matrices)

Constraints and feasibility

- specify control and state constraints (problem dependent)
- introduce slacks to “soften” constraints
- choose constraint penalty (large value on kappa)

Integral action (almost always a good idea to include).

Advanced issues: stability

Receding horizon control might yield unstable closed-loop

Stability can be guaranteed:

- for infinite-horizon unconstrained case (this is LQR)
- for finite-horizon unconstrained case
  - if final state is penalized correctly
  - if final state is enforced to lie in a given set
- for constrained finite-horizon
  - if final state enforced to lie in a sufficiently small set **and**
  - initial QP (solved at time zero) is feasible

Hard to verify for sure in advance…

Advanced issues: robustness

Consider the unconstrained quadratic program

$$\text{minimize} \quad u^T Q u + 2q^T u$$

has optimal solution $u = -Q^{-1}q$

In the MPC setting, Q and q depend on the system model (matrices A, B, C), weights Q_1, Q_2, and also horizons.

Solution is sensitive to uncertainties if Q is ill-conditioned

- Try scaling inputs and outputs in the model
- Modify weight matrices Q_1 and Q_2
- Almost always a good idea to include integral action

Advanced issues: observers

MPC, as presented here, assumes full state feedback.

In many cases, we will need to use an observer,

- to reconstruct states, and/or
- to filter out noise

Limited theory, but separation principle holds in some cases.

Suggests guideline

- design observer as for (unconstrained, infinite-horizon) LQG
- use estimated state in MPC calculations as if it was true state

New Course on MPC

EL2700 Model Predictive Control, 7.5cr, will be given in period 1, fall term 2016 by professor Mikael Johansson

Summary

Model predictive control (MPC)

- can handle state and control constraints
- predictive control computed via quadratic programming

Many parameters and their influence on the control

- System model, weights, horizons, constraints, …

Advanced issues:

- Feasibility and slacks to "soften" constraints
- Integral action
- Different prediction and control horizons
- Stability and the terminal weight
- The need for a state observer