

# C21 Model Predictive Control

Mark Cannon

4 lectures

Michaelmas Term 2017



## Lecture 1

# Introduction

# Organisation

- ▷ 4 lectures: week 3      {    Wednesday    12-1 pm    LR2  
                               {    Friday        12-1 pm    LR2
  
- week 4      {    Wednesday    12-1 pm    LR2  
                               {    Friday        12-1 pm    LR2
  
- ▷ 1 class:      week 5      Friday    5-6 pm    LR5
- week 6      {    Thursday    4-5 pm    LR5  
                               {    Friday        4-5 pm    LR6

# Course outline

1. Introduction to MPC and constrained control
2. Prediction and optimization
3. Closed loop properties
4. Disturbances and integral action
5. Robust tube MPC

- ① B. Kouvaritakis and M. Cannon, *Model Predictive Control: Classical, Robust and Stochastic*, Springer 2015

Recommended reading: Chapters 1, 2 & 3

- ② J.B. Rawlings and D.Q. Mayne, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009

- ③ J.M. Maciejowski, *Predictive control with constraints*. Prentice Hall, 2002

Recommended reading: Chapters 1–3, 6 & 8

# Introduction

Classical controller design:

1. Determine plant model
2. Design controller (e.g. PID)
3. Apply controller

discard model

$$x_{k+1} = f(x_k, u_k)$$

↓



Model predictive control (MPC):

1. Use model to predict system behaviour
2. Choose optimal trajectory
3. Repeat procedure (feedback)

# Introduction

Classical controller design:

1. Determine plant model
2. Design controller (e.g. PID)
3. Apply controller

discard model

$$x_{k+1} = f(x_k, u_k)$$

↓



Model predictive control (MPC):

1. Use model to **predict** system behaviour
2. Choose **optimal** trajectory
3. Repeat procedure (feedback)

user-defined optimality criterion



# Overview of MPC

Model predictive control strategy:

1. Prediction
2. Online optimization
3. Receding horizon implementation

## 1. Prediction

- \* Plant model:  $x_{k+1} = f(x_k, u_k)$
- \* Simulate forward in time (over a prediction horizon of  $N$  steps)

$$\text{input sequence: } \mathbf{u}_k = \begin{bmatrix} u_{0|k} \\ u_{1|k} \\ \vdots \\ u_{N-1|k} \end{bmatrix} \xrightarrow{\text{defines}} \text{state sequence: } \mathbf{x}_k = \begin{bmatrix} x_{1|k} \\ x_{2|k} \\ \vdots \\ x_{N|k} \end{bmatrix}$$

Notation:  $(u_{i|k}, x_{i|k}) = \text{predicted } i \text{ steps ahead | evaluated at time } k$   
 $x_{0|k} = x_k$

## 2. Optimization

- \* Predicted quality criterion/cost:  $J_k = \sum_{i=0}^N l_i(x_{i|k}, u_{i|k})$

$l_i(x, u)$ : stage cost

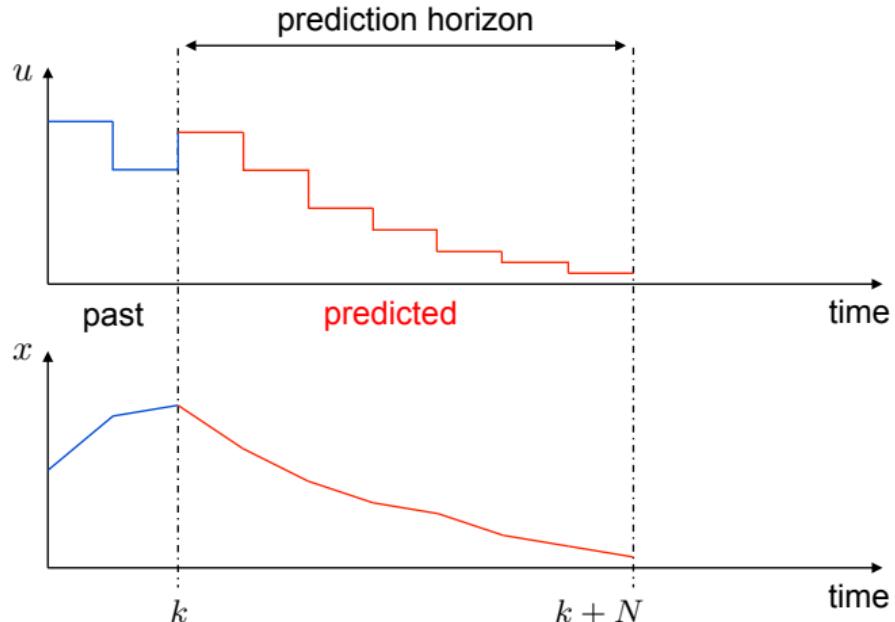
- \* Solve numerically to determine optimal input sequence:

$$\begin{aligned}\mathbf{u}_k^* &= \arg \min_{\mathbf{u}_k} J_k \\ &= (u_{0|k}^*, \dots, u_{N-1|k}^*)\end{aligned}$$

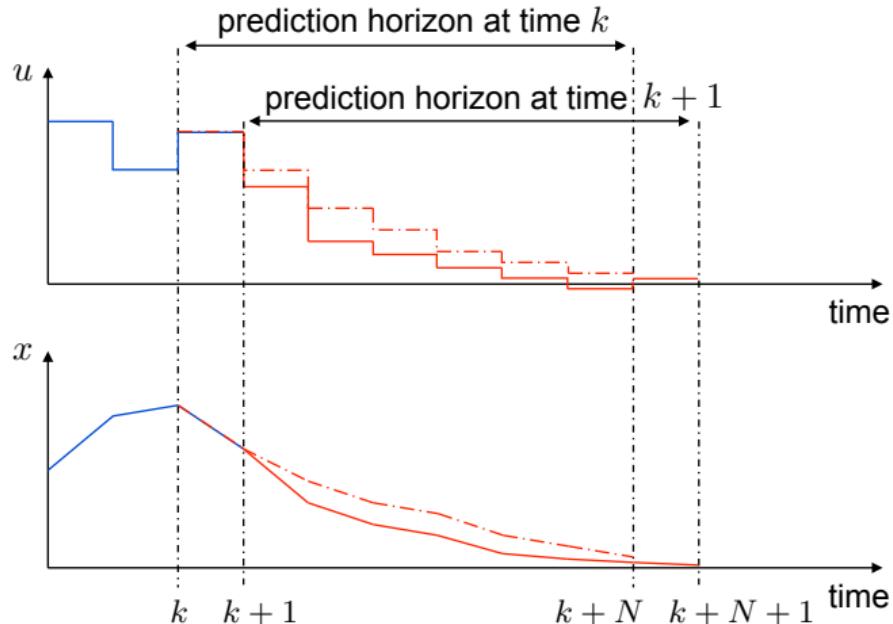
## 3. Implementation

- \* Use first element of  $\mathbf{u}_k^*$   $\implies$  actual plant input  $u_k = u_{0|k}^*$
- \* Repeat optimization at each sampling instant  $k = 0, 1, \dots$

# Overview of MPC



# Overview of MPC



# Overview of MPC

Optimization is repeated online at each sampling instant  $k = 0, 1, \dots$



receding horizon:

$$\begin{array}{ll} \mathbf{u}_k = (u_{0|k}, \dots, u_{N-1|k}) & \mathbf{x}_k = (x_{1|k}, \dots, x_{N|k}) \\ \mathbf{u}_{k+1} = (u_{0|k+1}, \dots, u_{N-1|k+1}) & \mathbf{x}_{k+1} = (x_{1|k+1}, \dots, x_{N|k+1}) \\ \vdots & \vdots \end{array}$$

- ★ provides feedback since  $\mathbf{u}_k^*, \mathbf{x}_k^*$  are functions of  $x_k$ 
  - so reduces effects of model error and measurement noise
- ★ and compensates for finite number of free variables in predictions
  - so improves closed-loop performance

## Example

Plant model:

$$x_{k+1} = \begin{bmatrix} 1.1 & 2 \\ 0 & 0.95 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 0.0787 \end{bmatrix} u_k$$
$$y_k = \begin{bmatrix} -1 & 1 \end{bmatrix} x_k$$

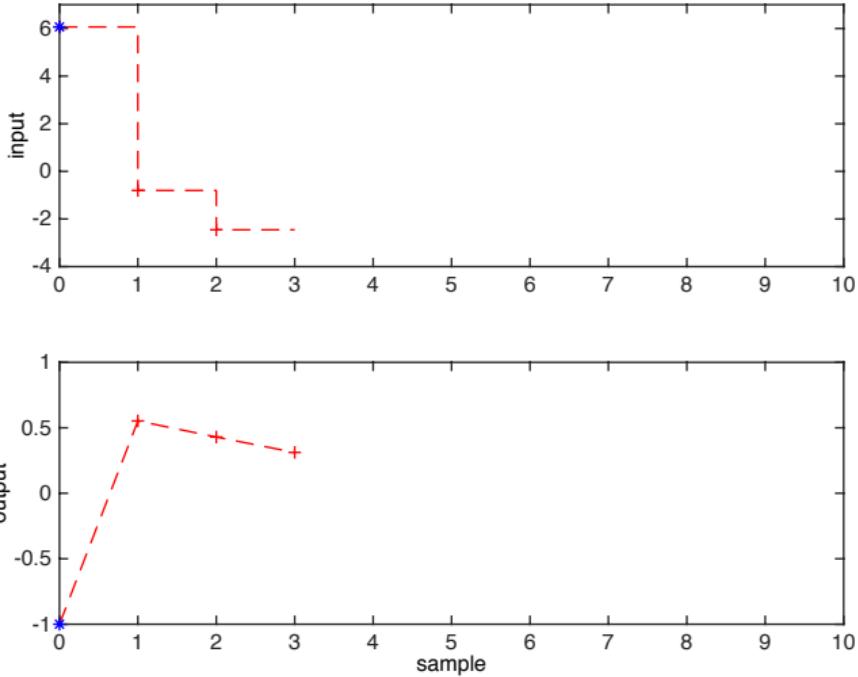
Cost:

$$\sum_{i=0}^{N-1} (y_{i|k}^2 + u_{i|k}^2) + y_{N|k}^2$$

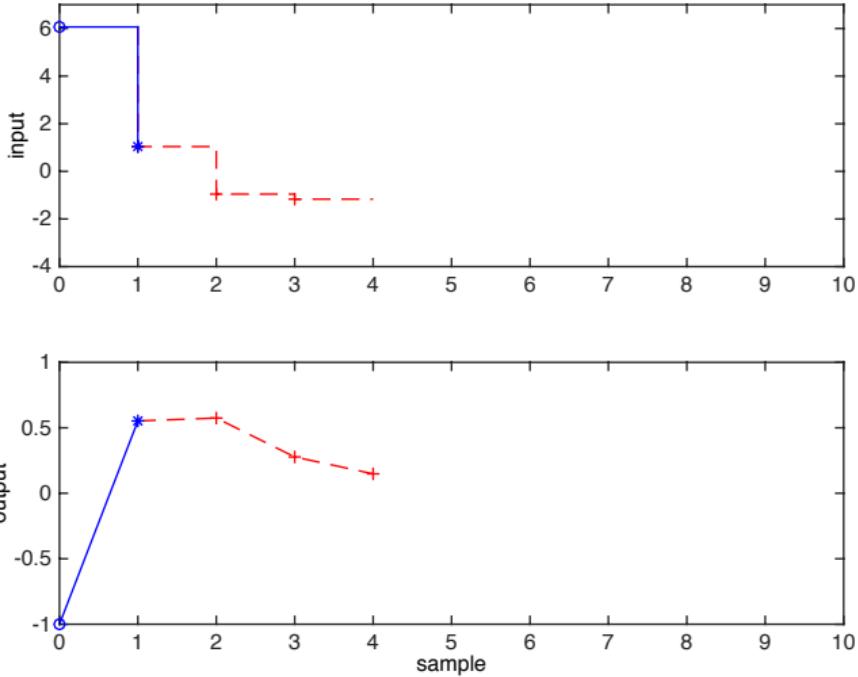
Prediction horizon:  $N = 3$

Free variables in predictions:  $\mathbf{u}_k = \begin{bmatrix} u_{0|k} \\ u_{1|k} \\ u_{2|k} \end{bmatrix}$

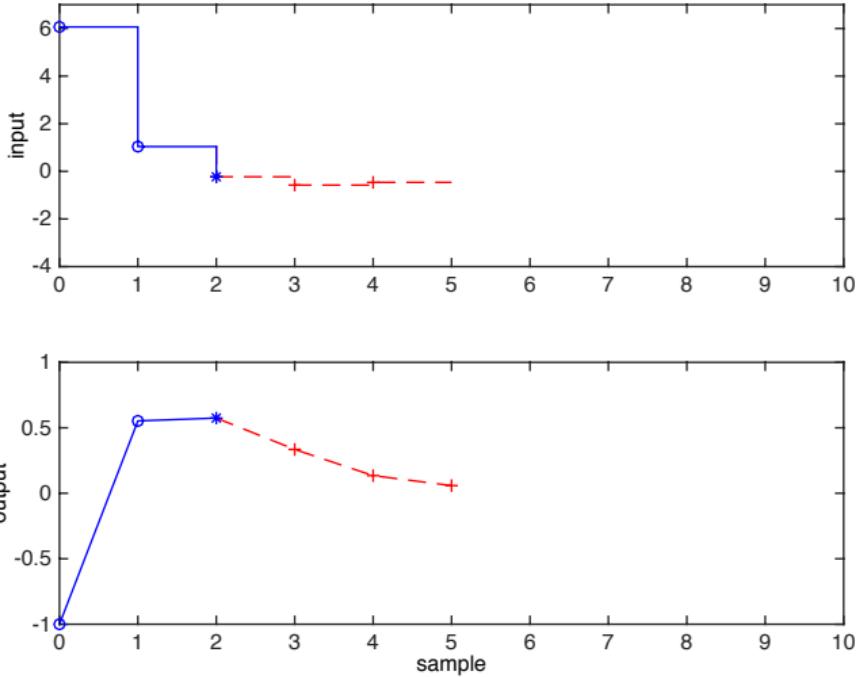
## Example



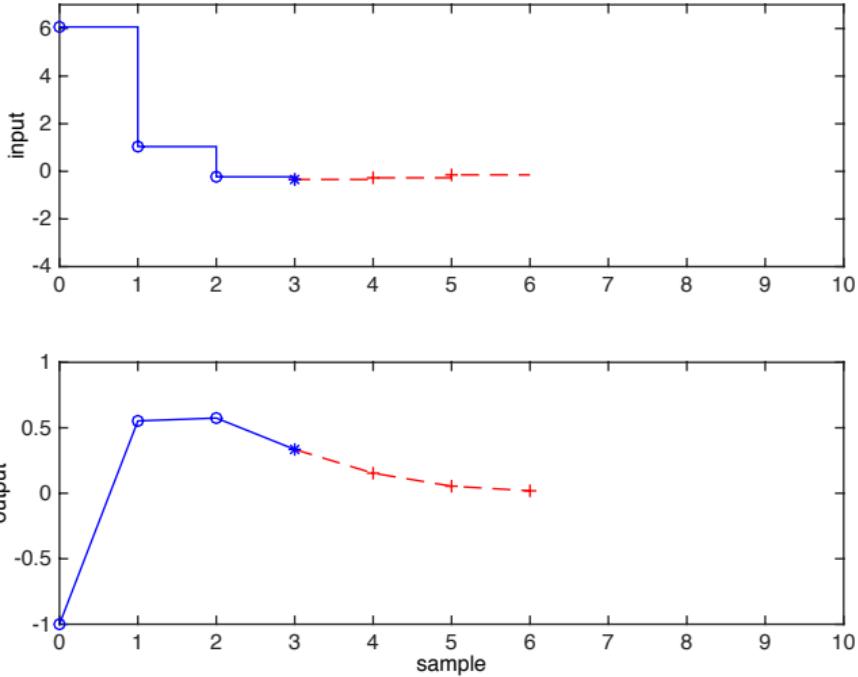
# Example



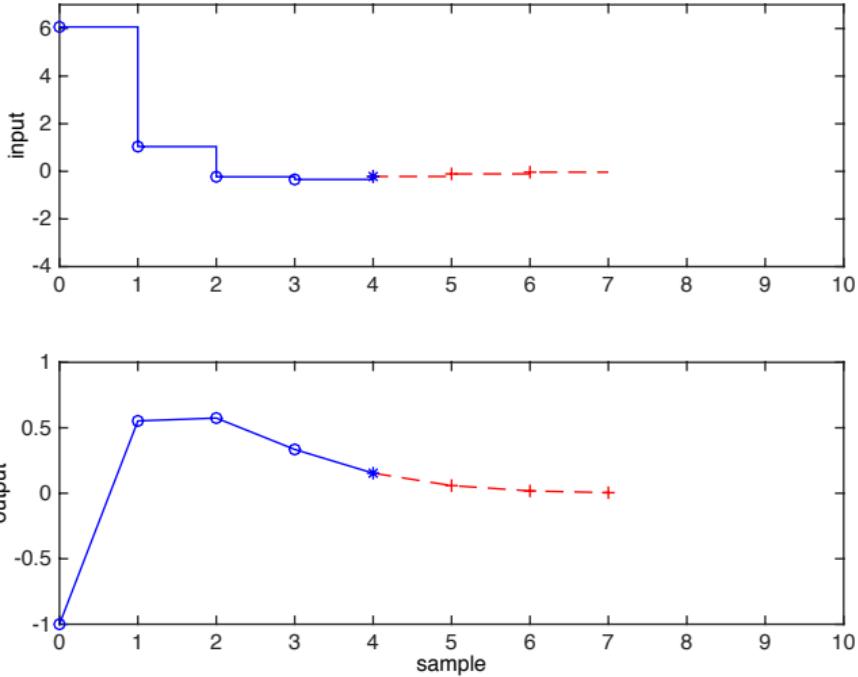
## Example



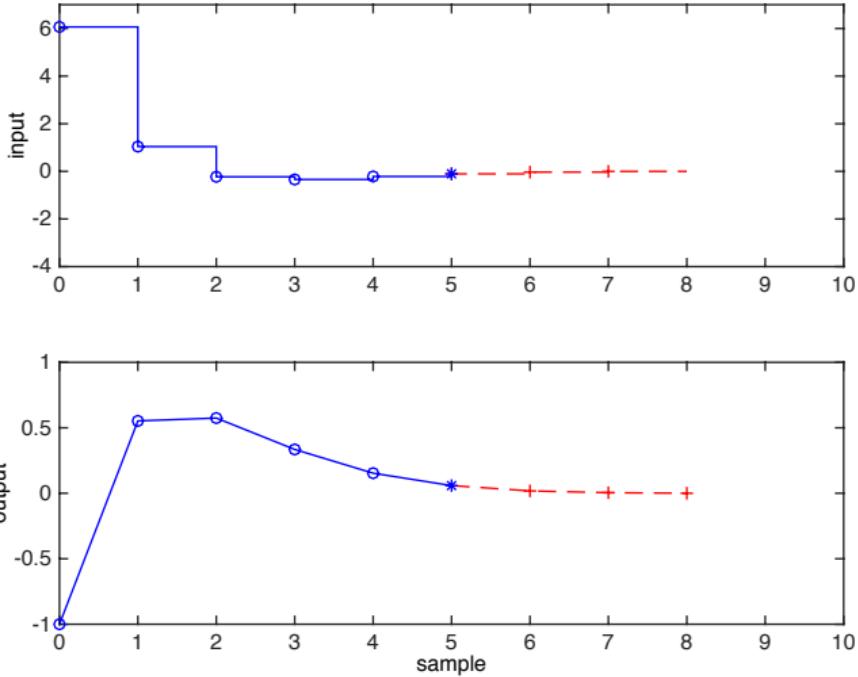
## Example



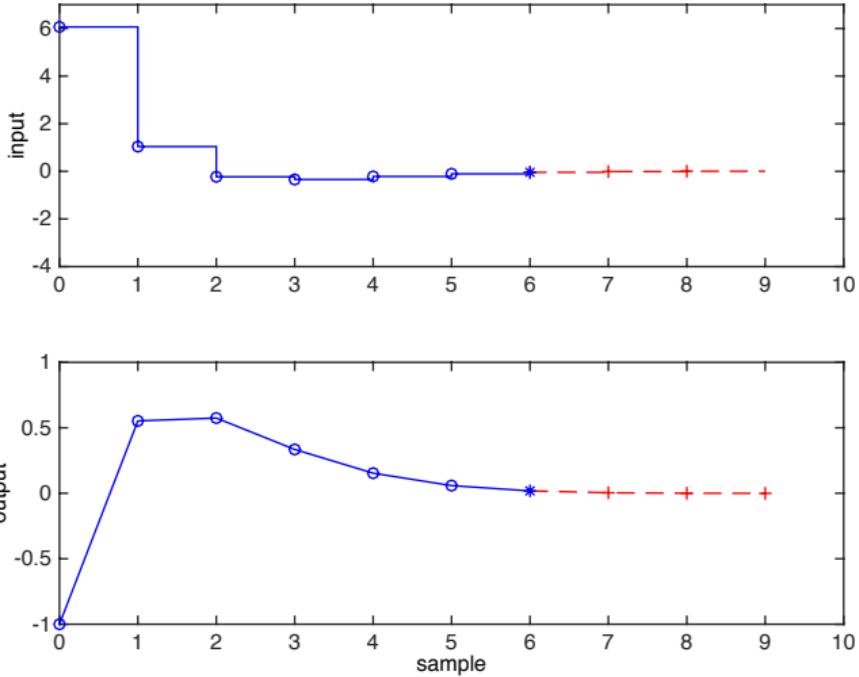
## Example



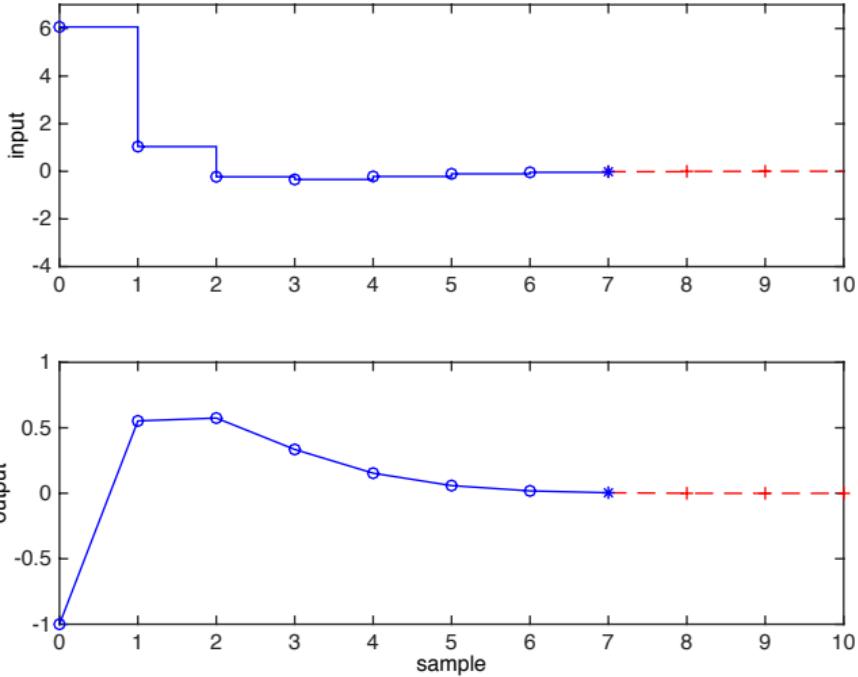
# Example



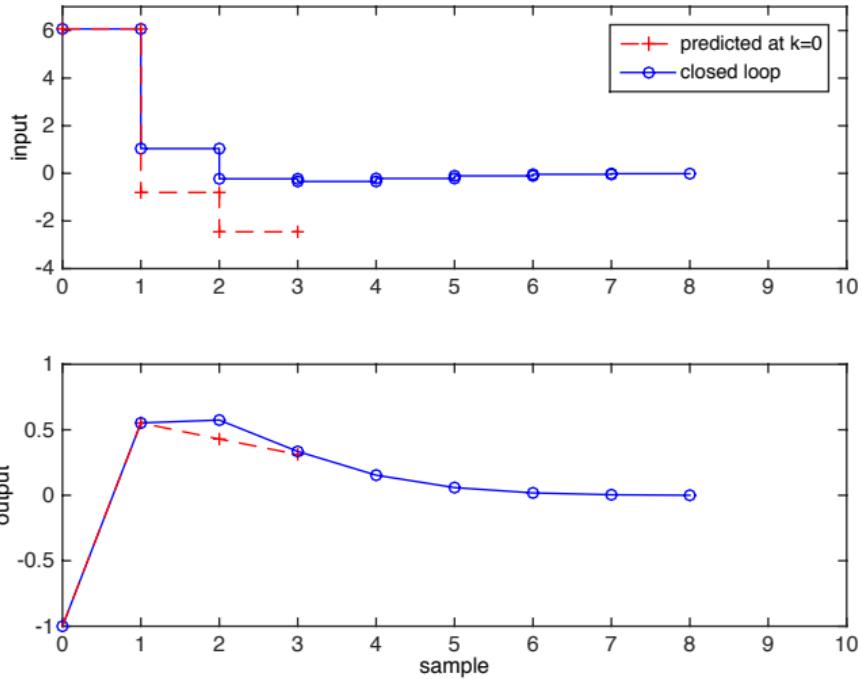
## Example



## Example



# Example



# Motivation for MPC

## Advantages

- ▷ Flexible plant model
  - e.g. multivariable
    - linear or nonlinear
    - deterministic, stochastic or fuzzy
- ▷ Handles constraints on control inputs and states
  - e.g. actuator limits
    - safety, environmental and economic constraints
- ▷ Approximately optimal control

