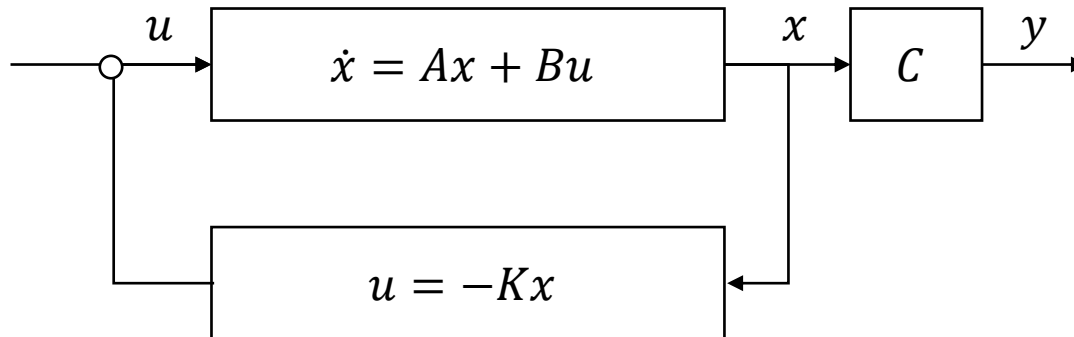


# Modern Control Theory

## Model Predictive Control (MPC)



# *Introduction*

---

## **Motivation**

Model Predictive Control (MPC) is the most effective way to control optimization dealing with constraints.

- pioneered by Richalet (1979) and Cutler & Ramaker (1979)
- well accepted by industry:
  - \_ MPC can handle multivariable processes with large time-delays, non-minimum-phase, unstable poles
  - \_ easy concept, easy tuning
  - \_ MPC can handle constraints, allows operation closer to constraints
  - \_ MPC can handle structure changes and actuator failures
- many successful applications have been reported

# *Introduction*

---

## **Ingredients of MPC:**

- Process & disturbance model
- Performance index
- Constraint handling
- Optimization
- Receding horizon principle

## **Process & disturbance models:**

The models applied in MPC serve two purposes:

- Prediction of the behavior of the future output of the process  
on the basis of inputs and known disturbances applied to the process in the past
- Calculation of the input signal to the process that minimizes the given objective function

These models are not necessarily the same.

# *Introduction*

---

## **Process & disturbance models:**

The models applied in MPC serve two purposes:

- Prediction of the behavior of the future output of the process  
on the basis of inputs and known disturbances applied to the process in the past
- Calculation of the input signal to the process that minimizes the given objective function

These models are not necessarily the same.

# *Introduction*

---

## Model assumptions:

- Linear
- Time-invariant
- Discrete time
- Causal
- Finite order
- State space description

## State space models:

- especially suited for MIMO systems
- compact model description
- compact and low order controller
- computations are well conditioned
- algorithms easy to implement

# Introduction

---

Two types of IO models are applied:

- Input-Output (IO) Model:

$$y(k) = G(z)u(k), \quad G(z) \text{ strictly proper}$$

- Increment-Input-Output (IIO) Model:

$$y(k) = G(z)\Delta u(k), \quad G(z) \text{ strictly proper}$$

where

$$\Delta u(k) = (1 - z^{-1})u(k) = u(k) - u(k - 1)$$

# Standard performance index

The standard performance index at time  $k$ :

$$J(v, k) = \sum_{j=0}^{N_p-1} \hat{z}^T(k+j|k) \Omega(j) \hat{z}(k+j|k)$$

- Generalized predictive control (GPC) performance index

$$J(\Delta u, k) = \sum_{j=N_m}^{N_p} |\hat{y}_p(k+j|k) - r(k+j|k)|^2 + \lambda^2 \sum_{j=1}^{N_c} |\Delta u(k+j-1|k)|^2$$

- Linear quadratic predictive control (LQPC) performance index

$$J(u, k) = \sum_{j=N_m}^{N_p} \hat{x}^T(k+j|k) Q \hat{x}(k+j|k) + \sum_{j=1}^{N_c} u^T(k+j-1|k) R u(k+j-1|k)$$