## Disadvantages

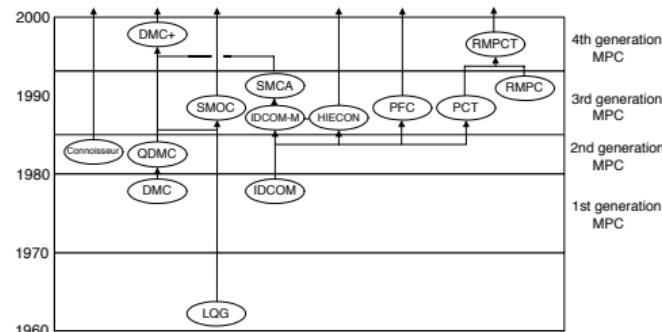
- ▷ Requires online optimization
  - e.g. large computation for nonlinear and uncertain systems

# Historical development

Control strategy reinvented several times

LQ(G) optimal control	1950's–1980's
industrial process control	1980's
constrained nonlinear control	1990's–today

Development of commercial MPC algorithms:



[from Qin & Badgwell (2003)]

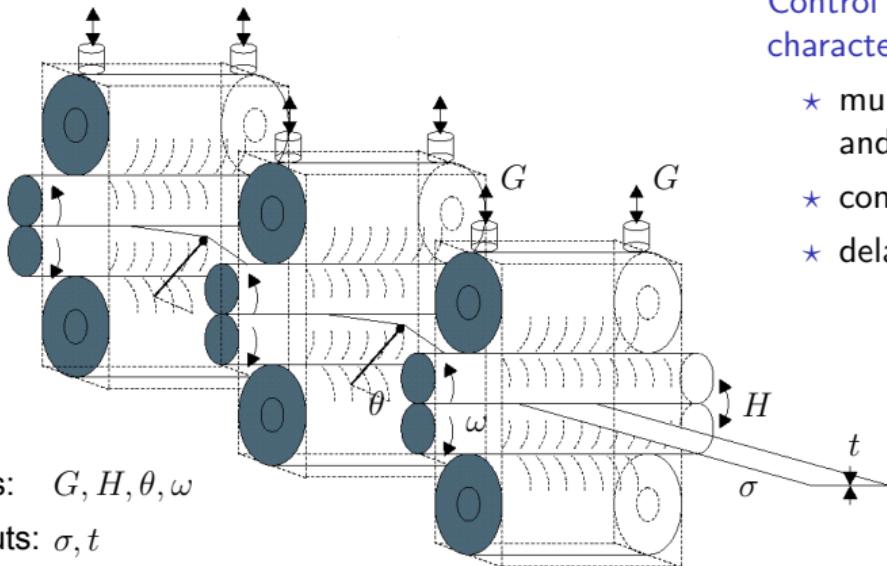
# Applications: Process Control

Steel hot rolling mill



# Applications: Process Control

Steel hot rolling mill



Inputs:  $G, H, \theta, \omega$

Outputs:  $\sigma, t$

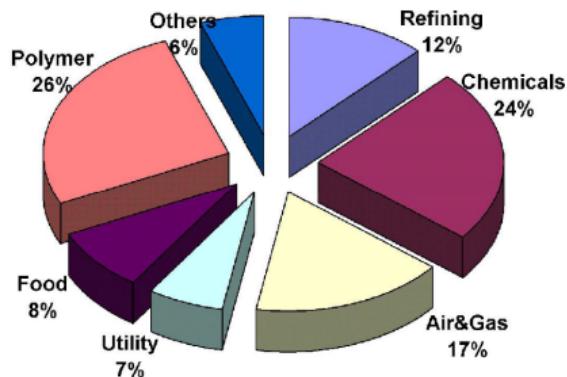
Objectives: control residual stress  $\sigma$   
and thickness  $t$

Control problem characteristics:

- \* multiple inputs and outputs
- \* constraints
- \* delays

# Applications: Chemical Process Control

- Applications of predictive control to more than 4,500 different chemical processes (based on a 2006 survey)
- MPC applications in the chemical industry



[from Nagy (2006)]

Typical control problems:

- ★ nonlinear dynamics
- ★ slow sampling rates
- ★ non-quadratic utility functions

# Applications: Electromechanical systems

## Variable-pitch wind turbines



### Control problem characteristics:

- ★ stochastic uncertainty
- ★ fatigue constraints

# Applications: Electromechanical systems

Predictive swing-up and balancing controllers



Autonomous racing for remote controlled cars



Control problem characteristics:

- ★ reference tracking
- ★ short sampling intervals
- ★ nonlinear dynamics

# Prediction model

Linear plant model:  $x_{k+1} = Ax_k + Bu_k$

- ▷ Predicted  $\mathbf{x}_k$  depends linearly on  $\mathbf{u}_k$  [details in Lecture 2]
- ▷ Therefore the cost is quadratic in  $\mathbf{u}_k$      $\mathbf{u}_k^T H \mathbf{u}_k + 2f^T \mathbf{u}_k + g(x_k)$   
and constraints are linear     $A_c \mathbf{u}_k \leq b(x_k)$
- ▷ Online optimization:

$$\min_{\mathbf{u}} \quad \mathbf{u}^T H \mathbf{u} + 2f^T \mathbf{u} \quad \text{s.t.} \quad A_c \mathbf{u} \leq b_c$$

is a convex Quadratic Program (QP),  
which is reliably and efficiently solvable

# Prediction model

Nonlinear plant model:  $x_{k+1} = f(x_k, u_k)$

▷ Predicted  $\mathbf{x}_k$  depends nonlinearly on  $\mathbf{u}_k$

▷ In general the cost is nonconvex in  $\mathbf{u}_k$        $J_k(x_k, \mathbf{u}_k)$   
and the constraints are nonconvex       $g_c(x_k, \mathbf{u}_k) \leq 0$

▷ Online optimization:

$$\min_{\mathbf{u}} \quad J_k(x_k, \mathbf{u}) \quad \text{s.t.} \quad g_c(x_k, \mathbf{u}) \leq 0$$

is nonconvex

may have local minima

may not be solvable efficiently or reliably

# Prediction model

## Discrete time prediction model

- ▷ Predictions optimized periodically at  $t = 0, T, 2T, \dots$
- ▷ Usually  $T = T_s = \text{sampling interval of model}$
- ▷ But  $T = nT_s$  for  $n > 1$  is also possible, e.g. if  $T_s <$  time required to perform online optimization

$n = \text{integer}$  so a time-shifted version of the optimal input sequence at time  $k$  can be implemented at time  $k + 1$

(allows a guarantee of stability – [Lecture 3])

e.g. if  $n = 1$ , then  $\mathbf{u}_{k+1} = (\underline{u}_{1|k}, \dots, \underline{u}_{N-1|k}, u_{N|k})$  is possible,  
where  $(u_{0|k}, \underline{u}_{1|k}, \dots, \underline{u}_{N-1|k}) = \mathbf{u}_k^*$

# Prediction model

## Continuous time prediction model

- ▷ Predicted  $u(t)$  need not be piecewise constant,
  - e.g. 1st order hold gives continuous, piecewise linear  $u(t)$
  - or  $u(t) = \text{polynomial in } t$  (piecewise quadratic, cubic etc)
- ▷ Continuous time prediction model can be integrated online,
  - which is useful for nonlinear continuous time systems
- ▷ This course: discrete-time model and  $T = T_s$  assumed

# Constraints

Constraints are present in almost all control problems

- ▷ Input constraints, e.g. box constraints:

$$\underline{u} \leq u_k \leq \bar{u} \quad (\text{absolute})$$

$$\underline{\Delta u} \leq u_k - u_{k-1} \leq \bar{\Delta u} \quad (\text{rate})$$

- ★ typically active during transients, e.g. valve saturation  
or d.c. motor saturation

- ▷ State constraints, e.g. box constraints

$$\underline{x} \leq x_k \leq \bar{x} \quad (\text{linear})$$

- ★ can be active during transients, e.g. aircraft stall speed
- ★ and in steady state, e.g. economic constraints

# Constraints

Classify constraints as either **hard** or **soft**

- ▷ Hard constraints must be satisfied at all times,  
if this is not possible, then the problem is **infeasible**
- ▷ Soft constraints can be violated to avoid infeasibility
- ▷ Strategies for handling soft constraints:
  - \* impose (hard) constraints on the probability of violating each soft constraint
  - \* or remove active constraints until the problem becomes feasible
- ▷ This course: only hard constraints are considered

# Constraint handling

Suboptimal methods for handling input constraints:

(a). Saturate the unconstrained control law

constraints are then usually ignored in controller design

(b). “De-tune” the unconstrained control law

increase the penalty on  $u$  in the performance objective

(c). Use an anti-windup strategy

to put limits on the state of a dynamic controller  
(typically the integral term of a PI or PID controller)

# Constraint handling

Effects of **input saturation**,  $\underline{u} \leq u_k \leq \bar{u}$

unconstrained control law:  $u = u^0$

saturated control law:  $u = \max\{\min\{u^0, \bar{u}\}, \underline{u}\}$

Example:

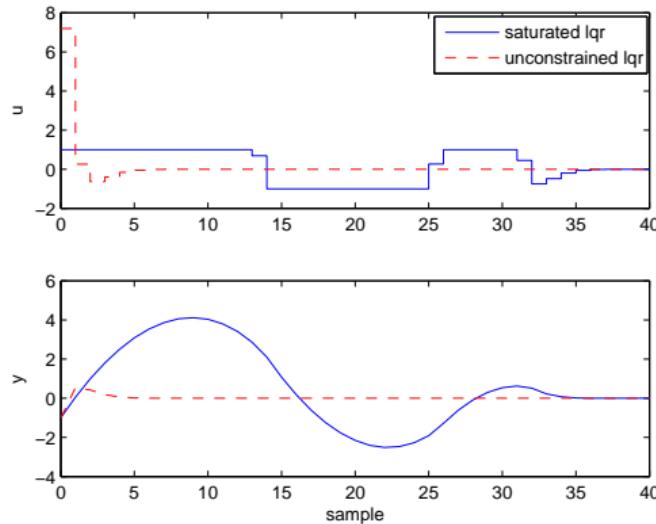
$(A, B, C)$  as before

$$\underline{u} = -1, \bar{u} = 1$$

$$u^0 = K_{LQ}x$$

Input saturation causes

- ★ poor performance
- ★ possible instability  
(since the open-loop system is unstable)



# Constraint handling

De-tuning of optimal control law:

$$K_{LQ} = \text{optimal gain for LQ cost } J^\infty = \sum_{k=0}^{\infty} (\|x_k\|_Q^2 + \|u_k\|_R^2)$$

Increase  $R$  until  $u = K_{LQ}x$  satisfies constraints for all initial conditions

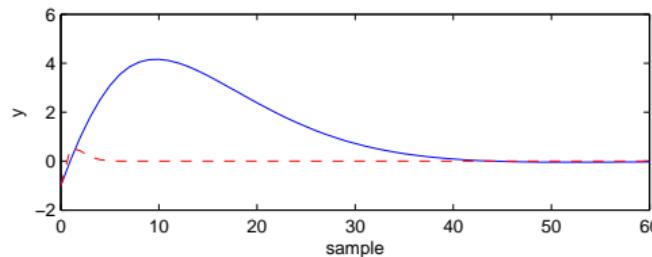
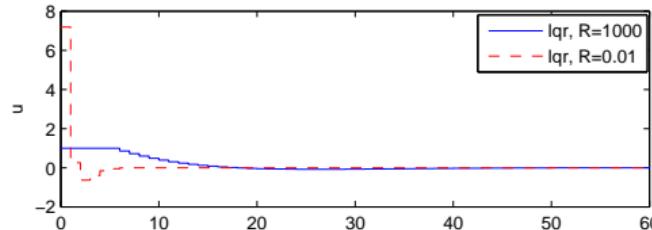
Example:

$(A, B, C)$  as before  
 $10^{-2} \leq R \leq 10^3$



settling time increased  
from 6 to 40

- ★  $y(t) \rightarrow 0$  slowly
- ★ stability can be ensured



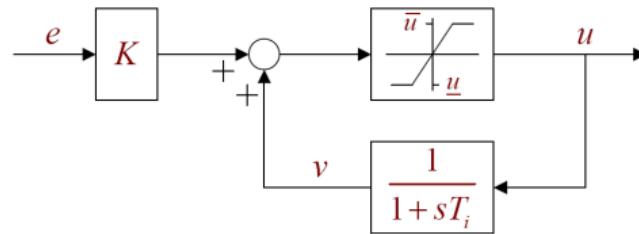
# Constraint handling

Anti-windup prevents instability of the controller while the input is saturated

Many possible approaches, e.g. anti-windup PI controller:

$$u = \max\{\min\{(Ke + v), \bar{u}\}, \underline{u}\}$$

$$T_i \dot{v} + v = u$$



$$\underline{u} \leq u \leq \bar{u} \implies u = K \left( e + \frac{1}{T_i} \int^t e dt \right)$$

$$u = \underline{u} \text{ or } \bar{u} \implies v(t) \rightarrow \underline{u} \text{ or } \bar{u} \text{ exponentially}$$

Strategy is suboptimal and may not prevent instability

# Constraint handling

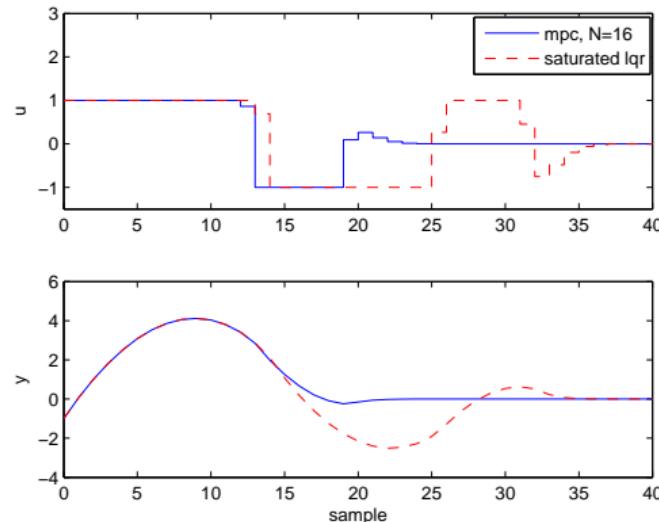
Anti-windup is based in the past behaviour of the system, whereas MPC optimizes future performance

Example:

$(A, B, C)$  as before

MPC vs saturated LQ  
(both based on cost  $J^\infty$ ):

- settling time reduced to 20 by MPC
- stability is guaranteed with MPC



# Summary

- ▷ Predict performance using plant model
  - e.g. linear or nonlinear, discrete or continuous time
- ▷ Optimize future (open loop) control sequence
  - computationally much easier than optimizing over feedback laws
- ▷ Implement first sample, then repeat optimization
  - provides feedback to reduce effect of uncertainty
- ▷ Comparison of common methods of handling constraints:
  - saturation, de-tuning, anti-windup, MPC

## Lecture 2

# Prediction and optimization

# Prediction and optimization

- Input and state predictions
- Unconstrained finite horizon optimal control
- Infinite prediction horizons and connection with LQ optimal control
- Incorporating constraints
- Quadratic programming

# Review of MPC strategy

At each sampling instant:

1. Use a model to **predict** system behaviour over a finite future horizon
2. Compute a control sequence by solving an **online optimization** problem
3. Apply the **first element** of optimal control sequence as control input



## Advantages

- ★ flexible plant model
- ★ constraints taken into account
- ★ optimal performance

## Disadvantage

- ★ online optimization required

# Prediction equations

Linear time-invariant model:  $x_{k+1} = Ax_k + Bu_k$

assume  $x_k$  is measured at  $k = 0, 1, \dots$

Predictions:  $\mathbf{x}_k = \begin{bmatrix} x_{1|k} \\ \vdots \\ x_{N|k} \end{bmatrix}$        $\mathbf{u}_k = \begin{bmatrix} u_{0|k} \\ \vdots \\ u_{N-1|k} \end{bmatrix}$

Quadratic cost:  $J_k = J(x_k, \mathbf{u}_k)$

$$= \sum_{i=0}^{N-1} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2) + \|x_{N|k}\|_P^2$$

$$\left( \|x\|_Q^2 = x^T Q x, \quad \|u\|_R^2 = u^T R u \right. \\ \left. P = \text{terminal weighting matrix} \right)$$

# Prediction equations

Linear time-invariant model:

$$x_{i+1|k} = Ax_{i|k} + Bu_{i|k}$$

assume  $x_k$  is measured at  $k = 0, 1, \dots$

$$x_{0|k} = x_k$$

$$x_{1|k} = Ax_k + Bu_{0|k}$$

⋮

$$x_{N|k} = A^N x_k + A^{N-1} Bu_{0|k} + A^{N-2} Bu_{1|k} + \cdots + Bu_{N-1|k}$$



$$\mathbf{x}_k = \mathcal{M}\mathbf{x}_k + \mathcal{C}\mathbf{u}_k$$

$$\mathcal{C} = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix} \quad \mathcal{M} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}$$

or

$$x_{i|k} = A^i x_k + \mathcal{C}_i \mathbf{u}_k, \quad \mathcal{C}_i = i\text{th row of } \mathcal{C}$$

# Prediction equations

Linear time-invariant model:

$$x_{i+1|k} = Ax_{i|k} + Bu_{i|k}$$

assume  $x_k$  is measured at  $k = 0, 1, \dots$

$$x_{0|k} = x_k$$

$$x_{1|k} = Ax_k + Bu_{0|k}$$

⋮

$$x_{N|k} = A^N x_k + A^{N-1} Bu_{0|k} + A^{N-2} Bu_{1|k} + \cdots + Bu_{N-1|k}$$



$$\mathbf{x}_k = \mathcal{M}x_k + \mathcal{C}\mathbf{u}_k$$

$$\mathcal{C} = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix} \quad \mathcal{M} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}$$

or

$$x_{i|k} = A^i x_k + \mathcal{C}_i \mathbf{u}_k, \quad \mathcal{C}_i = i\text{th row of } \mathcal{C}$$

# Prediction equations

Predicted cost:

$$\begin{aligned} J_k &= \sum_{i=0}^{N-1} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2) + \|x_{N|k}\|_P^2 \\ &= x_k^T Q x_k + \mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k \quad \left\{ \begin{array}{l} \mathbf{Q} = \text{diag}\{Q, \dots, Q, P\} \\ \mathbf{R} = \text{diag}\{R, \dots, R, R\} \end{array} \right. \end{aligned}$$



$$J_k = \mathbf{u}_k^T H \mathbf{u}_k + 2x_k^T F^T \mathbf{u}_k + x_k^T G x_k$$

where

$$H = \mathcal{C}^T \mathbf{Q} \mathcal{C} + \mathbf{R} \quad \leftarrow \mathbf{u} \times \mathbf{u} \text{ terms}$$

$$F = \mathcal{C}^T \mathbf{Q} \mathcal{M} \quad \leftarrow \mathbf{u} \times x \text{ terms}$$

$$G = \mathcal{M}^T \mathbf{Q} \mathcal{M} + Q \quad \leftarrow x \times x \text{ terms}$$

time-invariant model  $\implies H, F, G$  can be computed offline

# Prediction equations

Predicted cost:

$$\begin{aligned} J_k &= \sum_{i=0}^{N-1} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2) + \|x_{N|k}\|_P^2 \\ &= x_k^T Q x_k + \mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k \quad \begin{cases} \mathbf{Q} = \text{diag}\{Q, \dots, Q, P\} \\ \mathbf{R} = \text{diag}\{R, \dots, R, R\} \end{cases} \end{aligned}$$



$$J_k = \mathbf{u}_k^T H \mathbf{u}_k + 2x_k^T F^T \mathbf{u}_k + x_k^T G x_k$$

where

$$H = \mathcal{C}^T \mathbf{Q} \mathcal{C} + \mathbf{R} \quad \leftarrow \mathbf{u} \times \mathbf{u} \text{ terms}$$

$$F = \mathcal{C}^T \mathbf{Q} \mathcal{M} \quad \leftarrow \mathbf{u} \times x \text{ terms}$$

$$G = \mathcal{M}^T \mathbf{Q} \mathcal{M} + Q \quad \leftarrow x \times x \text{ terms}$$

time-invariant model  $\implies H, F, G$  can be computed offline

## Prediction equations – example

Plant model:  $x_{k+1} = Ax_k + Bu_k, \quad y_k = Cx_k$

$$A = \begin{bmatrix} 1.1 & 2 \\ 0 & 0.95 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0.079 \end{bmatrix}, \quad C = [-1 \quad 1]$$

Prediction horizon  $N = 4$ :  $\mathcal{C} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.079 & 0 & 0 & 0 \\ 0.157 & 0 & 0 & 0 \\ 0.075 & 0.079 & 0 & 0 \\ 0.323 & 0.157 & 0 & 0 \\ 0.071 & 0.075 & 0.079 & 0 \\ 0.497 & 0.323 & 0.157 & 0 \\ 0.068 & 0.071 & 0.075 & 0.079 \end{bmatrix}$

Cost matrices  $Q = C^T C$ ,  $R = 0.01$ , and  $P = Q$ :

$$H = \begin{bmatrix} 0.271 & 0.122 & 0.016 & -0.034 \\ 0.122 & 0.086 & 0.014 & -0.020 \\ 0.016 & 0.014 & 0.023 & -0.007 \\ -0.034 & -0.020 & -0.007 & 0.016 \end{bmatrix} \quad F = \begin{bmatrix} 0.977 & 4.925 \\ 0.383 & 2.174 \\ 0.016 & 0.219 \\ -0.115 & -0.618 \end{bmatrix}$$

$$G = \begin{bmatrix} 7.589 & 22.78 \\ 22.78 & 103.7 \end{bmatrix}$$

# Prediction equations: LTV model

Aside: Linear time-varying model:  $x_{k+1} = A_k x_k + B_k u_k$   
assume  $x_k$  is measured at  $k = 0, 1 \dots$

Predictions:  $x_{0|k} = x_k$

$$x_{1|k} = A_k x_k + B_k u_{0|k}$$

$$x_{2|k} = A_{k+1} A_k x_k + A_{k+1} B_k u_{0|k} + B_{k+1} u_{1|k}$$

⋮



$$x_{i|k} = \prod_{j=i-1}^0 A_{k+j} x_k + \mathcal{C}_i(k) \mathbf{u}_k, \quad i = 0, \dots, N$$

$$\mathcal{C}_i(k) = \begin{bmatrix} \prod_{j=i-1}^1 A_{k+j} B_k & \prod_{j=i-1}^2 A_{k+j} B_{k+1} & \cdots & B_{k+i-1} & 0 & \cdots & 0 \end{bmatrix}$$



$H(k)$ ,  $F(k)$ ,  $G(k)$  depend on  $k$  and must be computed online

# Unconstrained optimization

Minimize cost:  $\mathbf{u}^* = \arg \min_{\mathbf{u}} J, \quad J = \mathbf{u}^T H \mathbf{u} + 2x^T F^T \mathbf{u} + x^T G x$

differentiate w.r.t.  $\mathbf{u}$ :  $\nabla_{\mathbf{u}} J = 2H\mathbf{u} + 2Fx = 0$



$$\mathbf{u} = -H^{-1}Fx$$

$= \mathbf{u}^*$  if  $H$  is positive definite i.e. if  $H \succ 0$

Here  $H = \mathcal{C}^T \mathbf{Q} \mathcal{C} + \mathbf{R} \succ 0$  if:  $\begin{cases} R \succ 0 \ \& Q, P \succeq 0 & \text{or} \\ R \succeq 0 \ \& Q, P \succ 0 \ \& \mathcal{C} \text{ is full-rank} \end{cases}$



$(A, B)$  controllable

Receding horizon controller is linear state feedback:

$$u_k = -[I \ 0 \ \cdots \ 0] H^{-1} F x_k$$

is the closed loop response optimal? is it even stable?

# Unconstrained optimization

Minimize cost:  $\mathbf{u}^* = \arg \min_{\mathbf{u}} J, \quad J = \mathbf{u}^T H \mathbf{u} + 2x^T F^T \mathbf{u} + x^T G x$

differentiate w.r.t.  $\mathbf{u}$ :  $\nabla_{\mathbf{u}} J = 2H\mathbf{u} + 2Fx = 0$



$$\mathbf{u} = -H^{-1}Fx$$

$= \mathbf{u}^*$  if  $H$  is positive definite i.e. if  $H \succ 0$

Here  $H = \mathcal{C}^T \mathbf{Q} \mathcal{C} + \mathbf{R} \succ 0$  if: 
$$\begin{cases} R \succ 0 \ \& Q, P \succeq 0 & \text{or} \\ R \succeq 0 \ \& Q, P \succ 0 \ \& \mathcal{C} \text{ is full-rank} \end{cases}$$



$(A, B)$  controllable

Receding horizon controller is linear state feedback:

$$u_k = -[I \ 0 \ \cdots \ 0] H^{-1} F x_k$$

is the closed loop response optimal? is it even stable?

## Example

Model:  $A, B, C$  as before, cost:  $J_k = \sum_{i=0}^{N-1} (y_{i|k}^2 + 0.01u_{i|k}^2) + y_{N|k}^2$

► For  $N = 4$ :  $\mathbf{u}_k^* = -H^{-1}Fx_k = \begin{bmatrix} -4.36 & -18.7 \\ 1.64 & 1.24 \\ 1.41 & 3.00 \\ 0.59 & 1.83 \end{bmatrix} x_k$

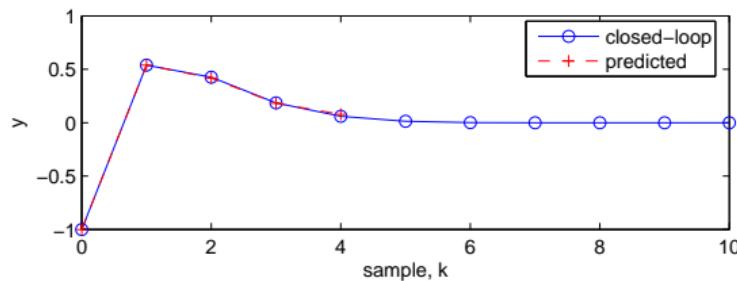
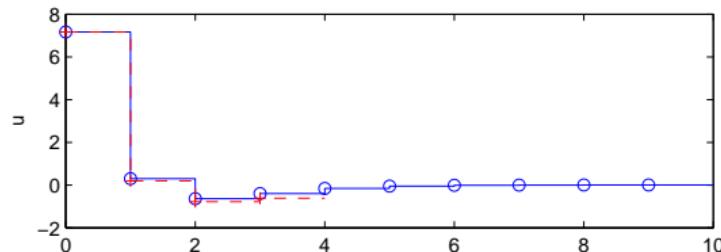
$$u_k = [-4.36 \quad -18.7] x_k$$

► For general  $N$ :  $u_k = K_N x_k$

	$N = 4$	$N = 3$	$N = 2$	$N = 1$
$\lambda(A + BK_N)$	$[-4.36 \quad -18.69]$ $0.29 \pm 0.17j$ stable	$[-3.80 \quad -16.98]$ $0.36 \pm 0.22j$ stable	$[1.22 \quad -3.95]$ $1.36, 0.38$ unstable	$[5.35 \quad 5.10]$ $2.15, 0.30$ unstable

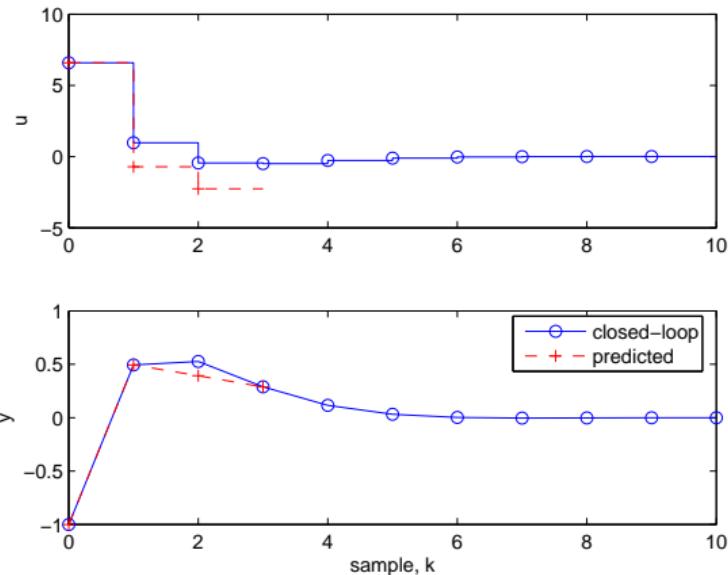
# Example

Horizon:  $N = 4$ ,  $x_0 = (0.5, -0.5)$



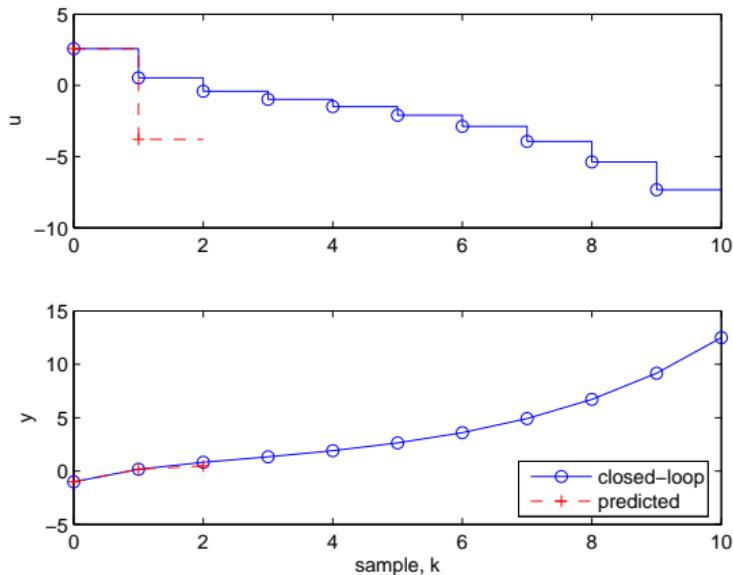
# Example

Horizon:  $N = 3$ ,  $x_0 = (0.5, -0.5)$



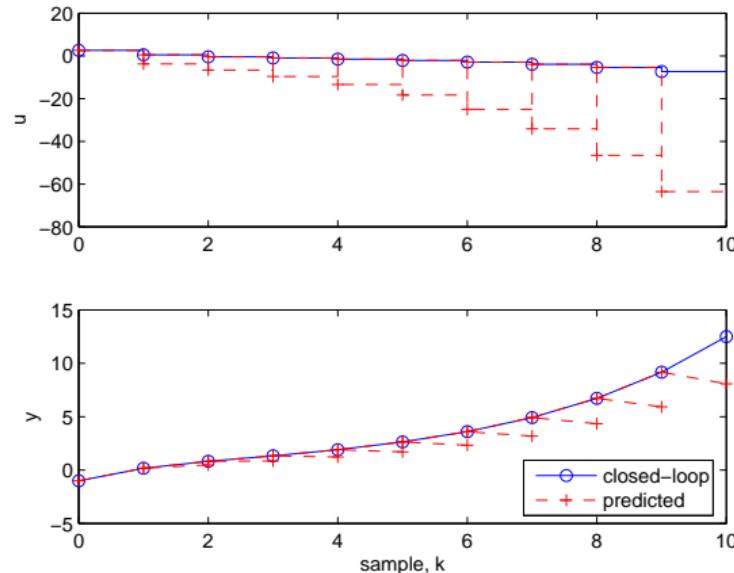
# Example

Horizon:  $N = 2$ ,  $x_0 = (0.5, -0.5)$



# Example

Horizon:  $N = 2$ ,  $x_0 = (0.5, -0.5)$



Observation: predicted and closed loop responses are different for small  $N$

# Receding horizon control

Why is this example unstable for  $N \leq 2$ ?

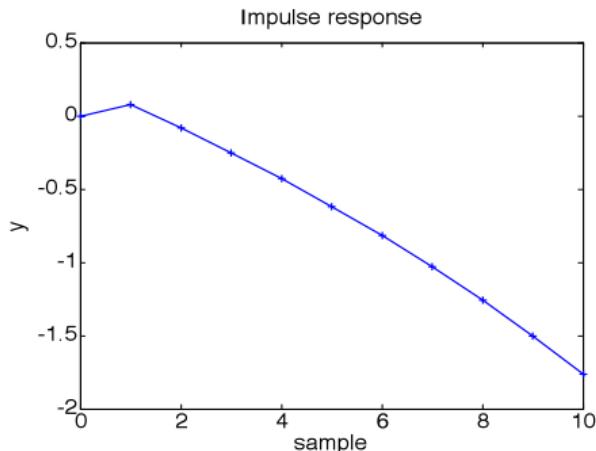
System is non-minimum phase



impulse response changes sign



hence short horizon causes instability



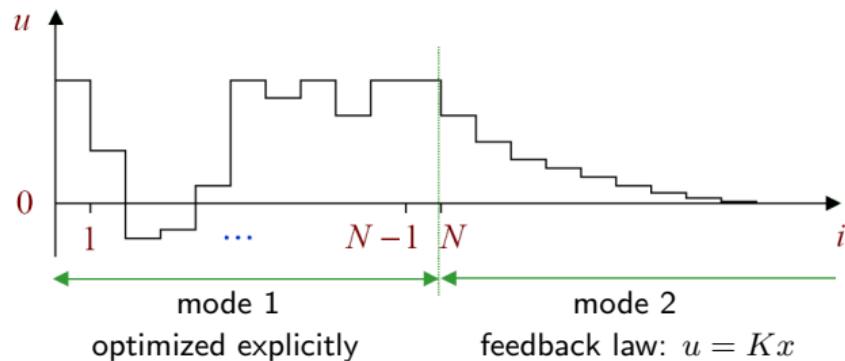
Solution:

- ★ use an **infinite** horizon cost
- ★ but keep a **finite** number of optimization variables in predictions

# Dual mode predictions

An infinite prediction horizon is possible with **dual mode** predictions:

$$u_{i|k} = \begin{cases} \text{optimization variables} & i = 0, \dots, N-1, \text{ mode 1} \\ Kx_{i|k} & i = N, N+1, \dots, \text{ mode 2} \end{cases}$$



Feedback gain  $K$ : stabilizing and determined offline

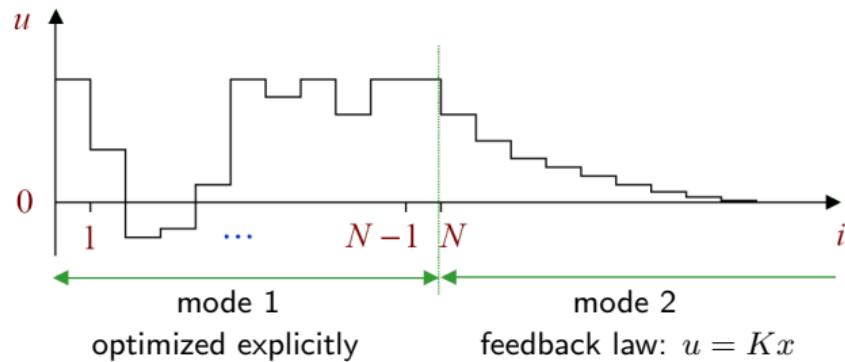
unconstrained LQ optimal for  $\sum_{i=0}^{\infty} (\|x_i\|_Q^2 + \|u_i\|_R^2)$

(usually)

# Dual mode predictions

An infinite prediction horizon is possible with **dual mode** predictions:

$$u_{i|k} = \begin{cases} \text{optimization variables} & i = 0, \dots, N-1, \text{ mode 1} \\ Kx_{i|k} & i = N, N+1, \dots, \text{ mode 2} \end{cases}$$



Feedback gain  $K$ : stabilizing and determined offline

unconstrained LQ optimal for  $\sum_{i=0}^{\infty} (\|x_i\|_Q^2 + \|u_i\|_R^2)$

(usually)

# Infinite horizon cost

If the predicted input sequence is

$$\{u_{0|k}, \dots, u_{N-1|k}, Kx_{N|k}, K\Phi x_{N|k}, \dots\}$$

then

$$\sum_{i=0}^{\infty} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2) = \sum_{i=0}^{N-1} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2) + \|x_{N|k}\|_P^2$$

where

$$P - (A + BK)^T P (A + BK) = Q + K^T R K$$

Lyapunov matrix equation (discrete time)

Note:

- ★ if  $Q + K^T R K \succ 0$ , then the solution  $P$  is unique and  $P \succ 0$
- ★ matlab:  

```
P = dlyap(Phi', RHS);  
Phi = A+B*K; RHS = Q+K'*R*K;
```
- ★  $P$  is the steady state Riccati equation solution if  $K$  is LQ-optimal

# Infinite horizon cost

If the predicted input sequence is

$$\{u_{0|k}, \dots, u_{N-1|k}, Kx_{N|k}, K\Phi x_{N|k}, \dots\}$$

then

$$\sum_{i=0}^{\infty} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2) = \sum_{i=0}^{N-1} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2) + \|x_{N|k}\|_P^2$$

where

$$P - (A + BK)^T P (A + BK) = Q + K^T R K$$

Lyapunov matrix equation (discrete time)

Note:

- ★ if  $Q + K^T R K \succ 0$ , then the solution  $P$  is unique and  $P \succ 0$
- ★ matlab:  

```
P = dlyap(Phi', RHS);  
Phi = A+B*K; RHS = Q+K'*R*K;
```
- ★  $P$  is the steady state Riccati equation solution if  $K$  is LQ-optimal

## Infinite horizon cost

Proof that the predicted cost over the mode 2 horizon is  $\|x_{N|k}\|_P^2$ :

Let  $J^\infty(x_0) = \sum_{i=0}^{\infty} (\|x_i\|_Q^2 + \|u_i\|_R^2)$ , with  $u_i = Kx_i$ ,  $x_{i+1} = \Phi x_i \forall i$ ,

- then 
$$\begin{aligned} J^\infty(x_0) &= \sum_{i=0}^{\infty} (\|\Phi^i x_0\|_Q^2 + \|K\Phi^i x_0\|_R^2) \\ &= x_0^T \underbrace{\left[ \sum_{i=0}^{\infty} (\Phi^i)^T (Q + K^T R K) \Phi^i \right]}_{=S} x_0 \end{aligned}$$

- but 
$$\begin{aligned} \Phi^T S \Phi &= \sum_{i=1}^{\infty} (\Phi^i)^T (Q + K^T R K) \Phi^i \\ &= S - (Q + K^T R K) \end{aligned}$$

so if  $\Phi = A + BK$ , then  $S = P$  and  $J^\infty(x_{N|k}) = \|x_{N|k}\|_P^2$

# Connection with LQ optimal control

Predicted cost:

$$J(x_k, (u_{0|k}, \dots, u_{N-1|k})) = \sum_{i=0}^{N-1} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2) + \|x_{N|k}\|_P^2$$

where  $P - (A + BK)^T P (A + BK) = Q + K^T R K$

and  $K$  = LQ-optimal



$$u_{0|k}^* = Kx_k \text{ where } \mathbf{u}_k^* = \arg \min_{\mathbf{u}} J(x_k, \mathbf{u}) = (u_{0|k}^*, \dots, u_{N-1|k}^*)$$



The Bellman principle of optimality implies:

$$\{u_{0|k}, u_{1,k}, \dots\} \text{ optimal} \iff \begin{cases} \{u_{0|k}, \dots, u_{N-1|k}\} \text{ optimal} \\ \text{and } K \text{ LQ-optimal} \end{cases}$$

## Connection with LQ optimal control – example

- Model parameters  $(A, B, C)$  as before

LQ optimal gain for  $Q = C^T C$ ,  $R = 0.01$ :  $K = \begin{bmatrix} -4.36 & -18.74 \end{bmatrix}$

Lyapunov equation solution:  $P = \begin{bmatrix} 3.92 & 4.83 \\ & 13.86 \end{bmatrix}$

- Cost matrices for  $N = 4$ :

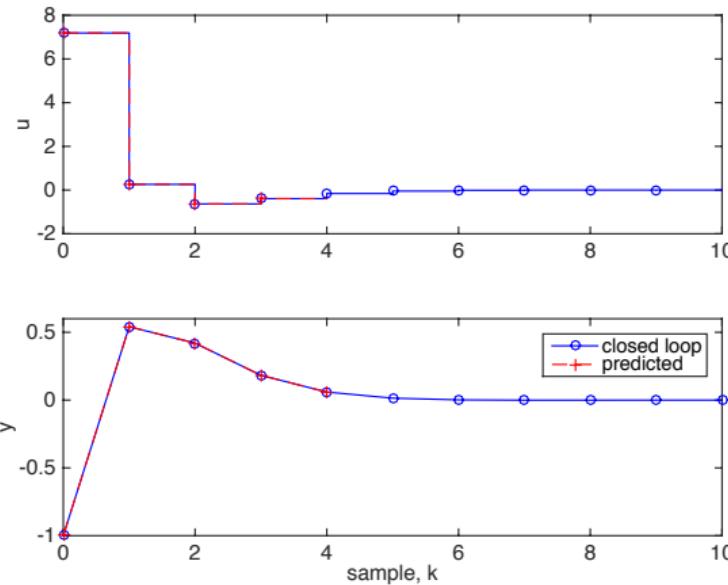
$$H = \begin{bmatrix} 1.44 & 0.98 & 0.59 & 0.26 \\ & 0.72 & 0.44 & 0.20 \\ & & 0.30 & 0.14 \\ & & & 0.096 \end{bmatrix} \quad F = \begin{bmatrix} 3.67 & 23.9 \\ 2.37 & 16.2 \\ 1.36 & 9.50 \\ 0.556 & 4.18 \end{bmatrix} \quad G = \begin{bmatrix} 13.8 & 66.7 \\ & 413 \end{bmatrix}$$

- Predictive control law:  $u_k = -[1 \ 0 \ 0 \ 0] H^{-1} F x_k$   
 $= \begin{bmatrix} -4.35 & -18.74 \end{bmatrix} x_k$

(identical to the LQ optimal controller)

## Connection with LQ optimal control – example

- ▶ Response for  $N = 4$ ,  $x_0 = (0.5, -0.5)$



Infinite horizon cost  
no constraints }  $\implies$  identical predicted and closed loop responses

# Dual mode predictions

**Aside:** Pre-stabilized dual mode predictions with better numerical stability

## ▷ Inputs

$$\begin{array}{ll} \text{mode 1} & u_{i|k} = Kx_{i|k} + c_{i|k}, \quad i = 0, 1, \dots, N-1 \\ \text{mode 2} & u_{i|k} = Kx_{i|k}, \quad i = N, N+1, \dots \end{array}$$

## ▷ Dynamics

$$\begin{array}{ll} \text{mode 1} & x_{i+1|k} = \Phi x_{i|k} + Bc_{i|k}, \quad i = 0, 1, \dots, N-1 \\ \text{mode 2} & x_{i+1|k} = \Phi x_{i|k}, \quad i = N, N+1, \dots \end{array}$$

where  $(c_{0|k}, \dots, c_{N-1|k})$  are optimization variables

# Dual mode predictions

**Aside:** Pre-stabilized dual mode predictions with better numerical stability

- ▷ Vectorized form:  $\mathbf{x}_k = \mathcal{M}x_k + \mathcal{C}\mathbf{c}_k$

$$\mathbf{x}_k := \begin{bmatrix} x_{1|k} \\ \vdots \\ x_{N|k} \end{bmatrix}, \quad \mathbf{c}_k := \begin{bmatrix} c_{0|k} \\ \vdots \\ c_{N-1|k} \end{bmatrix}$$

$$\mathcal{M} = \begin{bmatrix} \Phi \\ \Phi^2 \\ \vdots \\ \Phi^N \end{bmatrix}, \quad \mathcal{C} = \begin{bmatrix} B & 0 & \cdots & 0 \\ \Phi B & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \Phi^{N-1}B & \Phi^{N-2}B & \cdots & B \end{bmatrix}$$

- ▷ Cost:  $J(x_k, (u_{0|k}, \dots, u_{N-1|k})) = J(x_k, \mathbf{c}_k)$

# Input and state constraints

Infinite horizon unconstrained MPC = LQ optimal control

but MPC can also handle constraints

Consider constraints applied to mode 1 predictions:

- \* input constraints:  $\underline{u} \leq u_{i|k} \leq \bar{u}, \quad i = 0, \dots, N - 1$

$$\iff \begin{bmatrix} I \\ -I \end{bmatrix} \mathbf{u}_k \leq \begin{bmatrix} \bar{\mathbf{u}} \\ -\underline{\mathbf{u}} \end{bmatrix} \quad \text{where} \quad \bar{\mathbf{u}} = [\bar{u}^T \quad \dots \quad \bar{u}^T]^T$$
$$\underline{\mathbf{u}} = [\underline{u}^T \quad \dots \quad \underline{u}^T]^T$$

- \* state constraints:  $\underline{x} \leq x_{i|k} \leq \bar{x}, \quad i = 1, \dots, N$

$$\iff \begin{bmatrix} \mathcal{C}_i \\ -\mathcal{C}_i \end{bmatrix} \mathbf{u}_k \leq \begin{bmatrix} \bar{x} \\ -\underline{x} \end{bmatrix} + \begin{bmatrix} -A^i \\ A^i \end{bmatrix} x_k, \quad i = 1, \dots, N$$

## Input and state constraints

Constraints on mode 1 predictions can be expressed

$$A_c \mathbf{u}_k \leq b_c + B_c x_k$$

where  $A_c, B_c, b_c$  can be computed offline since model is time-invariant

The online optimization is a quadratic program (QP):

$$\begin{aligned} & \underset{\mathbf{u}}{\text{minimize}} && \mathbf{u}^T H \mathbf{u} + 2x_k^T F^T \mathbf{u} \\ & \text{subject to} && A_c \mathbf{u} \leq b_c + B_c x_k \end{aligned}$$

which is a convex optimization problem with a unique solution if

$$H = \mathcal{C}^T \mathcal{Q} \mathcal{C} + \mathbf{R} \text{ is positive definite}$$

## QP solvers: (a) Active set

Consider the QP:  $\mathbf{u}^* = \arg \min_{\mathbf{u}} \mathbf{u}^T H \mathbf{u} + 2f^T \mathbf{u}$   
subject to  $A\mathbf{u} \leq b$

and let  $(A_i, b_i) = i$ th row/element of  $(A, b)$

- ▷ Individual constraints are **active** or **inactive**

active	inactive
$A_i \mathbf{u}^* = b_i, \forall i \in \mathcal{I}$ $b_i$ affects solution	$A_i \mathbf{u}^* \leq b_i, \forall i \notin \mathcal{I}$ $b_i$ does not affect solution