# Constraints

- Inequality constraints

$$u_{\min} \leq u(k) \leq u_{\max} \Delta u_{\min} \leq \Delta u(k) \leq \Delta u_{\max}$$

$$y_{\min} \leq y(k) \leq y_{\max} x_{\min} \leq x(k) \leq x_{\max}$$

$$A\hat{\eta}(k+j) \leq b(k+j) \quad \forall j \geq 0$$

- Equality constraints: motivated by control algorithm itself

Control horizon constraint:  $\Delta u(k+j|k) = 0$  for  $j \geq N_c$

The state end-point constraint:  $\hat{x}(k+N_p|k) = x_{ss}$

$$M\hat{\phi}(k+j) = 0 \quad \forall j \geq 0$$



# Optimization

---

The MPC controller minimizes the standard performance index

$$J = \sum_{j=0}^{N_p-1} \hat{z}^T(k+j|k) \Omega(j) \hat{z}(k+j|k)$$

subject to the constraints

$$A\hat{\eta}(k+j) \leq b(k+j) \quad \forall j \geq 0$$

$$M\hat{\phi}(k+j) = 0 \quad \forall j \geq 0$$

# Optimization

---

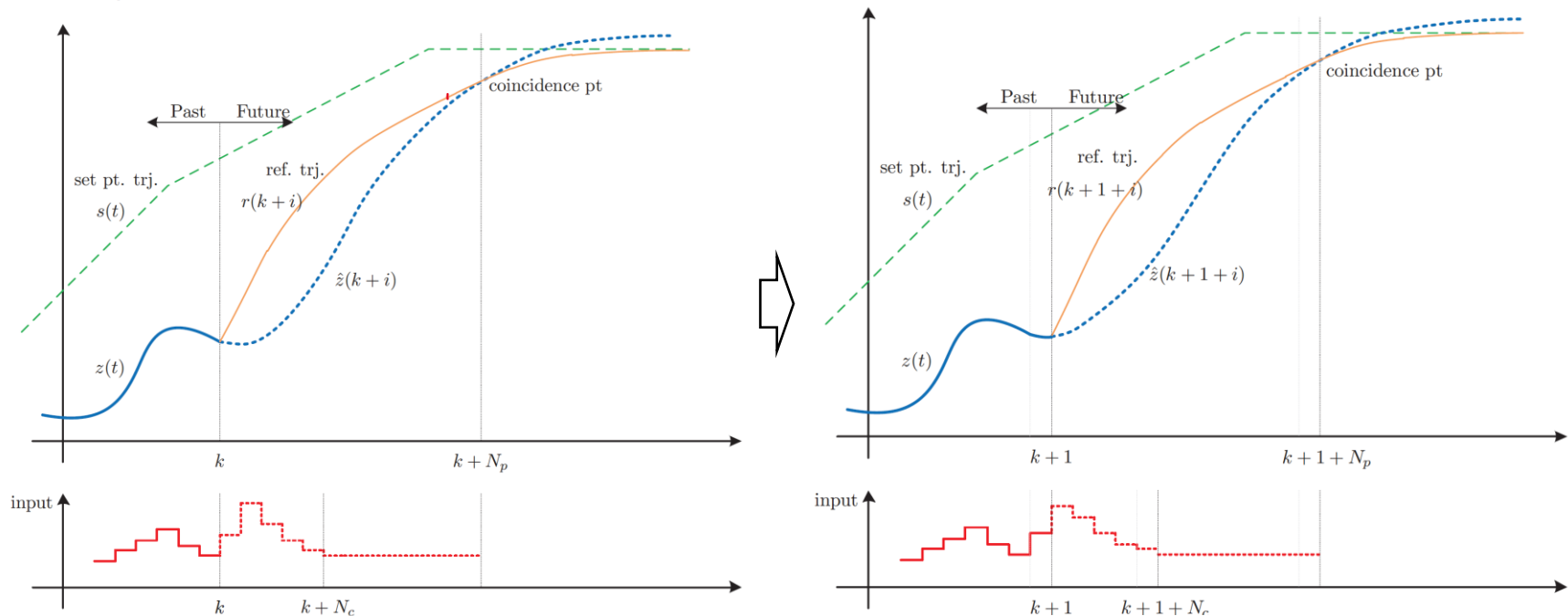
An optimization algorithm will be applied to compute a sequence of future control signals that minimizes the performance index subject to the given constraints.