- ▷ Equality constraint problem:  $\mathbf{u}^* = \arg \min_{\mathbf{u}} \mathbf{u}^T H \mathbf{u} + 2f^T \mathbf{u}$   
subject to  $A_i \mathbf{u} = b_i, \forall i \in \mathcal{I}$

- ▷ Solve QP by searching for  $\mathcal{I}$ 
  - \* one equality constraint problem solved at each iteration
  - \* optimality conditions (KKT conditions) identify solution

## QP solvers: (a) Active set

Consider the QP:  $\mathbf{u}^* = \arg \min_{\mathbf{u}} \mathbf{u}^T H \mathbf{u} + 2f^T \mathbf{u}$   
subject to  $A\mathbf{u} \leq b$

and let  $(A_i, b_i) = i$ th row/element of  $(A, b)$

- ▷ Individual constraints are **active** or **inactive**

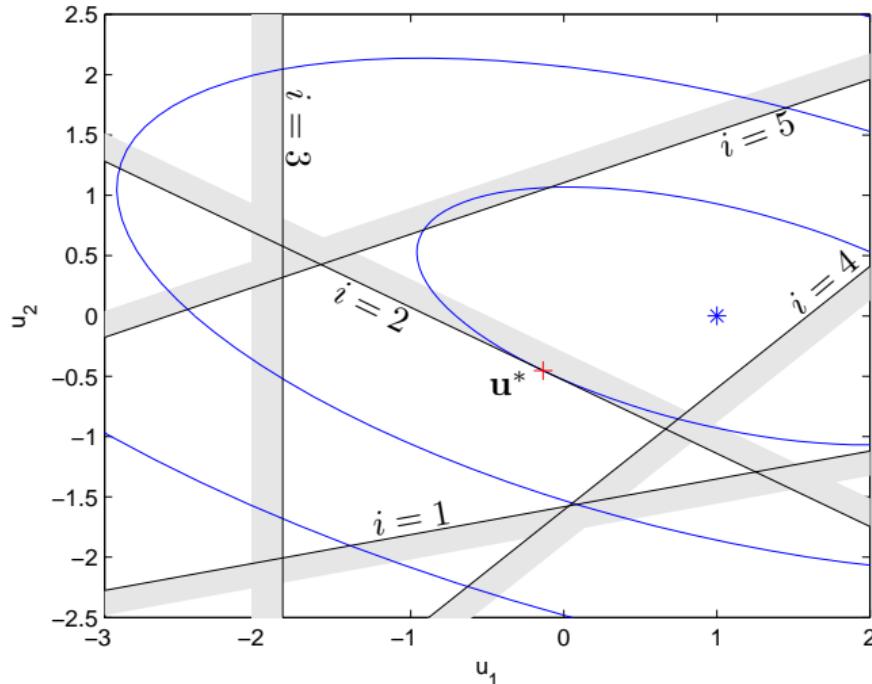
active	inactive
$A_i \mathbf{u}^* = b_i, \forall i \in \mathcal{I}$ $b_i$ affects solution	$A_i \mathbf{u}^* \leq b_i, \forall i \notin \mathcal{I}$ $b_i$ does not affect solution

- ▷ Equality constraint problem:  $\mathbf{u}^* = \arg \min_{\mathbf{u}} \mathbf{u}^T H \mathbf{u} + 2f^T \mathbf{u}$   
subject to  $A_i \mathbf{u} = b_i, \forall i \in \mathcal{I}$

- ▷ Solve QP by searching for  $\mathcal{I}$

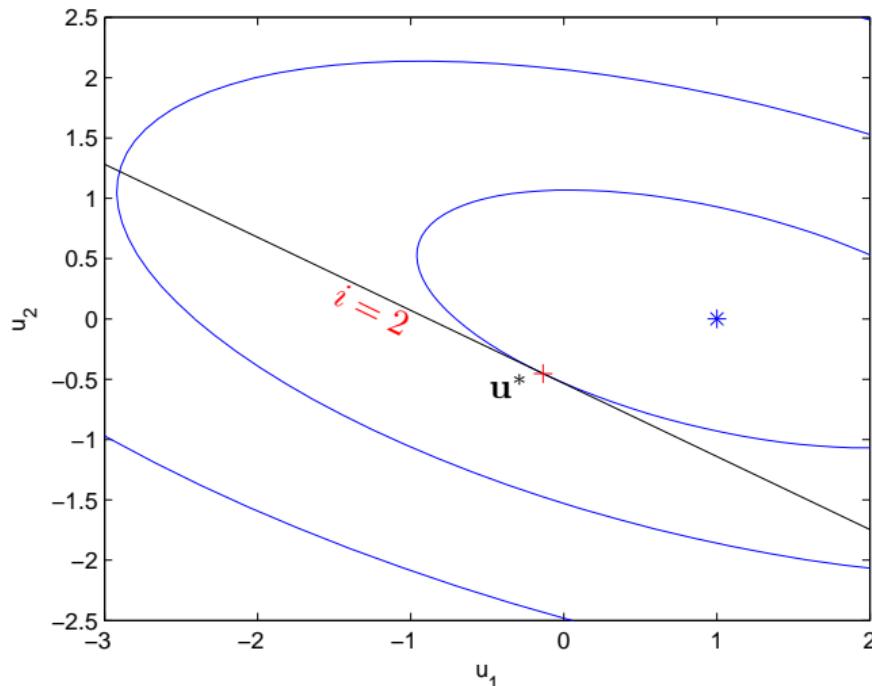
- ★ one equality constraint problem solved at each iteration
- ★ optimality conditions (**KKT conditions**) identify solution

## Active constraints – example



A QP problem with 5 inequality constraints  
active set at solution:  $\mathcal{I} = \{2\}$

## Active constraints – example



An equivalent equality constraint problem

## QP solvers: (a) Active set

- ▷ Computation:

$O(N^3 n_u^3)$  additions/multiplications per iteration (conservative estimate)

upper bound on number of iterations is exponential in problem size

- ▷ At each iteration choose trial active set using:  
cost gradient  
constraint sensitivities



number of iterations needed is often small in practice

- ▷ In MPC  $\mathbf{u}_k^* = \mathbf{u}^*(x_k)$  and  $\mathcal{I}_k = \mathcal{I}(x_k)$

hence initialize solver at time  $k$  using the solution computed at  $k - 1$

## QP solvers: (b) Interior point

- ▷ Solve an unconstrained problem at each iteration:

$$\mathbf{u}(\mu) = \min_{\mathbf{u}} \mu (\mathbf{u}^T H \mathbf{u} + 2f^T \mathbf{u}) + \phi(\mathbf{u})$$

where

$$\begin{aligned}\phi(\mathbf{u}) &= \text{barrier function} \quad (\phi \rightarrow \infty \text{ at constraints}) \\ \mathbf{u} &\rightarrow \mathbf{u}^* \text{ as } \mu \rightarrow \infty\end{aligned}$$

Increase  $\mu$  until  $\phi(\mathbf{u}^*) > 1/\epsilon$       ( $\epsilon$  = user-defined tolerance)

- ▷ # operations per iterations is constant, e.g.  $O(N^3 n_u^3)$   
# iterations for given  $\epsilon$  is polynomial in problem size



Computational advantages for large-scale problems

e.g. # variables  $> 10^2$ , # constraints  $> 10^3$

- ▷ No general method for initializing at solution estimate

## QP solvers: (b) Interior point

- ▷ Solve an unconstrained problem at each iteration:

$$\mathbf{u}(\mu) = \min_{\mathbf{u}} \mu (\mathbf{u}^T H \mathbf{u} + 2f^T \mathbf{u}) + \phi(\mathbf{u})$$

where

$$\begin{aligned}\phi(\mathbf{u}) &= \text{barrier function} \quad (\phi \rightarrow \infty \text{ at constraints}) \\ \mathbf{u} &\rightarrow \mathbf{u}^* \text{ as } \mu \rightarrow \infty\end{aligned}$$

Increase  $\mu$  until  $\phi(\mathbf{u}^*) > 1/\epsilon$       ( $\epsilon$  = user-defined tolerance)

- ▷ # operations per iterations is constant, e.g.  $O(N^3 n_u^3)$   
# iterations for given  $\epsilon$  is polynomial in problem size

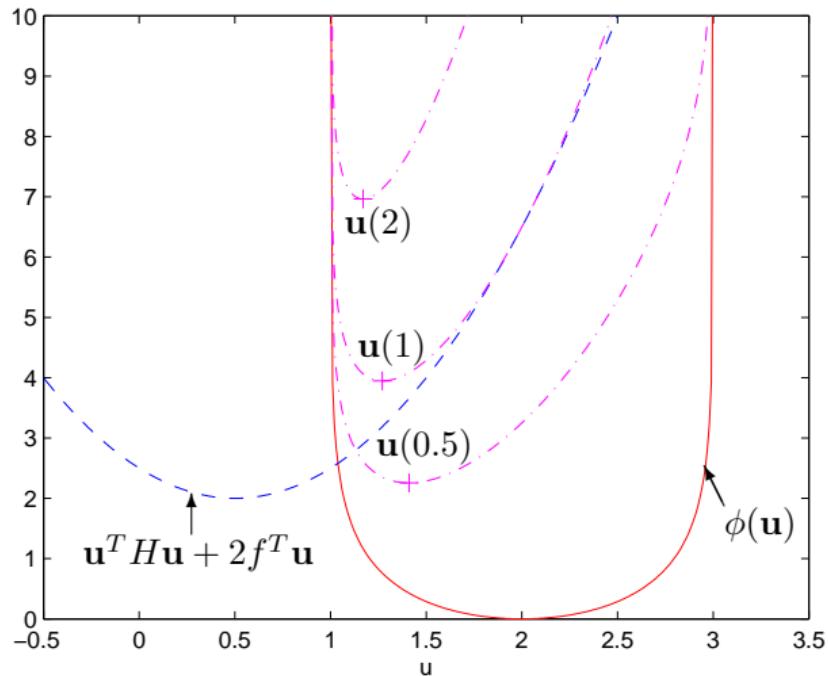


Computational advantages for large-scale problems

e.g. # variables  $> 10^2$ , # constraints  $> 10^3$

- ▷ No general method for initializing at solution estimate

## Interior point method – example



$\mathbf{u}(\mu) \rightarrow \mathbf{u}^* = 1$  as  $\mu \rightarrow \infty$

but  $\min_{\mathbf{u}} \mu(\mathbf{u}^T H \mathbf{u} + 2f^T \mathbf{u}) + \phi(\mathbf{u})$  becomes ill-conditioned as  $\mu \rightarrow \infty$

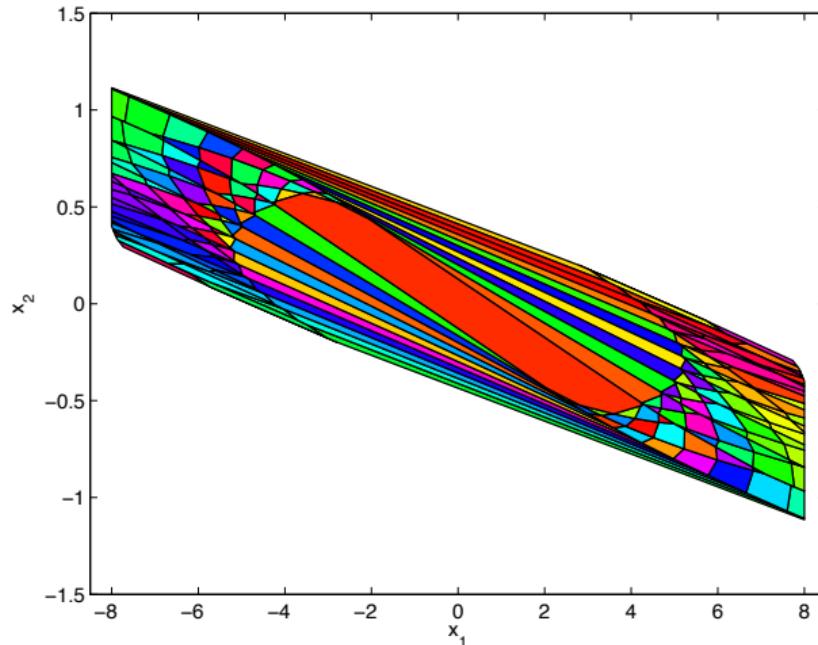
## QP solvers: (c) Multiparametric

Let  $\mathbf{u}^*(\mathbf{x}) = \arg \min_{\mathbf{u}} \mathbf{u}^T H \mathbf{u} + 2\mathbf{x}^T F^T \mathbf{u}$   
subject to  $A\mathbf{u} \leq b + B\mathbf{x}$

then:

- ★  $\mathbf{u}^*$  is a continuous function of  $x$
  - ★  $\mathbf{u}^*(x) = K_j x + k_j$  for all  $x$  in a polytopic set  $\mathcal{X}_j$
- 
- ▷ In principle each  $K_j$ ,  $k_j$  and  $\mathcal{X}_j$  can be determined offline
  - ▷ But number of sets  $\mathcal{X}_j$  is usually large (depends exponentially on problem size)  
so online determination of  $j$  such that  $x_k \in \mathcal{X}_j$  is difficult

# Multiparametric QP – example



Model:  $(A, B, C)$  as before,

cost:  $Q = C^T C$ ,  $R = 1$ , horizon:  $N = 10$

constraints:  $-1 \leq u \leq 1$ ,  $-\mathbf{1} \leq x/8 \leq \mathbf{1}$

# Summary

- ▷ Predicted control inputs:  $\mathbf{u}_k = \begin{bmatrix} u_{0|k} \\ \vdots \\ u_{N-1|k} \end{bmatrix}$
- and states:  $\mathbf{x}_k = \begin{bmatrix} x_{1|k} \\ \vdots \\ x_{N|k} \end{bmatrix} = \mathcal{M}\mathbf{x}_k + \mathcal{C}\mathbf{u}_k$
- ▷ Predicted cost: 
$$\begin{aligned} J(x_k, \mathbf{u}_k) &= \sum_{i=0}^{N-1} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2) + \|x_{N|k}\|_P^2 \\ &= \mathbf{u}_k^T H \mathbf{u}_k + 2x_k^T F^T \mathbf{u}_k + x_k^T G x_k \end{aligned}$$
- ▷ Online optimization of cost subject to linear state and input constraints is a QP problem:
$$\begin{aligned} &\underset{\mathbf{u}}{\text{minimize}} \quad \mathbf{u}^T H \mathbf{u} + 2x_k^T F^T \mathbf{u} \\ &\text{subject to} \quad A_c \mathbf{u} \leq b_c + B_c x_k \end{aligned}$$

## Lecture 3

# Closed loop properties of MPC

# Closed loop properties of MPC

- Review: infinite horizon cost
- Infinite horizon predictive control with constraints
- Closed loop stability
- Constraint-checking horizon
- Connection with constrained optimal control

## Review: infinite horizon cost

Short prediction horizons cause poor performance and instability, so

- ★ use an infinite horizon cost:  $J(x_k, \mathbf{u}_k) = \sum_{i=0}^{\infty} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2)$
- ★ keep optimization finite-dimensional by using **dual mode predictions**:  
$$u_{i|k} = \begin{cases} \text{optimization variables} & i = 0, \dots, N-1, \quad \text{mode 1} \\ Kx_{i|k} & i = N, N+1, \dots \quad \text{mode 2} \end{cases}$$

mode 1:  $\mathbf{u}_k = \begin{bmatrix} u_{0|k} \\ \vdots \\ u_{N-1|k} \end{bmatrix}$        $\mathbf{u}_k$  optimized online

mode 2:  $u_{i|k} = Kx_{i|k}$        $K$  chosen offline

## Review: infinite horizon cost

- ▷ Cost for mode 2:  $\sum_{i=N}^{\infty} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2) = \|x_{N|k}\|_P^2$

$P$  is the solution of the Lyapunov equation

$$P - (A + BK)^T P (A + BK) = Q + K^T R K$$

- ▷ Infinite horizon cost:

$$\begin{aligned} J(x_k, \mathbf{u}_k) &= \sum_{i=0}^{N-1} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2) + \|x_{N|k}\|_P^2 \\ &= \mathbf{u}_k^T H \mathbf{u}_k + 2x_k^T F^T \mathbf{u}_k + x_k^T G x_k \end{aligned}$$

## Review: MPC online optimization

- ▷ Unconstrained optimization:  $\nabla_{\mathbf{u}} J(x, \mathbf{u}^*) = 2H\mathbf{u}^* + 2Fx = 0$ , so

$$\mathbf{u}^*(x) = -H^{-1}Fx$$

⇒ **linear controller:**  $u_k = K_{\text{MPC}}x_k$

$K_{\text{MPC}}$  = LQ-optimal if  $K$  = LQ-optimal (in mode 2)

- ▷ Constrained optimization:

$$\mathbf{u}^*(x) = \arg \min_{\mathbf{u}} \quad \mathbf{u}^T H \mathbf{u} + 2x^T F^T \mathbf{u}$$

subject to  $A_c \mathbf{u} \leq b_c + B_c x$

⇒ **nonlinear controller:**  $u_k = K_{\text{MPC}}(x_k)$

# Constrained MPC – example

▷ Plant model:  $x_{k+1} = Ax_k + Bu_k, \quad y_k = Cx_k$

$$A = \begin{bmatrix} 1.1 & 2 \\ 0 & 0.95 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0.0787 \end{bmatrix}, \quad C = [-1 \quad 1]$$

Constraints:  $-1 \leq u_k \leq 1$

▷ MPC optimization (constraints applied only to mode 1 predictions):

$$\underset{\mathbf{u}}{\text{minimize}} \quad \sum_{i=0}^{N-1} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2) + \|x_{N|k}\|_P^2$$

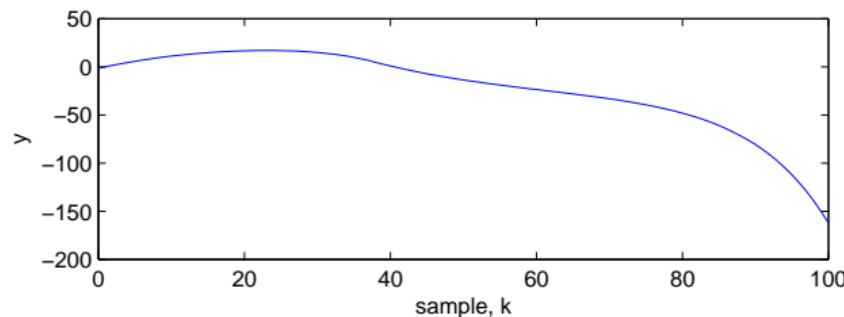
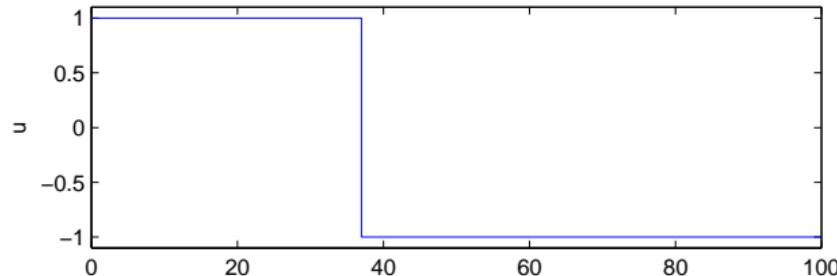
subject to  $-1 \leq u_{i|k} \leq 1, \quad i = 0, \dots, N - 1$

$$Q = C^T C, \quad R = 0.01, \quad N = 2$$

... performance? stability?

# Constrained MPC – example

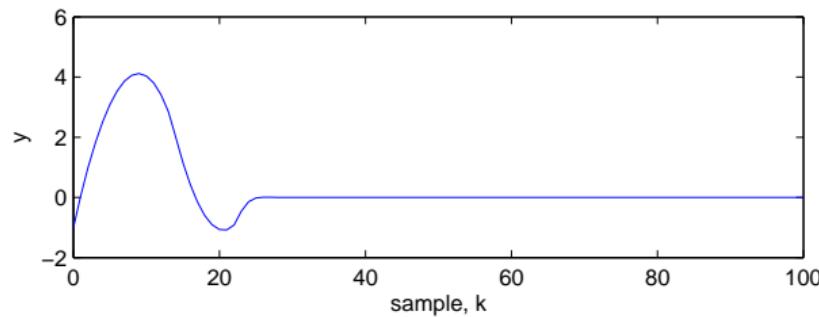
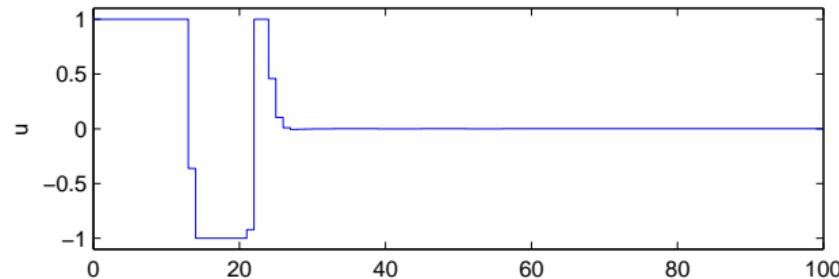
Closed loop response for  $x_0 = (0.8, -0.8)$



unstable

# Constrained MPC – example

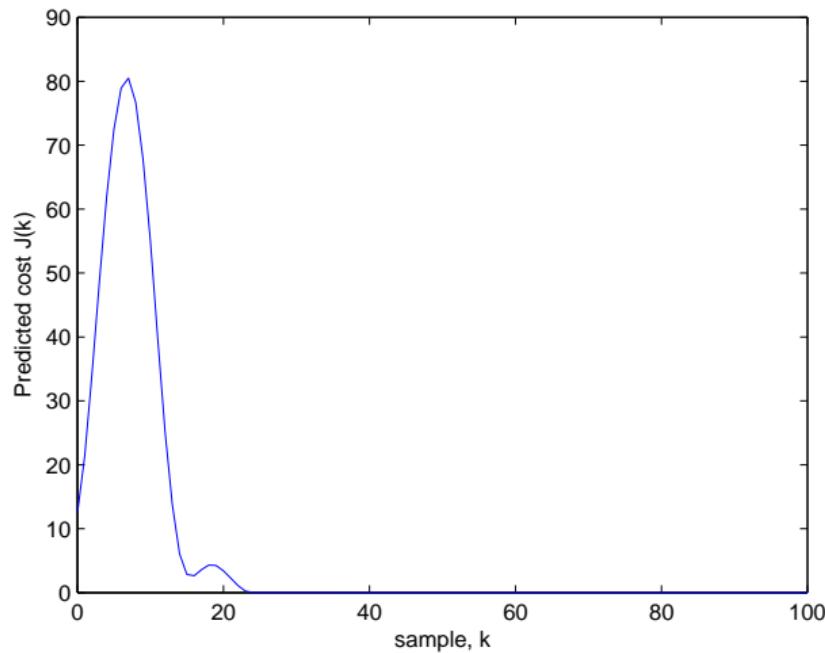
Closed loop response for  $x_0 = (0.5, -0.5)$



stable, but ...

# Constrained MPC – example

Optimal predicted cost  $x_0 = (0.5, -0.5)$



... increasing  $J_k \implies$  closed loop response does not follow predicted trajectory

# Stability analysis

How can we guarantee the closed loop stability of MPC?

- (a). Show that a Lyapunov function exists demonstrating stability
- (b). Ensure that optimization feasible is at each time  $k = 0, 1, \dots$

- ▷ For Lyapunov stability analysis:
  - ★ consider first the unconstrained problem
  - ★ use predicted cost as a trial Lyapunov function
- ▷ Guarantee feasibility of the MPC optimization recursively
  - by ensuring that feasibility at time  $k$  implies feasibility at  $k + 1$

# Stability analysis

How can we guarantee the closed loop stability of MPC?