- For
  - linear models
  - linear constraints
  - quadratic (2-norm) performance indexwe can use quadratic programming algorithms.
- For 1-norm or  $\infty$ -norm performance index: linear programming algorithms.

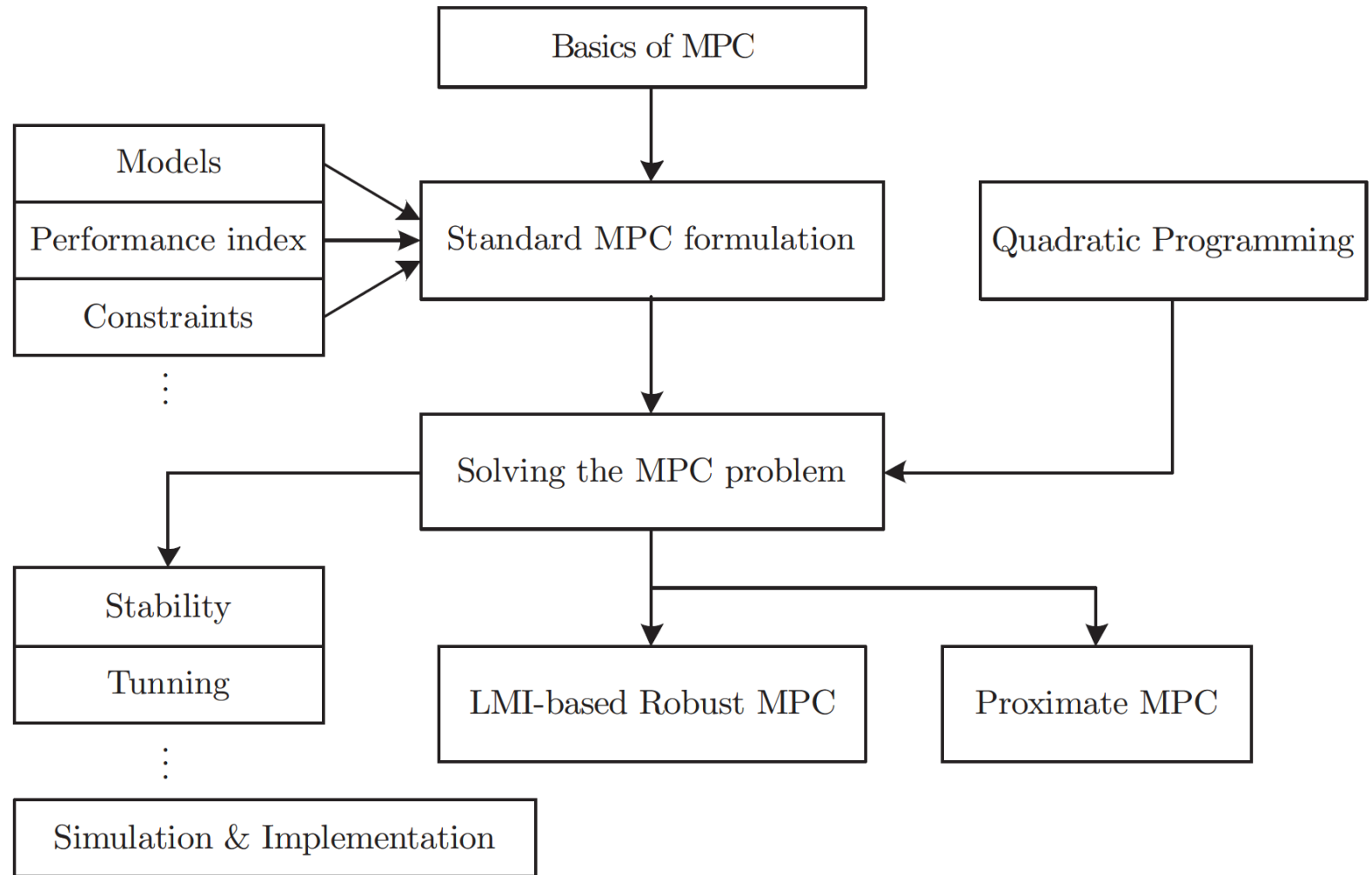
Both types of algorithms are convex and show fast convergence.