- (a). Show that a Lyapunov function exists demonstrating stability
- (b). Ensure that optimization feasible is at each time  $k = 0, 1, \dots$

- ▷ For Lyapunov stability analysis:
  - ★ consider first the unconstrained problem
  - ★ use predicted cost as a trial Lyapunov function
- ▷ Guarantee feasibility of the MPC optimization recursively
  - by ensuring that feasibility at time  $k$  implies feasibility at  $k + 1$

# Discrete time Lyapunov stability

Consider the system  $x_{k+1} = f(x_k)$ , with  $f(0) = 0$

▷ Definition:  $x = 0$  is a **stable** equilibrium point if

$\max_k \|x_k\|$  can be made arbitrarily small  
by making  $x_0$  sufficiently small

▷ If continuously differentiable  $V(x)$  exists with

- (i).  $V(x)$  is positive definite and
- (ii).  $V(x_{k+1}) - V(x_k) \leq 0$

then  $x = 0$  is a stable equilibrium point

# Discrete time Lyapunov stability

Consider the system  $x_{k+1} = f(x_k)$ , with  $f(0) = 0$

- ▷ Definition:  $x = 0$  is a **stable** equilibrium point if
  - for all  $R > 0$  there exists  $r$  such that
$$\|x_0\| < r \implies \|x_k\| < R \text{ for all } k$$

- ▷ If continuously differentiable  $V(x)$  exists with
  - (i).  $V(x)$  is positive definite and
  - (ii).  $V(x_{k+1}) - V(x_k) \leq 0$

then  $x = 0$  is a stable equilibrium point

# Discrete time Lyapunov stability

Consider the system  $x_{k+1} = f(x_k)$ , with  $f(0) = 0$

▷ Definition:  $x = 0$  is an **asymptotically stable** equilibrium point if

- (i).  $x = 0$  is stable and
- (ii).  $r$  exists such that  $\|x_0\| < r \implies \lim_{k \rightarrow \infty} x_k = 0$

▷ If continuously differentiable  $V(x)$  exists with

- (i).  $V(x)$  is positive definite and
- (ii).  $V(x_{k+1}) - V(x_k) < 0$  whenever  $x_k \neq 0$

then  $x = 0$  is an asymptotically stable equilibrium point

# Lyapunov stability

Trial Lyapunov function:

$$J^*(x_k) = J(x_k, \mathbf{u}_k^*)$$

$$\text{where } J(x_k, \mathbf{u}_k) = \sum_{i=0}^{\infty} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2)$$

★  $J^*(x)$  is positive definite if:

- (a).  $R \succeq 0$  and  $Q \succ 0$ , or
- (b).  $R \succ 0$  and  $Q \succeq 0$  and  $(A, Q)$  is observable

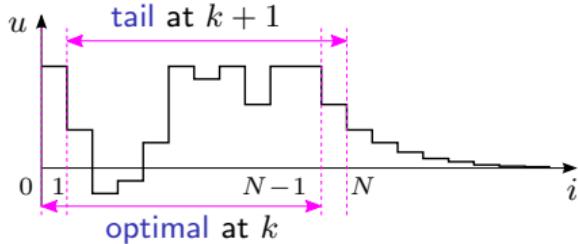
since then  $J^*(x_k) \geq 0$  and  $J^*(x_k) = 0$  if and only if  $x_k = 0$

★  $J^*(x)$  is continuously differentiable

...from analysis of MPC optimization as a multiparametric QP

# Lyapunov stability

Construct a bound on  $J^*(x_{k+1}) - J^*(x_k)$  using the “tail” of the optimal prediction at time  $k$



Optimal predicted sequences at time  $k$ :

$$\mathbf{u}_k^* = \begin{bmatrix} u_{0|k}^* \\ u_{1|k}^* \\ \vdots \\ u_{N-1|k}^* \\ Kx_{N|k}^* \\ \vdots \end{bmatrix} \quad \mathbf{x}_k^* = \begin{bmatrix} x_{1|k}^* \\ x_{2|k}^* \\ \vdots \\ x_{N|k}^* \\ \Phi x_{N|k}^* \\ \vdots \end{bmatrix}$$
$$(\Phi = A + BK)$$

$$\text{optimal at } k : \quad J^*(x_k) = J(x_k, \mathbf{u}_k^*)$$

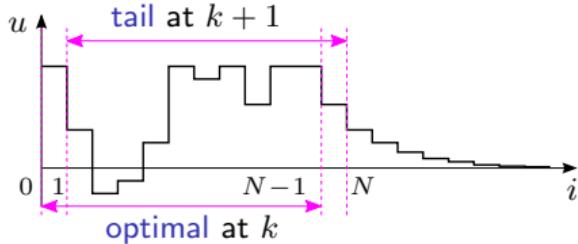
$$= \sum_{i=0}^{\infty} (\|x_{i|k}^*\|_Q^2 + \|u_{i|k}^*\|_R^2)$$

$$\text{tail at } k+1 : \quad \tilde{J}(x_{k+1}) = J(x_{k+1}, \tilde{\mathbf{u}}_{k+1})$$

$$= \sum_{i=1}^{\infty} (\|x_{i|k}^*\|_Q^2 + \|u_{i|k}^*\|_R^2)$$

# Lyapunov stability

Construct a bound on  $J^*(x_{k+1}) - J^*(x_k)$  using the “tail” of the optimal prediction at time  $k$



Tail sequences at time  $k + 1$ :

$$\tilde{\mathbf{u}}_{k+1} = \begin{bmatrix} u_{1|k}^* \\ \vdots \\ u_{N-1|k}^* \\ Kx_{N|k}^* \\ K\Phi x_{N|k}^* \\ \vdots \end{bmatrix} \quad \tilde{\mathbf{x}}_{k+1} = \begin{bmatrix} x_{2|k}^* \\ \vdots \\ x_{N|k}^* \\ \Phi x_{N|k}^* \\ \Phi^2 x_{N|k}^* \\ \vdots \end{bmatrix}$$

$(\Phi = A + BK)$

$$\text{optimal at } k : \quad J^*(x_k) = J(x_k, \mathbf{u}_k^*)$$

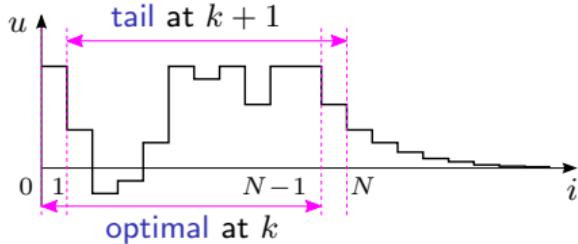
$$= \sum_{i=0}^{\infty} (\|x_{i|k}^*\|_Q^2 + \|u_{i|k}^*\|_R^2)$$

$$\text{tail at } k + 1 : \quad \tilde{J}(x_{k+1}) = J(x_{k+1}, \tilde{\mathbf{u}}_{k+1})$$

$$= \sum_{i=1}^{\infty} (\|x_{i|k}^*\|_Q^2 + \|u_{i|k}^*\|_R^2)$$

# Lyapunov stability

Construct a bound on  $J^*(x_{k+1}) - J^*(x_k)$  using the “tail” of the optimal prediction at time  $k$



Tail sequences at time  $k + 1$ :

$$\tilde{\mathbf{u}}_{k+1} = \begin{bmatrix} u_{1|k}^* \\ \vdots \\ u_{N-1|k}^* \\ Kx_{N|k}^* \\ K\Phi x_{N|k}^* \\ \vdots \end{bmatrix} \quad \tilde{\mathbf{x}}_{k+1} = \begin{bmatrix} x_{2|k}^* \\ \vdots \\ x_{N|k}^* \\ \Phi x_{N|k}^* \\ \Phi^2 x_{N|k}^* \\ \vdots \end{bmatrix}$$

$(\Phi = A + BK)$

optimal at  $k$  :  $J^*(x_k) = J(x_k, \mathbf{u}_k^*) = \sum_{i=0}^{\infty} (\|x_{i|k}^*\|_Q^2 + \|u_{i|k}^*\|_R^2)$

tail at  $k + 1$  :  $\tilde{J}(x_{k+1}) = J(x_{k+1}, \tilde{\mathbf{u}}_{k+1}) = \sum_{i=1}^{\infty} (\|x_{i|k}^*\|_Q^2 + \|u_{i|k}^*\|_R^2)$

# Lyapunov stability

Construct a bound on  $J^*(x_{k+1}) - J^*(x_k)$  using the “tail” of the optimal prediction at time  $k$

Predicted cost for the tail:

$$\tilde{J}(x_{k+1}) = J^*(x_k) - \|x_k\|_Q^2 - \|u_k\|_R^2$$

but  $\tilde{\mathbf{u}}_{k+1}$  is suboptimal at time  $k + 1$ , so

$$J^*(x_{k+1}) \leq \tilde{J}(x_{k+1})$$

Therefore

$$J^*(x_{k+1}) \leq J^*(x_k) - \|x_k\|_Q^2 - \|u_k\|_R^2$$

# Lyapunov stability

The bound  $J^*(x_{k+1}) - J^*(x_k) \leq -\|x_k\|_Q^2 - \|u_k\|_R^2$  implies:

- ① the closed loop cost cannot exceed the initial predicted cost, since summing both sides over all  $k \geq 0$  gives

$$\sum_{k=0}^{\infty} (\|x_k\|_Q^2 + \|u_k\|_R^2) \leq J^*(x_0)$$

- ②  $x = 0$  is asymptotically stable
  - \* if  $R \succeq 0$  and  $Q \succ 0$ , this follows from Lyapunov's direct method
  - \* if  $R \succ 0$ ,  $Q \succeq 0$  and  $(A, Q)$  observable, this follows from:
    - (a). stability of  $x = 0$        $\Leftarrow$  Lyapunov's direct method
    - (b).  $\lim_{k \rightarrow \infty} (\|x_k\|_Q^2 + \|u_k\|_R^2) = 0$     $\Leftarrow \sum_{k=0}^{\infty} (\|x_k\|_Q^2 + \|u_k\|_R^2) < \infty$

# Stability analysis

How can we guarantee the closed loop stability of MPC?

- (a). Show that a Lyapunov function exists demonstrating stability
- (b). Ensure that optimization feasible is at each time  $k = 0, 1, \dots$

- ▷ For Lyapunov stability analysis:
  - ★ consider first the unconstrained problem
  - ★ use predicted cost as a trial Lyapunov function
- ▷ Guarantee feasibility of the MPC optimization recursively
  - by ensuring that feasibility at time  $k \implies$  feasibility at  $k + 1$

# Stability analysis

How can we guarantee the closed loop stability of MPC?

- (a). Show that a Lyapunov function exists demonstrating stability
- (b). Ensure that optimization feasible is at each time  $k = 0, 1, \dots$

- ▷ For Lyapunov stability analysis:
  - ★ consider first the unconstrained problem
  - ★ use predicted cost as a trial Lyapunov function
- ▷ Guarantee feasibility of the MPC optimization recursively
  - by ensuring that feasibility at time  $k \implies$  feasibility at  $k + 1$

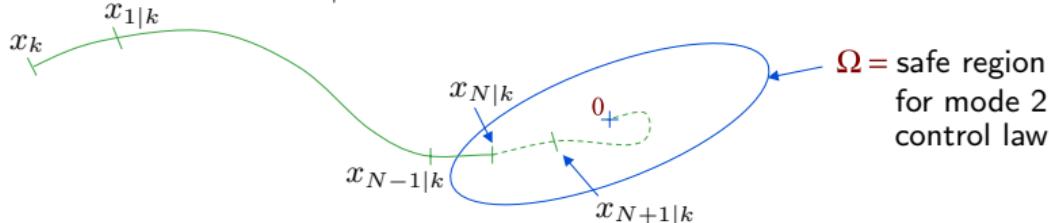
# Terminal constraint

The basic idea



# Terminal constraint

Terminal constraint:  $x_{N|k} \in \Omega$ , where  $\Omega = \text{terminal set}$



Choose  $\Omega$  so that:

$$(a). \quad x \in \Omega \implies \begin{cases} \underline{u} \leq Kx \leq \bar{u} \\ \underline{x} \leq x \leq \bar{x} \end{cases}$$

$$(b). \quad x \in \Omega \implies (A + BK)x \in \Omega$$

then  $\Omega$  is invariant for the mode 2 dynamics and constraints, so

$$x_{N|k} \in \Omega \implies \begin{cases} \underline{u} \leq u_{i|k} \leq \bar{u} \\ \underline{x} \leq x_{i|k} \leq \bar{x} \end{cases} \text{ for } i = N, N+1, \dots$$

i.e. constraints are satisfied over  
the infinite mode 2 prediction horizon

# Stability of constrained MPC

## Prototype MPC algorithm

At each time  $k = 0, 1, \dots$

(i). solve  $\mathbf{u}_k^* = \arg \min_{\mathbf{u}_k} J(x_k, \mathbf{u}_k)$

s.t.  $\underline{u} \leq u_{i|k} \leq \bar{u}, i = 0, \dots, N - 1$   
 $\underline{x} \leq x_{i|k} \leq \bar{x}, i = 1, \dots, N$   
 $x_{N|k} \in \Omega$

(ii). apply  $u_k = u_{0|k}^*$  to the system

Asymptotically stabilizes  $x = 0$  with region of attraction  $\mathcal{F}_N$ ,

$$\mathcal{F}_N = \left\{ x_0 : \exists \{u_0, \dots, u_{N-1}\} \text{ such that } \begin{array}{l} \underline{u} \leq u_i \leq \bar{u}, i = 0, \dots, N - 1 \\ \underline{x} \leq x_i \leq \bar{x}, i = 1, \dots, N \\ x_N \in \Omega \end{array} \right\}$$

= the set of all feasible initial conditions for  $N$ -step horizon

and terminal set  $\Omega$

# Terminal constraints

Make  $\Omega$  as large as possible so that the feasible set  $\mathcal{F}_N$  is maximized, i.e.

$$\Omega = \mathcal{X}_\infty = \lim_{j \rightarrow \infty} \mathcal{X}_j$$

where

- ★  $\mathcal{X}_j$  = initial conditions for which constraints are satisfied for  $j$  steps  
with  $u = Kx$   
 $= \left\{ x : \begin{array}{l} \underline{u} \leq K(A + BK)^i x \leq \bar{u} \\ \underline{x} \leq (A + BK)^i x \leq \bar{x} \end{array} \quad i = 0, \dots, j \right\}$
- ★  $\mathcal{X}_\infty = \mathcal{X}_\nu$  for some **finite**  $\nu$  if  $|\text{eig}(A + BK)| < 1$



$x \in \mathcal{X}_\infty$  if constraints are satisfied on a finite **constraint checking horizon**

## Terminal constraints – Example

Plant model:

$$x_{k+1} = Ax_k + Bu_k, \quad y_k = Cx_k$$

$$A = \begin{bmatrix} 1.1 & 2 \\ 0 & 0.95 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0.0787 \end{bmatrix} \quad C = [-1 \quad 1]$$

input constraints:

$$-1 \leq u_k \leq 1$$

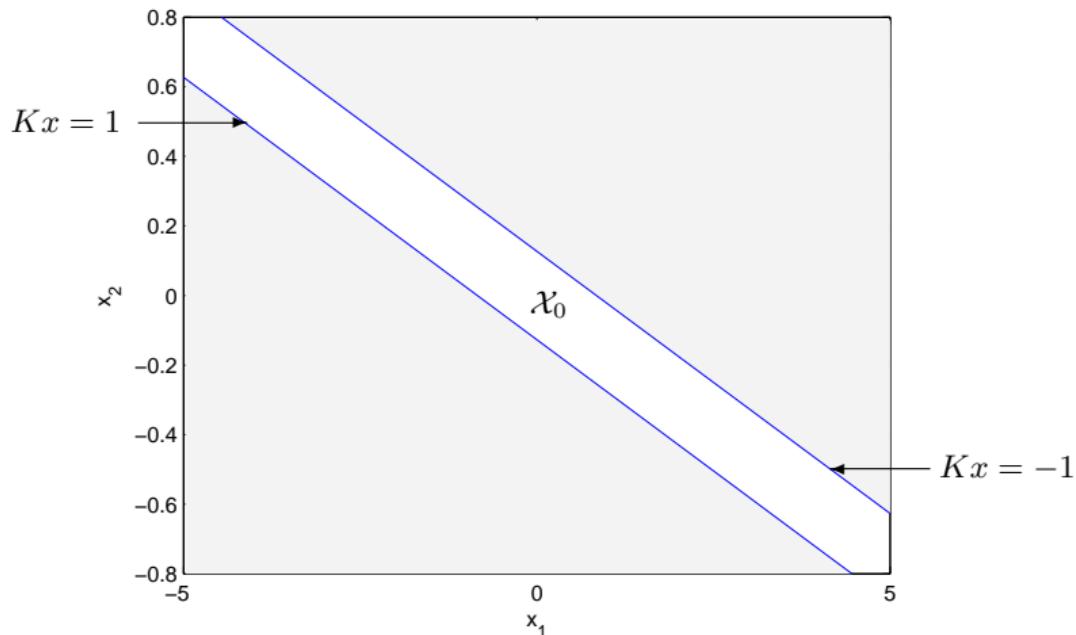
mode 2 feedback law:

$$K = [-1.19 \quad -7.88]$$

$$= K_{\text{LQ}} \text{ for } Q = C^T C, \quad R = 1$$

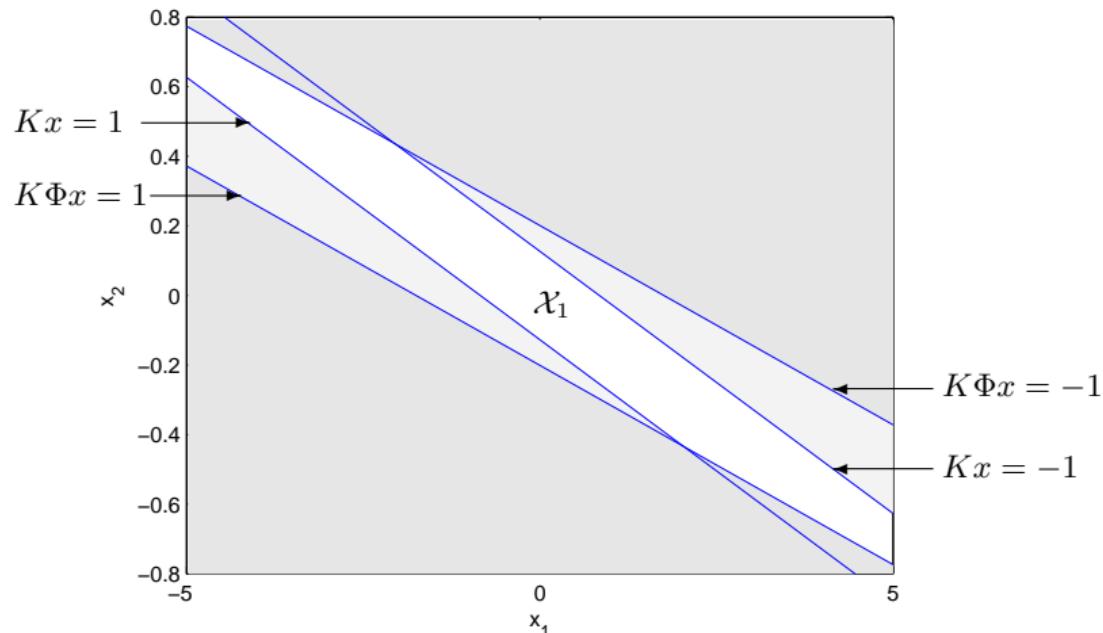
## Terminal constraints – example

Constraints:  $-1 \leq u \leq 1$



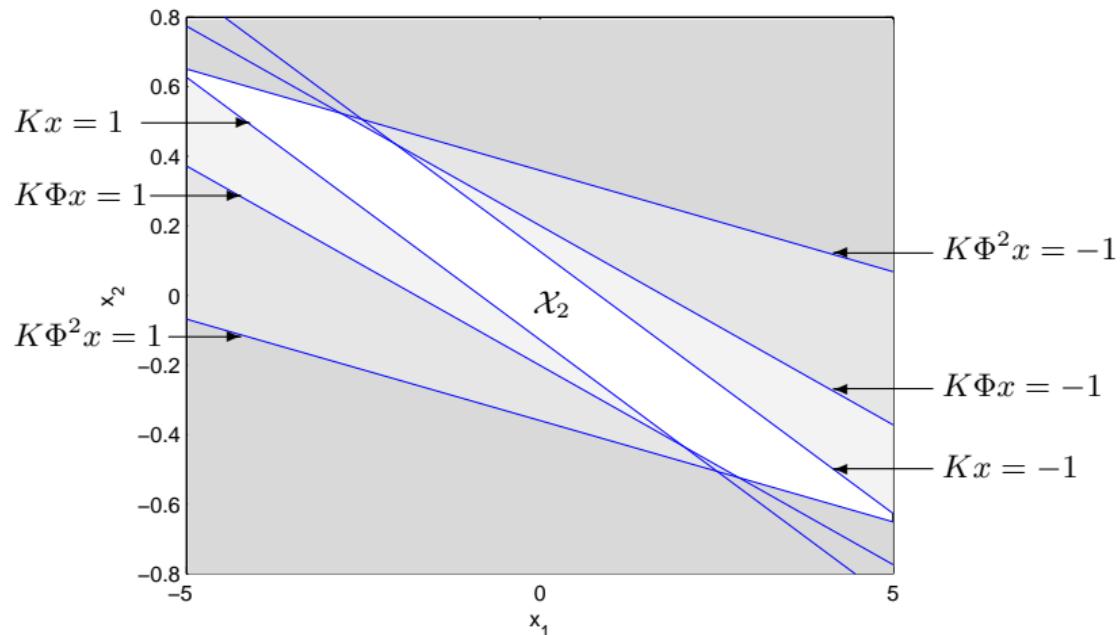
## Terminal constraints – example

Constraints:  $-1 \leq u \leq 1$



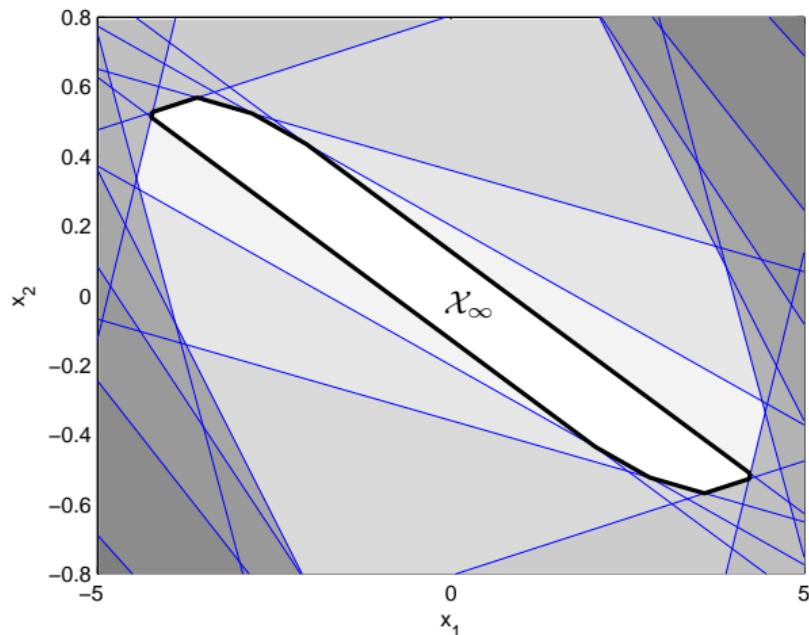
## Terminal constraints – example

Constraints:  $-1 \leq u \leq 1$



## Terminal constraints – example

Constraints:  $-1 \leq u \leq 1$



$$\mathcal{X}_4 = \mathcal{X}_5 = \dots = \mathcal{X}_j \text{ for all } j > 4 \text{ so } \mathcal{X}_4 = \mathcal{X}_\infty$$

## Terminal constraints – example

In this example  $\mathcal{X}_\infty$  is determined in a finite number of steps because

- ①  $(A + BK)$  is strictly stable, and
- ②  $((A + BK), K)$  is observable

④  $\Rightarrow \left\{ \begin{array}{l} \text{shortest distance of hyperplane} \\ K(A + BK)^i x \leq 1 \text{ from origin} \end{array} \right\} = \frac{1}{\|K(A + BK)^i\|_2}$   
 $\rightarrow \infty \quad \text{as } i \rightarrow \infty$

- ⑤  $\Rightarrow \mathcal{X}_\infty$  is bounded because  $x_0 \notin \mathcal{X}_\infty$  if  $x_0$  is sufficiently large