# Receding horizon principle

- at time  $k$ : Performance index  $J(k)$  is minimized
- first element of optimal control sequence  $v(k)$  is applied to the system
- horizon shifted
- optimization restarted for time  $k + 1$



# *Model predictive control*



# Quadratic program

The general Quadratic program (QP) can be stated as

$$\begin{aligned} \min_x J(x) &= \frac{1}{2}x^T H x + x^T f \\ \text{subject to} \quad &A_i x = b_i, \quad i \in \mathcal{S}_{eq} \\ &A_i x \leq b_i, \quad i \in \mathcal{S}_{ineq} \end{aligned}$$

where

$H = H^T$   $n_x$  by  $n_x$  matrix,

$f, x, A_i$  vectors with  $n_x$  elements

$\mathcal{S}_{eq}$  and  $\mathcal{S}_{ineq}$  set of equality and inequality constraints, respectively

$i \in \mathcal{S}_{eq} \cup \mathcal{S}_{ineq}$

# *QP unconstrained*

---

The objective function  $J$

$$J = \frac{1}{2}x^T Hx + x^T f$$

where

$H = H^T > 0$  and  $f$  are compatible matrix and vector

The necessary conditions for minimizing the objective function with equality constraints

$$\frac{\partial J}{\partial x} = Hx + f = 0$$

$$x^* = -H^{-1}f$$

# *QP for equality constraints*

The objective function  $J$  and the constraints are expressed as

$$J = \frac{1}{2}x^T Hx + x^T f$$
$$Ax = b$$

where

$H = H^T > 0$ ,  $f$ ,  $A$  and  $b$  are compatible matrices and vectors

$$A = \begin{bmatrix} \vdots \\ A_i \\ \vdots \end{bmatrix} \in \mathbb{R}^{n_b \times n_x}, \quad b = \begin{bmatrix} \vdots \\ b_i \\ \vdots \end{bmatrix} \in \mathbb{R}^{n_b}$$

# Lagrange multiplier

$$\mathcal{L}(x, \lambda) = f(x) - \lambda g(x)$$

An objective function in  $n_x + n_b$  variables  $x$  and  $\lambda$

$$J = \frac{1}{2}x^T Hx + x^T f + \lambda^T (Ax - b)$$

The necessary conditions for minimizing the objective function with equality constraints

$$\frac{\partial J}{\partial x} = Hx + f + A^T \lambda = 0$$

$$\frac{\partial J}{\partial \lambda} = Ax - b = 0$$

$$\lambda^* = -(AH^{-1}A^T)^{-1}(b + AH^{-1}f)$$

$$x^* = -H^{-1}(f + A^T \lambda^*) = x^* - H^{-1}A^T \lambda^* = -H^{-1}f - H^{-1}A^T \lambda^*$$

Note: Correction by active constraints



# *Lagrange Multipliers*

---

Note:

Need  $n_{b,eq}$  (# equality constraints)  $\leq n_x$  (# decision variables  $x$ ) for feasibility

If  $n_{b,eq} = n_x$ , the only feasible solution is the one that satisfies the constraints.