Here  $\{x : -1 \leq K(A + BK)^i x \leq 1\}$  contains  $\mathcal{X}_4$  for  $i > 4$



$$\mathcal{X}_\infty = \mathcal{X}_4$$

constraint checking horizon:  $\nu = 4$

## Terminal constraints – example

In this example  $\mathcal{X}_\infty$  is determined in a finite number of steps because

- ①  $(A + BK)$  is strictly stable, and
- ②  $((A + BK), K)$  is observable

④  $\Rightarrow \left\{ \begin{array}{l} \text{shortest distance of hyperplane} \\ K(A + BK)^i x \leq 1 \text{ from origin} \end{array} \right\} = \frac{1}{\|K(A + BK)^i\|_2}$   
 $\rightarrow \infty \quad \text{as } i \rightarrow \infty$

- ⑤  $\Rightarrow \mathcal{X}_\infty$  is bounded because  $x_0 \notin \mathcal{X}_\infty$  if  $x_0$  is sufficiently large

Here  $\{x : -1 \leq K(A + BK)^i x \leq 1\}$  contains  $\mathcal{X}_4$  for  $i > 4$



$$\mathcal{X}_\infty = \mathcal{X}_4$$

constraint checking horizon:  $\nu = 4$

# Terminal constraints

General case

Let  $\mathcal{X}_j = \{x : F\Phi^i x \leq \mathbf{1}, i = 0, \dots, j\}$  with  $\begin{cases} \Phi \text{ strictly stable} \\ (\Phi, F) \text{ observable} \end{cases}$

then:

- (i).  $\mathcal{X}_\infty = \mathcal{X}_\nu$  for finite  $\nu$
- (ii).  $\mathcal{X}_\nu = \mathcal{X}_\infty$  iff  $x \in \mathcal{X}_{\nu+1}$  whenever  $x \in \mathcal{X}_\nu$

Proof of (ii)

(a). for any  $j$ ,  $\mathcal{X}_{j+1} = \mathcal{X}_j \cap \{x : F\Phi^{j+1} x \leq \mathbf{1}\}$

so  $\mathcal{X}_j \supseteq \mathcal{X}_{j+1} \supseteq \lim_{j \rightarrow \infty} \mathcal{X}_j = \mathcal{X}_\infty$

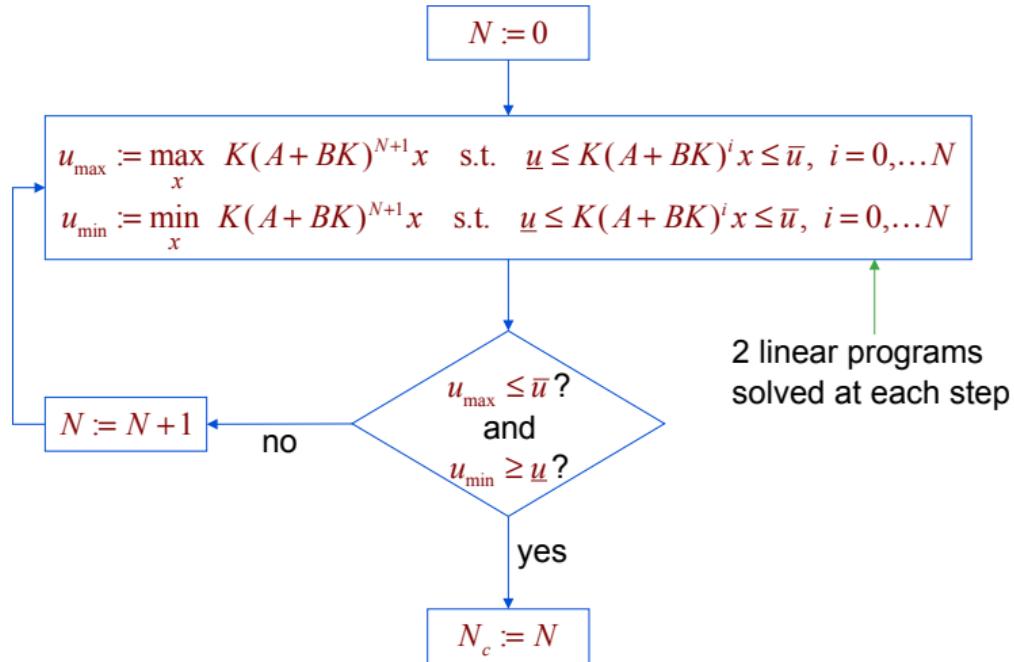
(b). if  $x \in \mathcal{X}_{\nu+1}$  whenever  $x \in \mathcal{X}_\nu$ , then  $\Phi x \in \mathcal{X}_\nu$  whenever  $x \in \mathcal{X}_\nu$

but  $\mathcal{X}_\nu \subseteq \{x : Fx \leq \mathbf{1}\}$  and it follows that  $\mathcal{X}_\nu \subseteq \mathcal{X}_\infty$

(a) & (b)  $\Rightarrow \mathcal{X}_\nu = \mathcal{X}_\infty$

# Terminal constraints – constraint checking horizon

Algorithm for computing constraint checking horizon  $N_c$   
for input constraints  $\underline{u} \leq u \leq \bar{u}$ :



# Constrained MPC

Define the terminal set  $\Omega$  as  $\mathcal{X}_{N_c}$

## MPC algorithm

At each time  $k = 0, 1, \dots$

(i). solve  $\mathbf{u}_k^* = \arg \min_{\mathbf{u}_k} J(x_k, \mathbf{u}_k)$

s.t.  $\underline{u} \leq u_{i|k} \leq \bar{u}, i = 0, \dots, N + N_c$

$\underline{x} \leq x_{i|k} \leq \bar{x}, i = 1, \dots, N + N_c$

(ii). apply  $u_k = u_{0|k}^*$  to the system

## Note

- \* predictions for  $i = N, \dots, N + N_c$ : 
$$\begin{cases} x_{i|k} = (A + BK)^{i-N} x_{N|k} \\ u_{i|k} = K(A + BK)^{i-N} x_{N|k} \end{cases}$$
- \*  $x_{N|k} \in \mathcal{X}_{N_c}$  implies linear constraints so online optimization is a QP

# Closed loop performance

Longer horizon  $N$  ensures improved predicted cost  $J^*(x_0)$

and is likely (but not certain) to give better closed-loop performance

**Example:** Cost vs  $N$  for  $x_0 = (-7.5, 0.5)$

$N$	6	7	8	11	$> 11$
$J^*(x_0)$	364.2	357.0	356.3	356.0	356.0
$J_{\text{cl}}(x_0)$	356.0	356.0	356.0	356.0	356.0

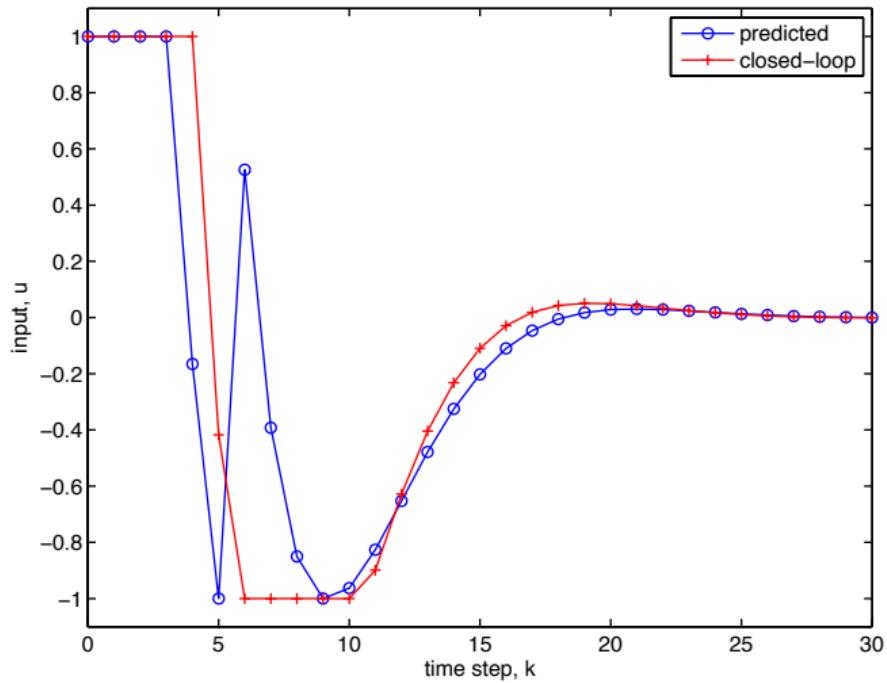
Closed loop cost:  $J_{\text{cl}}(x_0) := \sum_{k=0}^{\infty} (\|x_k\|_Q^2 + \|u_k\|_R^2)$

For this initial condition:

MPC with  $N = 11$  is identical to constrained LQ optimal control ( $N = \infty$ )!

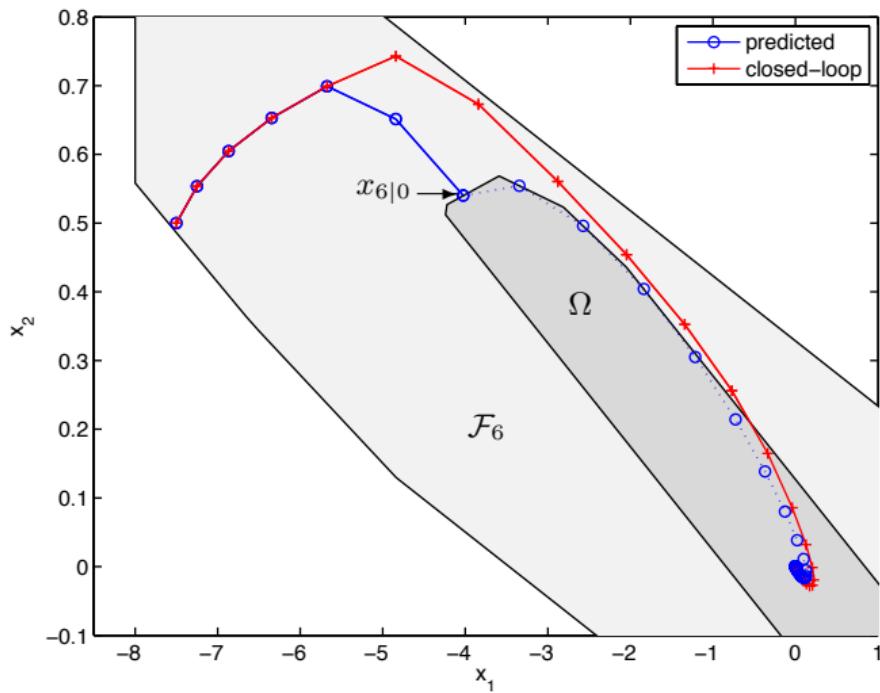
# Closed loop performance – example

Predicted and closed loop inputs for  $N = 6$



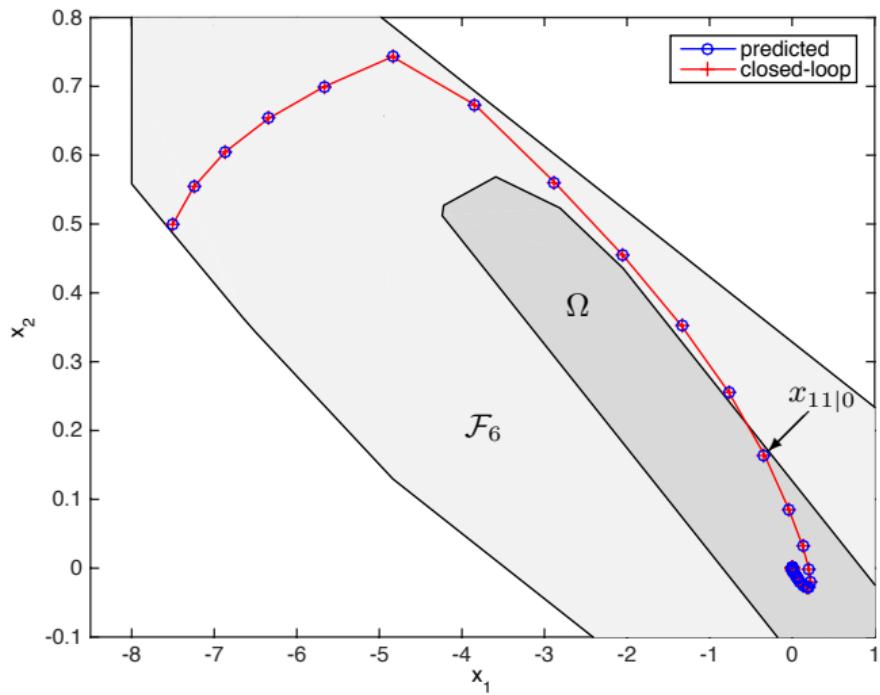
# Closed loop performance – example

Predicted and closed loop states for  $N = 6$



# Closed loop performance – example

Predicted and closed loop states for  $N = 11$



## Choice of mode 1 horizon – performance

- ▷ For this  $x_0$ :  $N = 11 \Rightarrow x_{N|0}$  lies in the interior of  $\Omega$



terminal constraint is inactive



no reduction in cost for  $N > 11$

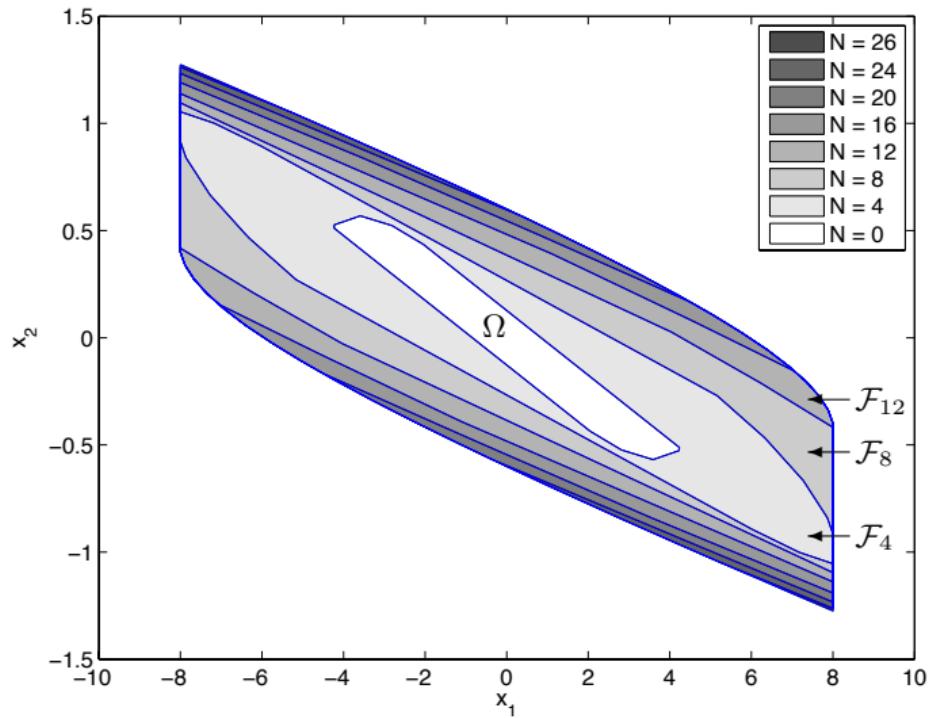
- ▷ Constrained LQ optimal performance is always obtained with  $N \geq N_\infty$  for some finite  $N_\infty$  dependent on  $x_0$
- ▷  $N_\infty$  may be large, implying high computational load  
but **closed loop** performance is often close to optimal for  $N < N_\infty$

(due to receding horizon)

in this example  $J_{\text{cl}}(x_0) \approx$  optimal for  $N \geq 6$

# Choice of mode 1 horizon – region of attraction

Increasing  $N$  increases the feasible set  $\mathcal{F}_N$



# Summary

- ▷ Linear MPC ingredients:
  - ★ Infinite cost horizon (via terminal cost)
  - ★ Terminal constraints (via constraint-checking horizon)
- ▷ Constraints are satisfied over an infinite prediction horizon
- ▷ Closed-loop system is asymptotically stable with region of attraction equal to the set of feasible initial conditions
- ▷ Ideal optimal performance if mode 1 horizon  $N$  is large enough (but finite)

## Lecture 4

# Robustness to disturbances

# Robustness to disturbances

- Review of nominal model predictive control
- Setpoint tracking and integral action
- Robustness to unknown disturbances
- Handling time-varying disturbances

## MPC with guaranteed stability – the basic idea



stabilizing linear controller: cost and constraints can be accounted for over an infinite prediction horizon

# Review

MPC optimization for linear model  $x_{k+1} = Ax_k + Bu_k$

$$\begin{aligned} & \underset{\mathbf{u}_k}{\text{minimize}} \quad \sum_{i=0}^{N-1} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2) + \|x_{N|k}\|_P^2 \\ & \text{subject to} \quad \underline{u} \leq u_{i|k} \leq \bar{u}, \quad i = 0, \dots, N + N_c \\ & \quad \underline{x} \leq x_{i|k} \leq \bar{x}, \quad i = 1, \dots, N + N_c \end{aligned}$$

where

- ★  $u_{i|k} = Kx_{i|k}$  for  $i \geq N$ , with  $K$  = unconstrained LQ optimal

- ★ terminal cost:  $\|x_{N|k}\|_P^2 = \sum_{i=N}^{\infty} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2)$ , with

$$P - \Phi^T P \Phi = Q + K^T R K, \quad \Phi = A + BK$$

- ★ terminal constraints are defined by the constraint checking horizon  $N_c$ :

$$\left. \begin{array}{l} \underline{u} \leq K\Phi^i x \leq \bar{u} \\ \underline{x} \leq \Phi^i x \leq \bar{x} \end{array} \right\} \quad i = 0, \dots, N_c \quad \Rightarrow \quad \left\{ \begin{array}{l} \underline{u} \leq K\Phi^{N_c+1} x \leq \bar{u} \\ \underline{x} \leq \Phi^{N_c+1} x \leq \bar{x} \end{array} \right.$$

# Review

MPC optimization for linear model  $x_{k+1} = Ax_k + Bu_k$

$$\begin{aligned} & \underset{\mathbf{u}_k}{\text{minimize}} \quad \sum_{i=0}^{N-1} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2) + \|x_{N|k}\|_P^2 \\ & \text{subject to} \quad \underline{u} \leq u_{i|k} \leq \bar{u}, \quad i = 0, \dots, N + N_c \\ & \quad \underline{x} \leq x_{i|k} \leq \bar{x}, \quad i = 1, \dots, N + N_c \end{aligned}$$

where

- \*  $u_{i|k} = Kx_{i|k}$  for  $i \geq N$ , with  $K$  = unconstrained LQ optimal

- \* terminal cost:  $\|x_{N|k}\|_P^2 = \sum_{i=N}^{\infty} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2)$ , with

$$P - \Phi^T P \Phi = Q + K^T R K, \quad \Phi = A + B K$$

- \* terminal constraints are defined by the constraint checking horizon  $N_c$ :

$$\left. \begin{array}{l} \underline{u} \leq K \Phi^i x \leq \bar{u} \\ \underline{x} \leq \Phi^i x \leq \bar{x} \end{array} \right\} \quad i = 0, \dots, N_c \quad \Rightarrow \quad \left\{ \begin{array}{l} \underline{u} \leq K \Phi^{N_c+1} x \leq \bar{u} \\ \underline{x} \leq \Phi^{N_c+1} x \leq \bar{x} \end{array} \right.$$

MPC optimization for linear model  $x_{k+1} = Ax_k + Bu_k$

$$\begin{aligned} & \underset{\mathbf{u}_k}{\text{minimize}} \quad \sum_{i=0}^{N-1} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2) + \|x_{N|k}\|_P^2 \\ & \text{subject to} \quad \underline{u} \leq u_{i|k} \leq \bar{u}, \quad i = 0, \dots, N + N_c \\ & \quad \underline{x} \leq x_{i|k} \leq \bar{x}, \quad i = 1, \dots, N + N_c \end{aligned}$$

where

- \*  $u_{i|k} = Kx_{i|k}$  for  $i \geq N$ , with  $K$  = unconstrained LQ optimal

- \* terminal cost:  $\|x_{N|k}\|_P^2 = \sum_{i=N}^{\infty} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2)$ , with

$$P - \Phi^T P \Phi = Q + K^T R K, \quad \Phi = A + B K$$

- \* terminal constraints are defined by the constraint checking horizon  $N_c$ :

$$\left. \begin{array}{l} \underline{u} \leq K\Phi^i x \leq \bar{u} \\ \underline{x} \leq \Phi^i x \leq \bar{x} \end{array} \right\} \quad i = 0, \dots, N_c \quad \Rightarrow \quad \left\{ \begin{array}{l} \underline{u} \leq K\Phi^{N_c+1} x \leq \bar{u} \\ \underline{x} \leq \Phi^{N_c+1} x \leq \bar{x} \end{array} \right.$$

# Review

MPC optimization for **nonlinear** model  $x_{k+1} = f(x_k, u_k)$

$$\begin{aligned} & \underset{\mathbf{u}_k}{\text{minimize}} \quad \sum_{i=0}^{N-1} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2) + \|x_{N|k}\|_P^2 \\ & \text{subject to} \quad \underline{u} \leq u_{i|k} \leq \bar{u}, \quad i = 0, \dots, N-1 \\ & \quad \underline{x} \leq x_{i|k} \leq \bar{x}, \quad i = 1, \dots, N-1 \\ & \quad x_{N|k} \in \Omega \end{aligned}$$

with

- \* mode 2 feedback:  $u_{i|k} = \kappa(x_{i|k})$  asymptotically stabilizes  $x = 0$  (locally)

- \* terminal cost:  $\|x_{N|k}\|_P^2 \geq \sum_{i=N}^{\infty} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2)$   
for mode 2 dynamics:  $x_{i+1|k} = f(x_{i|k}, \kappa(x_{i|k}))$

- \* terminal constraint set  $\Omega$ : invariant for mode 2 dynamics and constraints

$$\left. \begin{array}{l} f(x, \kappa(x)) \in \Omega \\ \underline{u} \leq \kappa(x) \leq \bar{u}, \quad \underline{x} \leq x \leq \bar{x} \end{array} \right\} \quad \text{for all } x \in \Omega$$

MPC optimization for **nonlinear** model  $x_{k+1} = f(x_k, u_k)$

$$\begin{aligned} & \underset{\mathbf{u}_k}{\text{minimize}} \quad \sum_{i=0}^{N-1} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2) + \|x_{N|k}\|_P^2 \\ & \text{subject to} \quad \underline{u} \leq u_{i|k} \leq \bar{u}, \quad i = 0, \dots, N-1 \\ & \quad \underline{x} \leq x_{i|k} \leq \bar{x}, \quad i = 1, \dots, N-1 \\ & \quad x_{N|k} \in \Omega \end{aligned}$$

with

- ★ mode 2 feedback:  $u_{i|k} = \kappa(x_{i|k})$  asymptotically stabilizes  $x = 0$  (locally)
- ★ terminal cost:  $\|x_{N|k}\|_P^2 \geq \sum_{i=N}^{\infty} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2)$   
for mode 2 dynamics:  $x_{i+1|k} = f(x_{i|k}, \kappa(x_{i|k}))$
- ★ terminal constraint set  $\Omega$ : invariant for mode 2 dynamics and constraints

$$\left. \begin{array}{l} f(x, \kappa(x)) \in \Omega \\ \underline{u} \leq \kappa(x) \leq \bar{u}, \quad \underline{x} \leq x \leq \bar{x} \end{array} \right\} \quad \text{for all } x \in \Omega$$