If  $n_{b,eq} > n_x$ , then there is no feasible solution to satisfy the constraints, infeasible

# *QP inequality constraints*

The objective function  $J$  and the constraints are expressed as

$$J = \frac{1}{2}x^T Hx + x^T f$$
$$Ax \leq b$$

where

$H = H^T > 0$ ,  $f$ ,  $A$  and  $b$  are compatible matrices and vectors

$$A = \begin{bmatrix} \vdots \\ A_i \\ \vdots \end{bmatrix} \in \mathbb{R}^{n_b \times n_x}, \quad b = \begin{bmatrix} \vdots \\ b_i \\ \vdots \end{bmatrix} \in \mathbb{R}^{n_b}$$

# *QP inequality constraints*

---

In the minimization with inequality constraints,

it is possible to have  $n_b$  (# constraints)  $\geq n_x$  (# decision variables)

The inequality constraints  $Ax \leq b$  may comprise active constraints and inactive constraints.

An inequality  $A_i x \leq b_i$  is said to be

active if  $A_i x = b_i$  and

inactive if  $A_i x < b_i$

# *KKT conditions*

---

**Karush-Kuhn-Tucker conditions:** Let  $x$  be a local minimum point for the problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & h(x) = 0, \quad g(x) \leq 0 \end{aligned}$$

and suppose  $x$  is a regular point for the constraints ( $\nabla h(x)$  and  $\nabla g(x)$  are linearly independent). Then there is a vector  $\lambda$  and a vector  $\mu \geq 0$  such that

$$\begin{aligned} \nabla f(x) + \mu^T \nabla h(x) + \lambda^T \nabla g(x) &= 0 \\ \lambda^T g(x) &= 0 \end{aligned}$$

# *KKT conditions*

---

QP for inequality constraints:

$$f(x) = \frac{1}{2}x^T Hx + f^T x, \quad g(x) = Ax - b \leq 0$$

$$\nabla f(x) = \nabla \left( \frac{1}{2}x^T Hx + f^T x \right) = x^T H + f^T$$

$$\nabla g(x) = \nabla (Ax - b) = A$$

$$Hx + f + A^T \lambda = 0$$

$$\lambda^T (Ax - b) = 0$$

$$Ax - b \leq 0$$

$$\lambda \geq 0$$

# *KKT conditions*

**KKT conditions:** The necessary condition for the QP problem with inequality constraints

$$Hx + f + A^T \lambda = 0$$

$$\lambda^T (Ax - b) = 0$$

$$Ax - b \leq 0$$

$$\lambda \geq 0$$

Note: Complementary slackness condition

Since  $\lambda \geq 0$  and  $Ax - b \leq 0$ ,  $\lambda^T (Ax - b) = 0$  is equivalent to the statement that

- $\lambda_i$  may be nonzero only if  $A_i x - b_i = 0$  (active)
- $\lambda_i > 0$  implies  $A_i x - b_i = 0$  (active)
- $A_i x - b_i < 0$  (inactive) implies  $\lambda_i = 0$

# *KKT conditions*

The Kuhn-Tucker conditions define the active and inactive constraints in terms of the Lagrange multipliers:

$\lambda_i \geq 0$ : an active constraint,  $A_i x - b_i = 0 \quad i \in \mathcal{S}_{act}$

$\lambda_i = 0$ : an inactive constraint,  $A_i x - b_i < 0 \quad i \in \mathcal{S}_{inact}$

Note:  $A^T \lambda = \sum_{i \in \mathcal{S}_{act}} \lambda_i A_i^T = A_{act}^T \lambda_{act}$

If the active set were known, the original problem could be replaced by the corresponding problem having equality constraints only:

Given  $A_{act}$  and  $b_{act}$ , the optimal solution has the closed-form (recall (2.3)):

$$\lambda_{act} = -(A_{act} H^{-1} A_{act}^T)^{-1} (b_{act} + A_{act} H^{-1} f)$$

$$x = -H^{-1} (f + A_{act}^T \lambda_{act})$$

# Active set method

Solution by iteration

At each iteration step of an algorithm,

- \_ Define the working set - a set of active constraints at the current point

$$\mathcal{S}_{act}^{(q)} \subseteq \mathcal{S}_{eq} \cup \mathcal{S}_{ineq}$$

- \_ Redefine the working set  $\mathcal{S}_{act}^{(q)}$  if necessary: While moving on the working surface, add new constraints to the working set, if a new constraint boundary is encountered.
- \_ Solve the equality constraint problem.

If all the Lagrange multipliers  $\lambda_i \geq 0$ , then the point is a local soln. to the original problem.

If there exists  $\lambda_i < 0$ , relax the constraint described by  $A_i$  and  $b_i$ : Delete it from the constraint equation:  $A^T \lambda = \sum_{i \in \mathcal{S}_{act}} \lambda_i A_i^T$



# *Dual problem*

---

In the active set methods, the active constraints need to be identified along with the optimal decision variables.

Programming an active method is not straightforward.

If there are many constraints, the computational load is quite large.

Dual problem utilizes dual variables (Lagrangian multipliers)  
to systematically identify the constraints that are not active that can then be eliminated.

⇒ very simple programming procedures for finding optimal solutions of  
constrained minimization problems.

# Dual problem

Assuming feasibility (i.e., there is an  $x$  such that  $Ax < b$ ), the primal problem is equivalent to

$$\max_{\lambda \geq 0} \min_x \left( \frac{1}{2} x^T H x + x^T f + \lambda^T (Ax - b) \right)$$

The minimization over  $x$  is unconstrained and is attained by

$$x = -H^{-1}(f + A^T \lambda)$$

Substituting this leads to

$$\max_{\lambda \geq 0} \left( -\frac{1}{2} \lambda^T \bar{H} \lambda - \lambda^T \bar{f} - \frac{1}{2} f^T H^{-1} f \right)$$

Equivalently

$$\min_{\lambda \geq 0} \left( \frac{1}{2} \lambda^T \bar{H} \lambda + \lambda^T \bar{f} + \frac{1}{2} b^T H^{-1} b \right)$$

where

$$\bar{H} = AH^{-1}A^T, \quad \bar{f} = AH^{-1}f + b$$

# Dual problem

The dual objective function

$$J = \frac{1}{2} \lambda^T \bar{H} \lambda + \lambda^T \bar{f} + \frac{1}{2} b^T H^{-1} b$$

The dual problem  $\min_{\lambda \geq 0} J$  is also a QP problem with  $\lambda$  as the decision variable.

The dual problem may be much easier to solve than the primal problem because the constraints  $\lambda \geq 0$  are simpler

With the set of optimal Lagrange multipliers  $\lambda_{act}$  with corresponding constraints (treated as equality constraints) described by  $A_{act}$  and  $b_{act}$ , the primal variable vector  $x$  is obtained from

$$x = -H^{-1} f - H^{-1} A_{act}^T \lambda_{act}$$

Equivalently, with  $\lambda_i = 0$  for inactive constraints,

$$x = -H^{-1} f - H^{-1} A^T \lambda$$

# Dual problem

The dual variables will converge only if

the active constraints are linearly independent and

$$n_{b,act} \text{ (\# active constraints)} \leq n_x \text{ (\# decision variables)}.$$

When the conditions are satisfied, the one-dimensional search converges to the set of  $\lambda^*$ :

$$\lambda_{inact}^* = 0 \text{ for inactive constraints}$$

$$\lambda_{act}^* \geq 0 \text{ for active constraints}$$

The positive component collected as a vector is called  $\lambda_{act}^*$  with its value defined by

$$\lambda_{act}^* = - \left( A_{act} H^{-1} A_{act}^T \right)^{-1} \left( b_{act} + A_{act} H^{-1} f \right)$$

The existence of a set of bounded  $\lambda_{act}^*$  is determined by the existence of  $\left( A_{act} H^{-1} A_{act}^T \right)^{-1}$ , which assures the convergence in the convex problem.