MPC optimization for **nonlinear** model  $x_{k+1} = f(x_k, u_k)$

$$\begin{aligned} & \underset{\mathbf{u}_k}{\text{minimize}} \quad \sum_{i=0}^{N-1} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2) + \|x_{N|k}\|_P^2 \\ & \text{subject to} \quad \underline{u} \leq u_{i|k} \leq \bar{u}, \quad i = 0, \dots, N-1 \\ & \quad \underline{x} \leq x_{i|k} \leq \bar{x}, \quad i = 1, \dots, N-1 \\ & \quad x_{N|k} \in \Omega \end{aligned}$$

with

- ★ mode 2 feedback:  $u_{i|k} = \kappa(x_{i|k})$  asymptotically stabilizes  $x = 0$  (locally)
- ★ terminal cost:  $\|x_{N|k}\|_P^2 \geq \sum_{i=N}^{\infty} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2)$   
for mode 2 dynamics:  $x_{i+1|k} = f(x_{i|k}, \kappa(x_{i|k}))$
- ★ terminal constraint set  $\Omega$ : invariant for mode 2 dynamics and constraints

$$\left. \begin{aligned} & f(x, \kappa(x)) \in \Omega \\ & \underline{u} \leq \kappa(x) \leq \bar{u}, \quad \underline{x} \leq x \leq \bar{x} \end{aligned} \right\} \quad \text{for all } x \in \Omega$$

# Comparison

## ▷ Linear MPC

terminal cost  $\leftarrow$  exact cost over the mode 2 horizon

terminal constraint set  $\leftarrow$  contains all feasible initial conditions for mode 2

## ▷ Nonlinear MPC

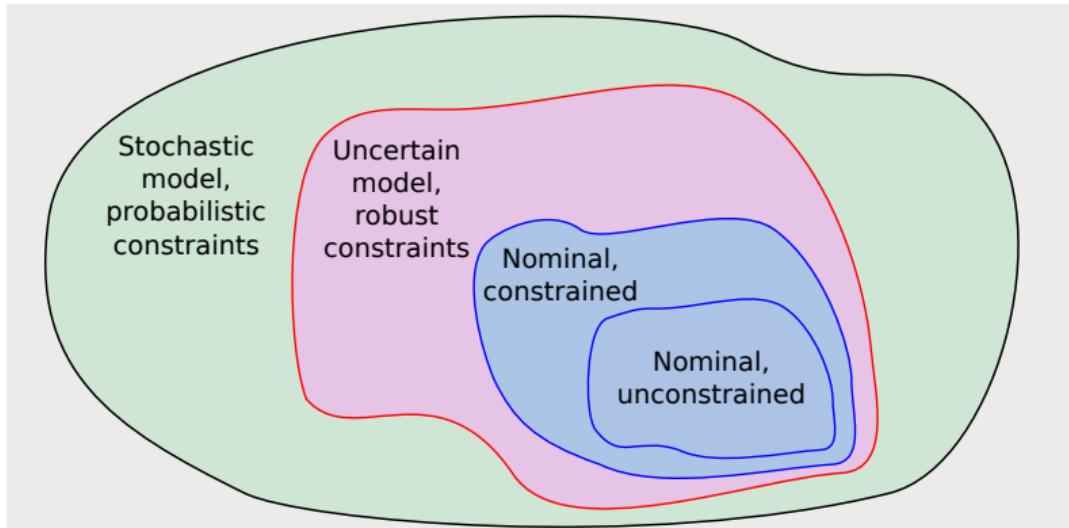
terminal cost  $\leftarrow$  upper bound on cost over mode 2 horizon

terminal constraint set  $\leftarrow$  invariant set (usually not the largest) for mode 2 dynamics and constraints

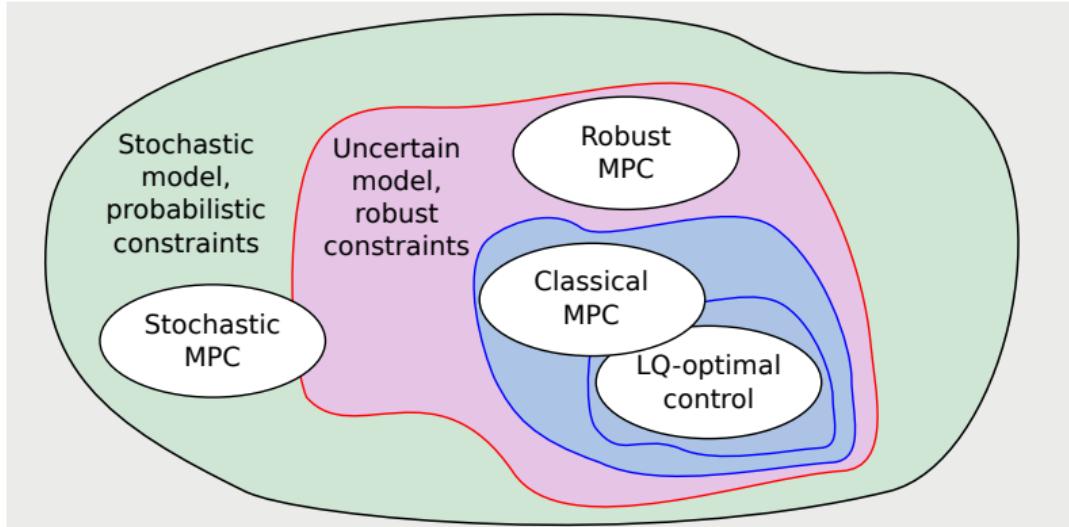
# Model uncertainty



# Model uncertainty



# Model uncertainty



# Model uncertainty

Common causes of model error and uncertainty:

- ▶ Unknown or time-varying model parameters
  - ▷ unknown loads & inertias, static friction
  - ▷ unknown d.c. gain
- ▶ Random (stochastic) model parameters
  - ▷ random process noise or sensor noise
- ▶ Incomplete measurement of states
  - ▷ state estimation error

# Setpoint tracking

Output setpoint:  $y^0$

$$y \rightarrow y^0 \Rightarrow \begin{cases} x \rightarrow x^0 \\ u \rightarrow u^0 \end{cases} \text{ where } \begin{aligned} x^0 &= Ax^0 + Bu^0 \\ y^0 &= Cx^0 \end{aligned}$$

$\Downarrow$

$$y^0 = C(I - A)^{-1}Bu^0$$

Setpoint for  $(u^0, x^0)$  is unique iff  $C(I - A)^{-1}B$  is invertible

e.g. if  $\dim(u) = \dim(y)$ , then

$$\begin{cases} u^0 = (C(I - A)^{-1}B)^{-1}y^0 \\ x^0 = (I - A)^{-1}Bu^0 \end{cases}$$

Tracking problem:  $y_k \rightarrow y^0$  subject to  $\begin{cases} \underline{u} \leq u_k \leq \bar{u} \\ \underline{x} \leq x_k \leq \bar{x} \end{cases}$

is only feasible if  $\underline{u} \leq u^0 \leq \bar{u}$  and  $\underline{x} \leq x^0 \leq \bar{x}$

# Setpoint tracking

Unconstrained tracking problem:

$$\underset{\mathbf{u}_k^\delta}{\text{minimize}} \quad \sum_{i=0}^{\infty} (\|x_{i|k}^\delta\|_Q^2 + \|u_{i|k}^\delta\|_R^2)$$

where  $x^\delta = x - x^0$   
 $u^\delta = u - u^0$

solution:  $u_k = Kx_k^\delta + u^0, \quad K = K_{LQ}$

Constrained tracking problem:

$$\underset{\mathbf{u}_k^\delta}{\text{minimize}} \quad \sum_{i=0}^{\infty} (\|x_{i|k}^\delta\|_Q^2 + \|u_{i|k}^\delta\|_R^2)$$

subject to  $\underline{u} \leq u_{i|k}^\delta + u^0 \leq \bar{u}, \quad i = 0, 1, \dots$   
 $\underline{x} \leq x_{i|k}^\delta + x^0 \leq \bar{x}, \quad i = 1, 2, \dots$

solution:  $u_k = u_{0|k}^{\delta*} + u^0$

# Setpoint tracking

If  $\hat{u}^0$  is used instead of  $u^0$

(e.g. d.c. gain  $C(I - A)^{-1}B$  unknown)

then  $u_k = u_{0|k}^{\delta*} + \hat{u}^0$  implies

$$\begin{aligned} u_k^\delta &= u_{0|k}^{\delta*} + (\hat{u}^0 - u^0) \\ x_{k+1}^\delta &= Ax_k^\delta + Bu_{0|k}^{\delta*} + B\underbrace{(\hat{u}^0 - u^0)}_{\text{constant disturbance}} \end{aligned}$$

and if  $u_{0|k}^{\delta*} \rightarrow Kx_k^\delta$  as  $k \rightarrow \infty$ , then

$$\begin{aligned} \lim_{k \rightarrow \infty} x_k^\delta &= (I - A - BK)^{-1}B(\hat{u}^0 - u^0) \neq 0 \\ \lim_{k \rightarrow \infty} y_k - y^0 &= \underbrace{C(I - A - BK)^{-1}B(\hat{u}^0 - u^0)}_{\text{steady state tracking error}} \neq 0 \end{aligned}$$

# Additive disturbances

Convert (constant) setpoint tracking problem into a regulation problem:

$$x \leftarrow x^\delta, y \leftarrow y^\delta, u \leftarrow u^\delta$$

Consider the effect of additive disturbance  $w$ :

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

Assume that  $w_k$  is unknown at time  $k$ , but is known to be

- ★ constant:  $w_k = w$  for all  $k$
- ★ or time-varying within a known polytopic set:  $w_k \in \mathcal{W}$  for all  $k$

where  $\mathcal{W} = \text{conv}\{w^{(1)}, \dots, w^{(r)}\}$

or  $\mathcal{W} = \{w : Hw \leq \mathbf{1}\}$

# Additive disturbances

Convert (constant) setpoint tracking problem into a regulation problem:

$$x \leftarrow x^\delta, y \leftarrow y^\delta, u \leftarrow u^\delta$$

Consider the effect of additive disturbance  $w$ :

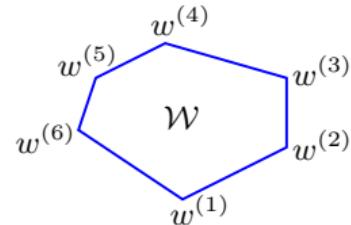
$$\begin{aligned}x_{k+1} &= Ax_k + Bu_k + Dw_k, \\y_k &= Cx_k\end{aligned}$$

Assume that  $w_k$  is unknown at time  $k$ , but is known to be

- ★ constant:  $w_k = w$  for all  $k$
- ★ or time-varying within a known polytopic set:  $w_k \in \mathcal{W}$  for all  $k$

where  $\mathcal{W} = \text{conv}\{w^{(1)}, \dots, w^{(r)}\}$

or  $\mathcal{W} = \{w : Hw \leq \mathbf{1}\}$



# Integral action

Introduce integral action to remove steady state error in  $y$   
by considering the **augmented system**:

$$z_k = \begin{bmatrix} x_k \\ v_k \end{bmatrix}, \quad z_{k+1} = \begin{bmatrix} A & 0 \\ C & I \end{bmatrix} z_k + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k + \begin{bmatrix} D \\ 0 \end{bmatrix} w_k$$

$v_k$  = integrator state

$$v_{k+1} = v_k + y_k$$

- ★ Linear feedback     $u_k = Kx_k + K_I v_k$   
is stabilizing if     $\left| \text{eig}\left(\begin{bmatrix} A + BK & BK_I \\ C & I \end{bmatrix}\right) \right| < 1$

- ★ If the closed-loop system is (strictly) stable and  $w_k \rightarrow w = \text{constant}$

then     $u_k \rightarrow u^{ss} \implies v_k \rightarrow v^{ss} \implies y_k \rightarrow 0$  even if  $w \neq 0$

... but arbitrary  $K_I$  may destabilize the closed loop system

# Integral action

Introduce integral action to remove steady state error in  $y$   
by considering the **augmented system**:

$$z_k = \begin{bmatrix} x_k \\ v_k \end{bmatrix}, \quad z_{k+1} = \begin{bmatrix} A & 0 \\ C & I \end{bmatrix} z_k + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k + \begin{bmatrix} D \\ 0 \end{bmatrix} w_k$$

$v_k$  = integrator state

$$v_{k+1} = v_k + y_k$$

- ★ Linear feedback  $u_k = Kx_k + K_I v_k$   
is stabilizing if  $\left| \text{eig}\left(\begin{bmatrix} A + BK & BK_I \\ C & I \end{bmatrix}\right) \right| < 1$

- ★ If the closed-loop system is (strictly) stable and  $w_k \rightarrow w = \text{constant}$

then  $u_k \rightarrow u^{ss} \implies v_k \rightarrow v^{ss} \implies y_k \rightarrow 0$  even if  $w \neq 0$

... but arbitrary  $K_I$  may destabilize the closed loop system

# Integral action

Introduce integral action to remove steady state error in  $y$   
by considering the **augmented system**:

$$z_k = \begin{bmatrix} x_k \\ v_k \end{bmatrix}, \quad z_{k+1} = \begin{bmatrix} A & 0 \\ C & I \end{bmatrix} z_k + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k + \begin{bmatrix} D \\ 0 \end{bmatrix} w_k$$

$v_k$  = integrator state

$$v_{k+1} = v_k + y_k$$

- ★ Linear feedback  $u_k = Kx_k + K_I v_k$   
is stabilizing if  $\left| \text{eig}\left(\begin{bmatrix} A + BK & BK_I \\ C & I \end{bmatrix}\right) \right| < 1$

- ★ If the closed-loop system is (strictly) stable and  $w_k \rightarrow w = \text{constant}$

then  $u_k \rightarrow u^{ss} \implies v_k \rightarrow v^{ss} \implies y_k \rightarrow 0$  even if  $w \neq 0$

... but arbitrary  $K_I$  may destabilize the closed loop system

# Integral action

Ensure stability by using a modified cost:

$$\underset{\mathbf{u}_k}{\text{minimize}} \quad \sum_{i=0}^{\infty} (\|z_{i|k}\|_{Q_z}^2 + \|u_{i|k}\|_R^2) \quad Q_z = \begin{bmatrix} Q & 0 \\ 0 & Q_I \end{bmatrix}$$

with predictions generated by an augmented model

$$z_{i+1|k} = \begin{bmatrix} A & 0 \\ C & I \end{bmatrix} z_{i|k} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_{i|k}, \quad z_{0|k} = \begin{bmatrix} x_k \\ v_k \end{bmatrix}$$

- ★ this is a **nominal** prediction model since  $w_k = 0$  is assumed
- ★ unconstrained solution:  $u_k = K_z z_k = Kx_k + K_I v_k$
- ★ if  $\left( \begin{bmatrix} A & 0 \\ C & I \end{bmatrix}, \begin{bmatrix} Q & 0 \\ 0 & Q_I \end{bmatrix} \right)$  is observable and  $w_k \rightarrow w = \text{constant}$   
then  $u_k \rightarrow u^{ss} \implies v_k \rightarrow v^{ss} \implies y_k \rightarrow 0$

even if  $w \neq 0!!$

# Integral action

Ensure stability by using a modified cost:

$$\underset{\mathbf{u}_k}{\text{minimize}} \quad \sum_{i=0}^{\infty} (\|z_{i|k}\|_{Q_z}^2 + \|u_{i|k}\|_R^2) \quad Q_z = \begin{bmatrix} Q & 0 \\ 0 & Q_I \end{bmatrix}$$

with predictions generated by an augmented model

$$z_{i+1|k} = \begin{bmatrix} A & 0 \\ C & I \end{bmatrix} z_{i|k} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_{i|k}, \quad z_{0|k} = \begin{bmatrix} x_k \\ v_k \end{bmatrix}$$

- ★ this is a **nominal** prediction model since  $w_k = 0$  is assumed
- ★ unconstrained solution:  $u_k = K_z z_k = Kx_k + K_I v_k$
- ★ if  $\left( \begin{bmatrix} A & 0 \\ C & I \end{bmatrix}, \begin{bmatrix} Q & 0 \\ 0 & Q_I \end{bmatrix} \right)$  is observable and  $w_k \rightarrow w = \text{constant}$   
then  $u_k \rightarrow u^{ss} \implies v_k \rightarrow v^{ss} \implies y_k \rightarrow 0$

even if  $w \neq 0!!$

# Integral action – example

Plant model:

$$x_{k+1} = Ax_k + Bu_k + Dw \quad y_k = Cx_k$$
$$A = \begin{bmatrix} 1.1 & 2 \\ 0 & 0.95 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0.0787 \end{bmatrix} \quad D = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad C = [-1 \quad 1]$$

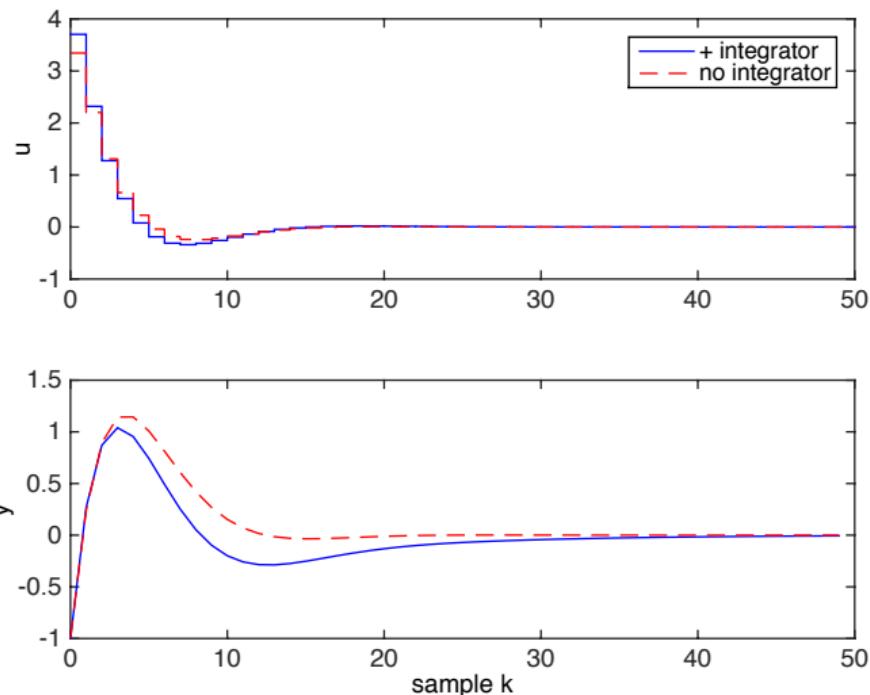
Constraints: **none**

Cost weighting matrices:  $Q_z = \begin{bmatrix} C^T C & 0 \\ 0 & 0.01 \end{bmatrix}$ ,  $R = 1$

Unconstrained LQ optimal feedback gain:

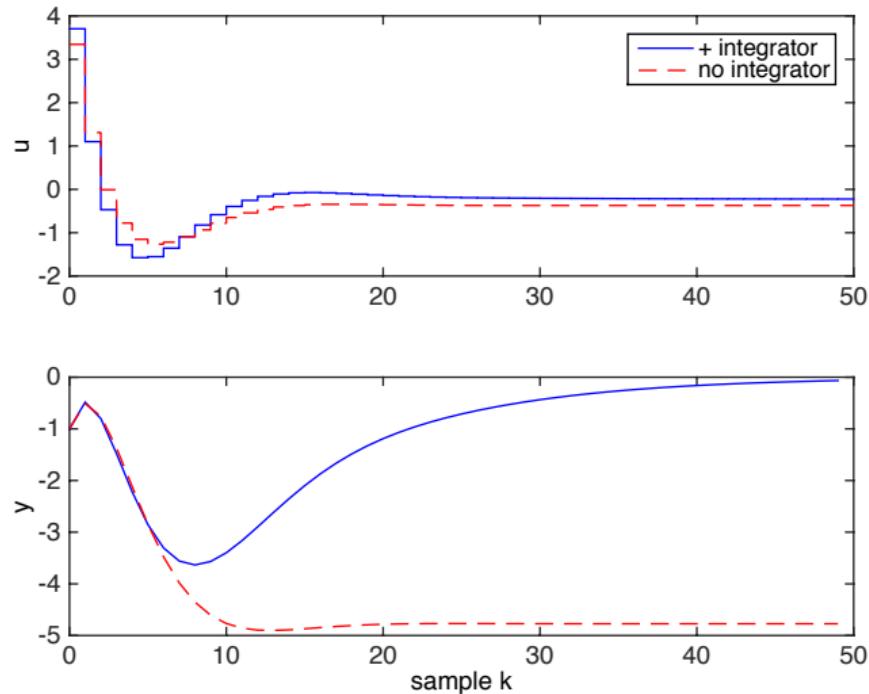
$$K_z = [-1.625 \quad -9.033 \quad 0.069]$$

## Integral action – example



Closed loop response for initial condition:  $x_0 = (0.5, -0.5)$   
no disturbance:  $w = 0$

## Integral action – example



Closed loop response for initial condition:  $x_0 = (0.5, -0.5)$   
constant disturbance:  $w = 0.75$

# Constrained MPC

Naive constrained MPC strategy:  $w = 0$  assumed in predictions

$$\begin{aligned} & \underset{\mathbf{u}_k}{\text{minimize}} \quad \sum_{i=0}^{N-1} (\|z_{i|k}\|_{Q_z}^2 + \|u_{i|k}\|_R^2) + \|z_{N|k}\|_P^2 \\ & \text{subject to} \quad \underline{u} \leq u_{i|k} \leq \bar{u}, \quad i = 0, \dots, N + N_c \\ & \quad \underline{x} \leq x_{i|k} \leq \bar{x}, \quad i = 1, \dots, N + N_c \end{aligned}$$

with:  $P$  and  $N_c$  determined for mode 2 control law  $u_{i|k} = K_z z_{i|k}$

initial prediction state:  $z_{0|k} = \begin{bmatrix} x_k \\ v_k \end{bmatrix}$  where  $v_{k+1} = v_k + y_k$

\* If closed loop system is stable

$$\text{then } u_k \rightarrow u^{ss} \implies v_k \rightarrow v^{ss} \implies y_k \rightarrow 0$$

\* but disturbance  $w_k$  is ignored in predictions, so

$$\begin{cases} J^*(z_{k+1}) - J^*(z_k) \cancel{\leq} 0 \\ \text{feasibility at time } k \cancel{\Rightarrow} \text{ feasibility at } k+1 \end{cases}$$

therefore no guarantee of stability

# Constrained MPC

Naive constrained MPC strategy:  $w = 0$  assumed in predictions

$$\begin{aligned} & \underset{\mathbf{u}_k}{\text{minimize}} \quad \sum_{i=0}^{N-1} (\|z_{i|k}\|_{Q_z}^2 + \|u_{i|k}\|_R^2) + \|z_{N|k}\|_P^2 \\ & \text{subject to} \quad \underline{u} \leq u_{i|k} \leq \bar{u}, \quad i = 0, \dots, N + N_c \\ & \quad \underline{x} \leq x_{i|k} \leq \bar{x}, \quad i = 1, \dots, N + N_c \end{aligned}$$

with:  $P$  and  $N_c$  determined for mode 2 control law  $u_{i|k} = K_z z_{i|k}$

initial prediction state:  $z_{0|k} = \begin{bmatrix} x_k \\ v_k \end{bmatrix}$  where  $v_{k+1} = v_k + y_k$

\* If closed loop system is stable

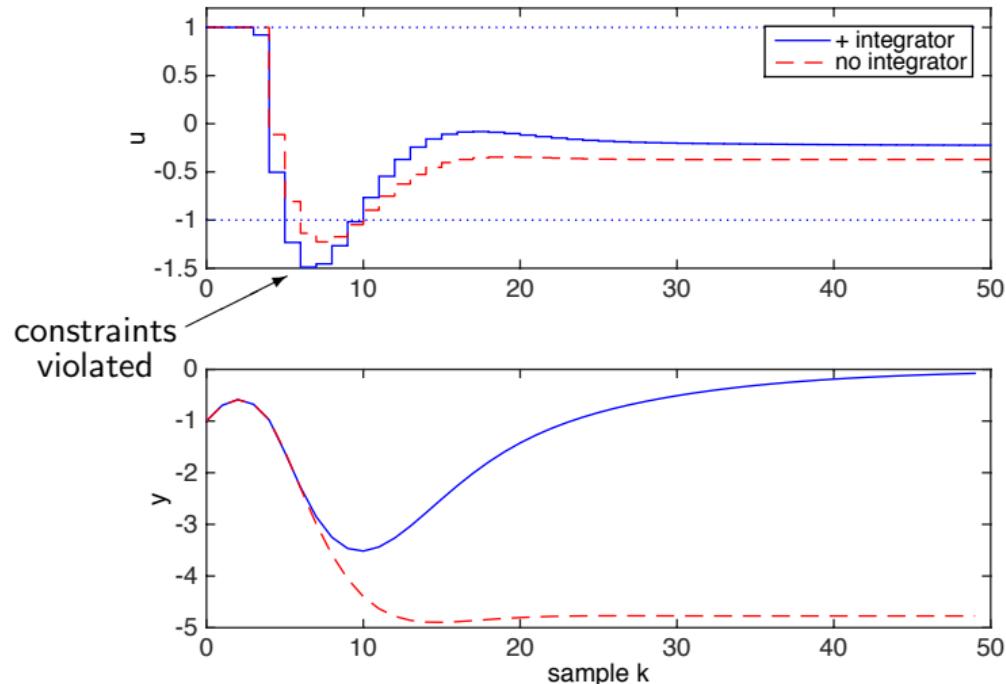
$$\text{then } u_k \rightarrow u^{ss} \implies v_k \rightarrow v^{ss} \implies y_k \rightarrow 0$$