# Dual problem

The dual objective function

$$J = \frac{1}{2} \lambda^T \bar{H} \lambda + \lambda^T \bar{f} + \frac{1}{2} b^T H^{-1} b$$

The necessary condition

$$\begin{aligned} \frac{\partial J}{\partial \lambda} &= \bar{H} \lambda + \bar{f} = 0 \\ \lambda &\geq 0 \end{aligned}$$

Given  $\lambda^*$ , we have

$$x^* = -H^{-1}(f + A^T \lambda^*)$$

Note:

$$\bar{H} = A H^{-1} A^T = [\bar{H}_{i,j}] = [A_i H^{-1} A_j^T] \geq 0$$

$$\bar{H}_{i,i} = A_i H^{-1} A_i^T > 0 \text{ for any } A_i \neq 0$$

# Models

---

State space models:

- especially suited for MIMO systems
- compact model description
- compact and low order controller
- computations are well conditioned
- algorithms easy to implement

$$G(q) = \left[ \begin{array}{c|c} \Phi & \Gamma \\ \hline C & D \end{array} \right] = C(qI - \Phi)^{-1}\Gamma + D \quad : \quad \begin{cases} x(k) = \Phi x(k) + \Gamma u(k) \\ y(k) = Cx(k) + Du(k) \end{cases}$$
$$y(k) = G(q)u(k)$$

# Models

- Input-Output (IO) Model:

$$y(k) = G_{io}(q)u(k), \quad G_{io}(q) \text{ strictly proper}$$

$$x_{io}(k+1) = \Phi_{io}x_{io}(k) + \Gamma_{io}u(k)$$

$$y(k) = C_{io}x_{io}(k)$$

- Increment-Input-Output (IIO) Model:

$$y(k) = G_{iio}(q)\Delta u(k), \quad G_{iio}(q) \text{ strictly proper}$$

$$x_{iio}(k+1) = \Phi_{iio}x_{iio}(k) + \Gamma_{iio}\Delta u(k)$$

$$y(k) = C_{iio}x_{iio}(k)$$

where

$$\Delta u(k) = (1 - q^{-1})u(k) = u(k) - u(k-1)$$

# Models

## Relation IO Model and IIO model:

Introduce a new state vector

$$x_{iio} = \begin{bmatrix} y(k-1) \\ \Delta x_{io} \end{bmatrix}$$

and define the system matrices

$$\Phi_{iio} = \begin{bmatrix} I & C_{io} \\ 0 & \Phi_{io} \end{bmatrix}, \quad \Gamma_{iio} = \begin{bmatrix} 0 \\ \Gamma_{io} \end{bmatrix}, \quad C_{iio} = \begin{bmatrix} I & C_{io} \end{bmatrix}$$

then

$$G_{iio}(q) = G_{io}(q)(1 - q^{-1})^{-1}$$



# Models

## Advantage of Using IIO models:

e.g. 1: Given an IO system with  $\|G_{io}(1)\| < \infty$ , minimize an IO-performance index

$$J(k) = \sum_{i=0}^{N-1} \|y(k+i) - r(k+i)\|^2 + \lambda^2 \|u(k+i)\|^2$$

Let  $r_k = r_{ss} \neq 0$  for  $k \rightarrow \infty$ .

Then using  $y_{ss} = G_{io}(1)u_{ss}$  we find

$$\begin{aligned} J_{ss} &= \|y_{ss} - r_{ss}\|^2 + \lambda^2 \|u_{ss}\|^2 \\ &= \|G_{io}(1)u_{ss} - r_{ss}\|^2 + \lambda^2 \|u_{ss}\|^2 \\ &= u_{ss}^T (G_{io}^T(1)G_{io}(1) + \lambda^2 I) u_{ss} - 2u_{ss}^T G_{io}^T(1)r_{ss} + r_{ss}^T r_{ss} \end{aligned}$$

# Models

---

Minimizing  $J_{ss}$  over  $u_{ss}$  means that

$$u_{ss} = (G_{io}^T(1)G_{io}(1) + \lambda^2 I)^{-1} G_{io}^T(1)r_{ss}$$

$$y_{ss} = G_{io}(1) (G_{io}^T(1)G_{io}(1) + \lambda^2 I)^{-1} G_{io}^T(1)r_{ss}$$

It is clear that  $y_{ss} \neq r_{ss}$  for  $\lambda > 0$

There will be a steady-state error

# Models

e.g. 2: Given an IIO system  $G_{iio}(q) = G_{io}(q)(1 - q^{-1})^{-1}$  with  $\|G_{io}(1)\| < \infty$ .