\* but disturbance  $w_k$  is ignored in predictions, so

$$\begin{cases} J^*(z_{k+1}) - J^*(z_k) \not\leq 0 \\ \text{feasibility at time } k \not\Rightarrow \text{feasibility at } k+1 \end{cases}$$

therefore no guarantee of stability

# Constrained MPC – example



Closed loop response with  
**constraints:**  $-1 \leq u \leq 1$

initial condition:  $x_0 = (0.5, -0.5)$   
disturbance:  $w = 0.75$

## Robust constraints

If predictions satisfy constraints  $\begin{cases} \text{for all prediction times } i = 0, 1, \dots \\ \text{for all disturbances } w_i \in \mathcal{W} \end{cases}$

then feasibility of constraints at time  $k$  ensures feasibility at time  $k + 1$

- ▷ Linear dynamics plus additive disturbance enables decomposition

$$\begin{array}{ll} \text{nominal predicted state} & s_{i|k} \\ \text{uncertain predicted state} & e_{i|k} \end{array}$$

where

$$x_{i|k} = s_{i|k} + e_{i|k} \quad \begin{cases} s_{i+1|k} = \Phi s_{i|k} + B c_{i|k} & s_{0|k} = x_k \\ e_{i+1|k} = \Phi e_{i|k} + D w_{i|k} & e_{0|k} = 0 \end{cases}$$

- ▷ Pre-stabilized predictions:

$$u_{i|k} = K x_{i|k} + c_{i|k} \text{ and } \Phi = A + BK$$

where  $K = K_{\text{LQ}}$  is the unconstrained LQ optimal gain

## Pre-stabilized predictions – example

Scalar system:  $x_{k+1} = 2x_k + u_k + w_k,$  constraint:  $|x_k| \leq 2$

uncertainty:  $e_{i|k} = \sum_{j=0}^{i-1} 2^j w = (2^i - 1)w,$  disturbance:  $w_k = w$   
 $|w| \leq 1$

# Pre-stabilized predictions – example

Scalar system:  $x_{k+1} = 2x_k + u_k + w_k$ , constraint:  $|x_k| \leq 2$

uncertainty:  $e_{i|k} = \sum_{j=0}^{i-1} 2^j w = (2^i - 1)w$ , disturbance:  $w_k = w$   
 $|w| \leq 1$

Robust constraints:

$$|s_{i|k} + e_{i|k}| \leq 2 \text{ for all } |w| \leq 1$$



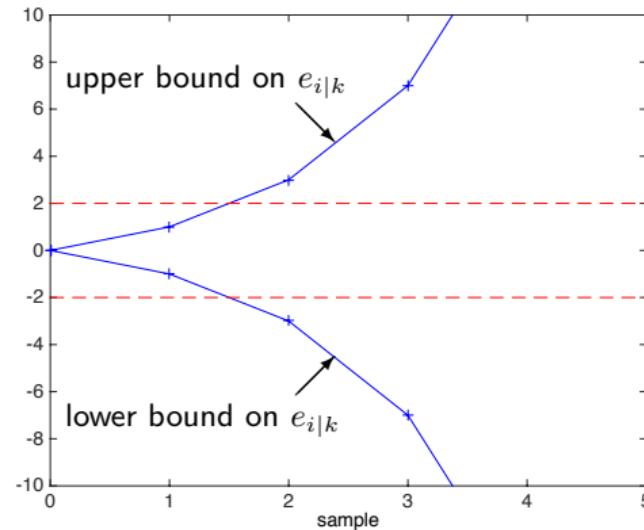
$$|s_{i|k}| \leq 2 - \max_{|w| \leq 1} |e_{i|k}|$$



$$|s_{i|k}| \leq 2 - (2^i - 1)$$



infeasible for all  $i > 1$



## Pre-stabilized predictions – example

Avoid infeasibility by using pre-stabilized predictions:

$$u_{i|k} = Kx_{i|k} + c_{i|k}, \quad K = -1.9, \quad c_{i|k} = \begin{cases} \text{free} & i = 0, \dots, N-1 \\ 0 & i \geq N \end{cases}$$

stable predictions:  $e_{i|k} = \sum_{j=0}^{i-1} 0.1^j w = (1 - 0.1^i)w/0.9, \quad |w| \leq 1$

# Pre-stabilized predictions – example

Avoid infeasibility by using pre-stabilized predictions:

$$u_{i|k} = Kx_{i|k} + c_{i|k}, \quad K = -1.9, \quad c_{i|k} = \begin{cases} \text{free} & i = 0, \dots, N-1 \\ 0 & i \geq N \end{cases}$$

stable predictions:  $e_{i|k} = \sum_{j=0}^{i-1} 0.1^j w = (1 - 0.1^i)w/0.9, \quad |w| \leq 1$

Robust constraints:

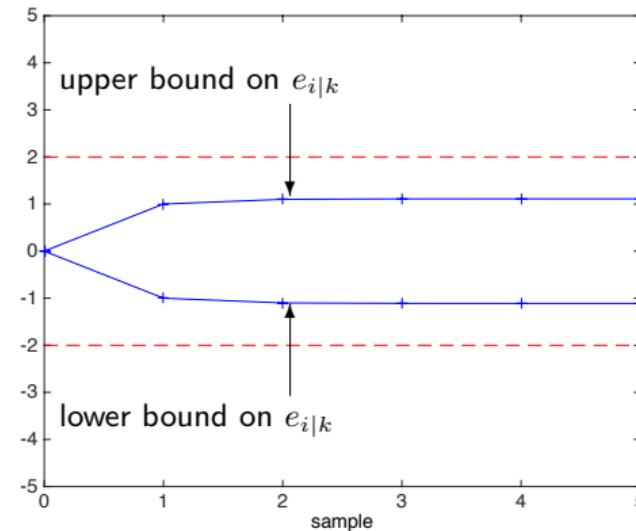
$$|s_{i|k} + e_{i|k}| \leq 2 \quad \text{for all } |w| \leq 1$$



$$|s_{i|k}| \leq 2 - \max_{|w| \leq 1} |e_{i|k}|$$

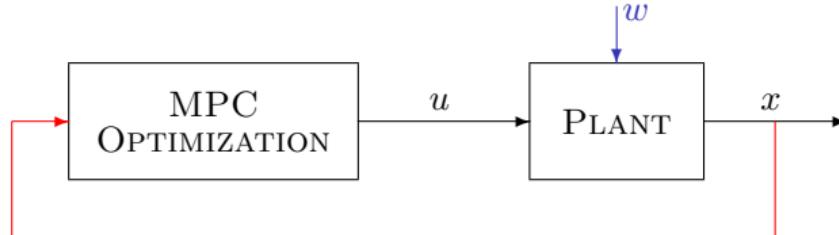


$$|s_{i|k}| \leq \underbrace{2 - (1 - 0.1^i)/0.9}_{>0 \text{ for all } i}$$

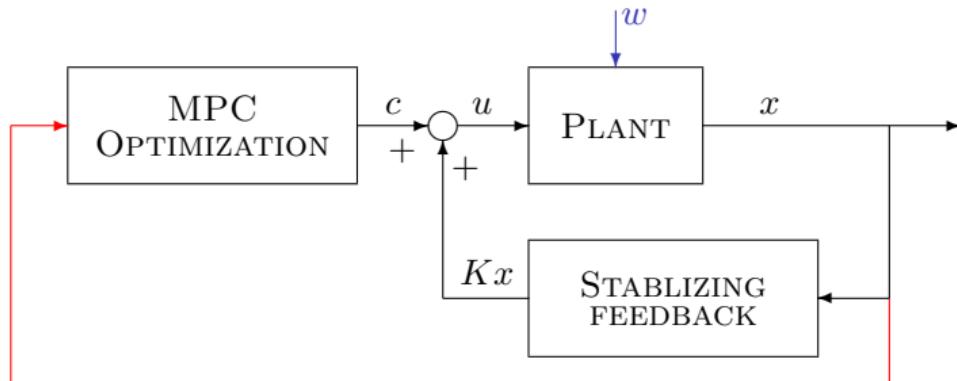


# Pre-stabilized predictions

- ▷ Feedback structure of MPC with open loop predictions:



- ▷ Feedback structure of MPC with pre-stabilized predictions:



# General form of robust constraints

How can we impose (general linear) constraints robustly?

- ★ Pre-stabilized predictions:

$$x_{i|k} = s_{i|k} + e_{i|k}$$
$$\begin{cases} s_{i+1|k} = \Phi s_{i|k} + B c_{i|k} & s_{0|k} = x_k \\ e_{i+1|k} = \Phi e_{i|k} + D w_{i|k} & e_{0|k} = 0 \end{cases}$$
$$\implies e_{i|k} = D w_{i-1} + \Phi D w_{i-2} + \dots + \Phi^{i-1} D w_0$$

- ★ General linear constraints:  $F x_{i|k} + G u_{i|k} \leq \mathbf{1}$   
are equivalent to **tightened constraints** on nominal predictions:

$$(F + GK)s_{i|k} + Gc_{i|k} \leq \mathbf{1} - h_i$$

where  $h_0 = 0$

$$h_i = \max_{w_0, \dots, w_{i-1} \in \mathcal{W}} (F + GK)e_{i|k}, \quad i = 1, 2, \dots$$

(i.e.  $h_i = h_{i-1} + \max_{w \in \mathcal{W}} (F + GK)w$   
requiring one LP for each row of  $h_i$ )

# General form of robust constraints

How can we impose (general linear) constraints robustly?

- ★ Pre-stabilized predictions:

$$x_{i|k} = s_{i|k} + e_{i|k}$$
$$\begin{cases} s_{i+1|k} = \Phi s_{i|k} + B c_{i|k} & s_{0|k} = x_k \\ e_{i+1|k} = \Phi e_{i|k} + D w_{i|k} & e_{0|k} = 0 \end{cases}$$
$$\implies e_{i|k} = D w_{i-1} + \Phi D w_{i-2} + \dots + \Phi^{i-1} D w_0$$

- ★ General linear constraints:  $F x_{i|k} + G u_{i|k} \leq \mathbf{1}$   
are equivalent to **tightened constraints** on nominal predictions:

$$(F + GK)s_{i|k} + Gc_{i|k} \leq \mathbf{1} - h_i$$

where  $h_0 = 0$

$$h_i = \max_{w_0, \dots, w_{i-1} \in \mathcal{W}} (F + GK)e_{i|k}, \quad i = 1, 2, \dots$$

(i.e.  $h_i = h_{i-1} + \max_{w \in \mathcal{W}} (F + GK)w$   
requiring one LP for each row of  $h_i$ )

# Tube interpretation

The uncertainty in predictions:  $e_{i+1|k} = \Phi e_{i|k} + D w_i$ ,  $w_i \in \mathcal{W}$   
evolves inside a **tube** (a sequence of sets):  $e_{i|k} \in E_{i|k}$ , where

$$E_{i|k} = D\mathcal{W} \oplus \Phi D\mathcal{W} \oplus \cdots \oplus \Phi^{i-1} D\mathcal{W}, \quad i = 1, 2, \dots$$

Hence we can define:

- ★ a state tube  $x_{i|k} = s_{i|k} + e_{i|k} \in \mathcal{X}_{i|k}$

$$\mathcal{X}_{i|k} = \{s_{i|k}\} \oplus E_{i|k}, \quad i = 0, 1, \dots$$

- ★ a control input tube  $u_{i|k} = Kx_{i|k} + c_{i|k} = Ks_{i|k} + c_{i|k} + Ke_{i|k} \in \mathcal{U}_{i|k}$

$$\mathcal{U}_{i|k} = \{Ks_{i|k} + c_{i|k}\} \oplus KE_{i|k}, \quad i = 0, 1, \dots$$

and impose constraints robustly for the state and input tubes

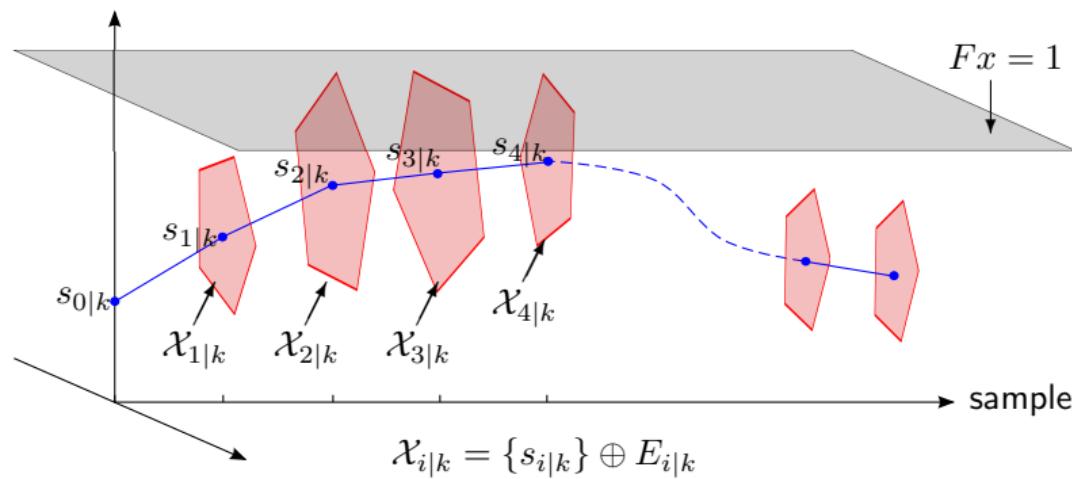
(where  $\oplus$  is Minkowski set addition)

# Tube interpretation

The uncertainty in predictions:  $e_{i+1|k} = \Phi e_{i|k} + D w_i$ ,  $w_i \in \mathcal{W}$   
evolves inside a **tube** (a sequence of sets):  $e_{i|k} \in E_{i|k}$ , where

$$E_{i|k} = D\mathcal{W} \oplus \Phi D\mathcal{W} \oplus \cdots \oplus \Phi^{i-1} D\mathcal{W}, \quad i = 1, 2, \dots$$

e.g. for constraints  $Fx \leq \mathbf{1}$  ( $G = 0$ )



## Prototype robust MPC algorithm

Offline: compute  $N_c$  and  $h_1, \dots, h_{N_c}$ . Online at  $k = 0, 1, \dots$ :

(i). solve  $\mathbf{c}_k^* = \arg \min_{\mathbf{c}_k} J(x_k, \mathbf{c}_k)$

$$\text{s.t. } (F + GK)s_{i|k} + Gc_{i|k} \leq \mathbf{1} - h_i, \quad i = 0, \dots, N + N_c$$

(ii). apply  $u_k = Kx_k + c_{0|k}^*$  to the system

- ★ tightened linear constraints are applied to nominal predictions

- ★  $N_c$  is the constraint-checking horizon:

$$(F + GK)\Phi^i s \leq \mathbf{1} - h_i, \quad i = 0, \dots, N_c \implies (F + GK)\Phi^{N_c+1}s \leq \mathbf{1} - h_{N_c+1}$$

- ★ the online optimization is robustly recursively feasible

## Prototype robust MPC algorithm

Offline: compute  $N_c$  and  $h_1, \dots, h_{N_c}$ . Online at  $k = 0, 1, \dots$ :

- (i). solve  $\mathbf{c}_k^* = \arg \min_{\mathbf{c}_k} J(x_k, \mathbf{c}_k)$   
s.t.  $(F + GK)s_{i|k} + Gc_{i|k} \leq \mathbf{1} - h_i, i = 0, \dots, N + N_c$
- (ii). apply  $u_k = Kx_k + c_{0|k}^*$  to the system

two alternative cost functions:

- ★ nominal cost (i.e. cost evaluated assuming  $w_i = 0$  for all  $i$ )

$$J(x_k, \mathbf{c}_k) = \sum_{i=0}^{\infty} (\|s_{i|k}\|_Q^2 + \|Ks_{i|k} + c_{i|k}\|_R^2) = \|x_k\|_P^2 + \|\mathbf{c}_k\|_{W_c}^2$$

- ★ worst case cost, defined in terms of a desired disturbance gain  $\gamma$ :

$$J(x_k, \mathbf{c}_k) = \max_{w_i \in \mathcal{W}, i=0,1,\dots} \sum_{i=0}^{\infty} (\|x_{i|k}\|_Q^2 + \|u_{i|k}\|_R^2 - \gamma^2 \|w_i\|^2)$$

# Convergence of robust MPC with nominal cost

If  $u_{i|k} = Kx_{i|k} + c_{i|k}$  for  $K = K_{\text{LQ}}$ , then:

- the unconstrained optimum is  $\mathbf{c}_k = 0$ , so the nominal cost is

$$J(x_k, \mathbf{c}_k) = \|x_k\|_P^2 + \|\mathbf{c}_k\|_{W_c}^2$$

and  $W_c$  is block-diagonal:  $W_c = \text{diag}\{P_c, \dots, P_c\}$

- recursive feasibility  $\Rightarrow \tilde{\mathbf{c}}_{k+1} = (c_{1|k}^*, \dots, c_{N-1|k}^*, 0)$  feasible at  $k+1$

- hence  $\|\mathbf{c}_{k+1}^*\|_{W_c}^2 \leq \|\mathbf{c}_k^*\|_{W_c}^2 - \|c_{0|k}^*\|_{P_c}^2$

$$\Rightarrow \sum_{k=0}^{\infty} \|c_{0|k}\|_{P_c}^2 \leq \|\mathbf{c}_0^*\|_{W_c}^2 < \infty$$

$$\Rightarrow \lim_{k \rightarrow \infty} c_{0|k} = 0$$

- therefore  $u_k \rightarrow Kx_k$  as  $k \rightarrow \infty$   
 $x_k \rightarrow$  the (minimal) robustly invariant set  
under unconstrained LQ optimal feedback

# Robust MPC with constant disturbance

Assume  $w_k = w = \text{constant}$  for all  $k$

combine: pre-stabilized predictions  
augmented state space model

- ★ Predicted state and input sequences:

$$\begin{aligned}x_{i|k} &= [I \quad 0] (s_{i|k} + e_{i|k}) \\u_{i|k} &= K_z (s_{i|k} + e_{i|k}) + c_{i|k}\end{aligned}$$

- ★ Prediction model:

$$\text{nominal} \quad s_{i+1|k} = \Phi s_{i|k} + \begin{bmatrix} B \\ 0 \end{bmatrix} c_{i|k} \quad \Phi = \begin{bmatrix} A & 0 \\ C & I \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} K_z$$

$$\text{uncertain} \quad e_{i|k} = \sum_{j=0}^{i-1} \Phi^j \begin{bmatrix} D \\ 0 \end{bmatrix} w \quad s_{0|k} = \begin{bmatrix} x_k \\ v_k \end{bmatrix}, \quad e_{0|k} = 0$$

- ★ Nominal cost:

$$J(x_k, v_k, \mathbf{c}_k) = \sum_{i=0}^{\infty} (\|s_{i|k}\|_{Q_z}^2 + \|K_z s_{i|k} + c_{i|k}\|_R^2)$$

# Robust MPC with constant disturbance

Assume  $w_k = w = \text{constant for all } k$

combine: pre-stabilized predictions  
augmented state space model

★ robust state constraints:

$$\underline{x} \leq x_{i|k} \leq \bar{x} \iff \underline{x} + h_i \leq s_{i|k} \leq \bar{x} - h_i$$

$$h_i = \max_{w \in \mathcal{W}} [I \quad 0] \sum_{j=0}^{i-1} \Phi^j \begin{bmatrix} D \\ 0 \end{bmatrix} w$$

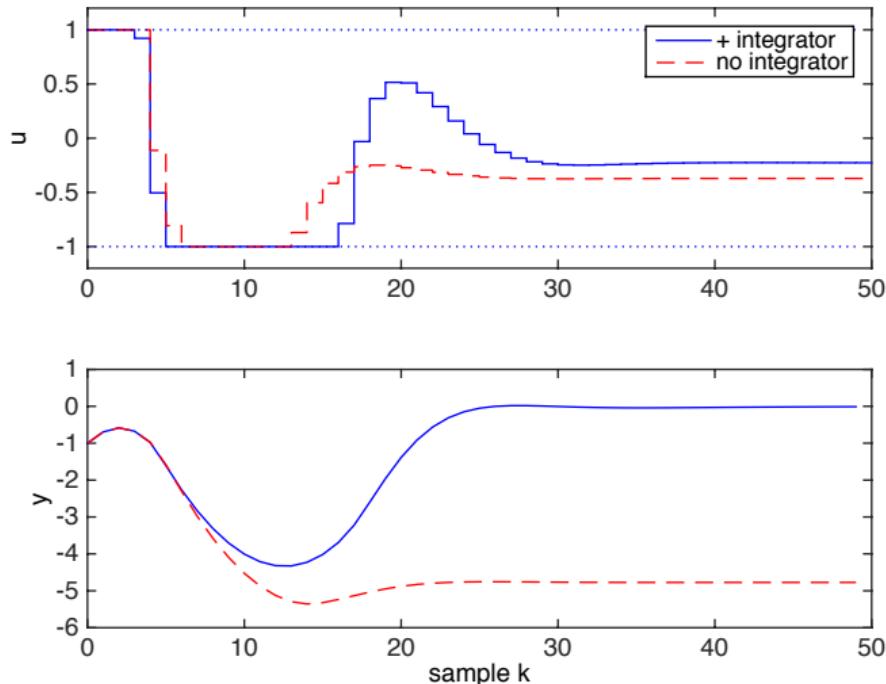
★ robust input constraints:

$$\underline{u} \leq u_{i|k} \leq \bar{u} \iff \underline{u} + h'_i \leq K_z s_{i|k} + c_{i|k} \leq \bar{u} - h'_i$$

$$h'_i = \max_{w \in \mathcal{W}} K_z \sum_{j=0}^{i-1} \Phi^j \begin{bmatrix} D \\ 0 \end{bmatrix} w$$

★  $N_c$  and  $h_i, h'_i$  for  $i = 1, \dots, N_c$  can be computed offline

# Robust MPC with constant disturbance – example



Closed loop response with  
constraints:  $-1 \leq u \leq 1$

initial condition:  $x_0 = (0.5, -0.5)$   
disturbance:  $w = 0.75$

# Summary

- ▷ Integral action: augment model with integrated output error  
include integrated output error in cost

then

- (i). closed loop system is stable if  $w = 0$
- (ii). steady state error must be zero if response is stable for  $w \neq 0$

- ▷ Robust MPC: use pre-stabilized predictions  
apply constraints for all possible future uncertainty

then

- (i). constraint feasibility is guaranteed at all times if initially feasible
- (ii). closed loop system inherits the stability and convergence properties of unconstrained LQ optimal control (assuming nominal cost)

## ① Introduction and Motivation

Basic MPC strategy; prediction models; input and state constraints; constraint handling: saturation, anti-windup, predictive control

## ② Prediction and optimization

Input/state prediction equations; unconstrained optimization. Infinite horizon cost; dual mode predictions. Incorporating constraints; quadratic programming.

## ③ Closed loop properties

Lyapunov analysis based on predicted cost. Recursive feasibility; terminal constraints; the constraint checking horizon. Constrained LQ-optimal control.

## ④ Robustness to disturbances

Setpoint tracking; MPC with integral action. Robustness to constant disturbances: prestabilized predictions and robust feasibility. Handling time-varying disturbances.

The End