Minimize an IIO-performance index

$$J(k) = \sum_{i=0}^{N-1} \|y(k+i) - r(k+i)\|^2 + \lambda^2 \|\Delta u(k+i)\|^2$$

Let  $r(k) = r_{ss} \neq 0$  for  $k \rightarrow \infty$ .

$$J_{ss} = \|y_{ss} - r_{ss}\|^2 + \lambda^2 \|\Delta u_{ss}\|^2$$

Minimum  $J_{ss} = 0$  is reached for  $u_{ss} = G_{io}^{-1}(1)r_{ss}$  which gives  $y_{ss} = r_{ss}$  and  $\Delta u_{ss} = 0$ .

It is clear that  $y_{ss} = r_{ss}$  for any  $\lambda > 0$ .

There will be no steady-state error

# Models

Standard state space model:

$$x(k+1) = \Phi x(k) + \Gamma_1 w(k) + \Gamma_2 v(k)$$

$$z(k) = C_1 x(k) + D_{11} w(k) + D_{12} v(k)$$

$$y(k) = C_2 x(k) + D_{21} w(k) + D_{22} v(k)$$

Assumptions:

- $(\Phi, \Gamma_2)$  controllable,  $(\Phi, C_2)$  observable
- $(\Phi, C_1)$  observable,  $(\Phi, \Gamma_1)$  controllable
- $D_{21} = 0, D_{22} = 0$
- $C_1^T D_{12} = 0$

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

# Models

The controlled variable:

$$z(k) = \begin{bmatrix} z_1(k) \\ z_2(k) \end{bmatrix} = \begin{bmatrix} \eta(k) \\ D_{12}v(k) \end{bmatrix} = \begin{bmatrix} C_1x(k) \\ D_{12}v(k) \end{bmatrix}, \quad D_{11} = 0$$

$$z^T(k)z(k) = z_1^T(k)z_1(k) + z_2^T(k)z_2(k) = x^T(k)C_1^TC_1x(k) + v^T(k)D_{12}^TD_{12}v(k)$$

For tracking a reference trajectory  $r(\cdot)$

$$z(k) = \begin{bmatrix} z_1(k) \\ z_2(k) \end{bmatrix} = \begin{bmatrix} C_1x(k) - r(k) \\ D_{12}v(k) \end{bmatrix}, \quad D_{11}w(k) = -C_1x_r(k) = -r(k)$$

$$z^T(k)z(k) = z_1^T(k)z_1(k) + z_2^T(k)z_2(k) = (C_1x(k) - r(k))^T (C_1x(k) - r(k)) \\ + v^T(k)D_{12}^TD_{12}v(k)$$

# Models

## Lifting prediction and control:

- prediction

$$\tilde{x}(k) = \begin{bmatrix} x(k) \\ \vdots \\ x(k + N_p - 1) \end{bmatrix}, \quad \tilde{\eta}(k) = \begin{bmatrix} \eta(k) \\ \vdots \\ \eta(k + N_p - 1) \end{bmatrix}, \quad \tilde{z}(k) = \begin{bmatrix} z(k) \\ \vdots \\ z(k + N_p - 1) \end{bmatrix}$$

- control

$$\tilde{v}(k) = \begin{bmatrix} v(k) \\ \vdots \\ v(k + N_c - 1) \end{bmatrix}$$

# Models

---

## Optimization:

The MPC controller minimizes the standard performance index at time  $k$ :

$$J = \sum_{i=0}^{N_p-1} z^T(k+i|k) \Omega(i) z(k+i|k)$$

where

$$z(k+i|k) = \begin{bmatrix} z_1(k+i|k) \\ z_2(k+i|k) \end{bmatrix}$$
$$\Omega(i) = \text{diag}(\Omega_1(i), \Omega_2(i))$$