

## 1 System Theory

### 1.1 Nonlinear Systems

tim: Put here linearization formulae

### 1.2 Linear Systems

$$\dot{x}(t) = A^c x(t) + B^c u(t)$$

$$y(t) = C^c x(t) + D^c u(t)$$

$$x(t) = e^{A^c(t-t_0)} x_0 + \int_{t_0}^t e^{A^c(t-\tau)} B^c u(\tau) d\tau$$

$$e^{A^c t} = \sum_{n=0}^{\infty} \frac{(A^c t)^n}{n!}$$

$$x_{k+N} = A^N x_k + \sum_{i=0}^{N-1} A^i B u_{k+N-1-i}$$

### 1.3 Lyapunov Stability

we define a system to be stable in the sense of Lyapunov, if it stays in any arbitrarily small neighborhood of the origin when it is disturbed slightly

**Lyapunov stable** if for every  $\epsilon > 0$  there exists a  $\delta(\epsilon)$  such that

$$\text{norm } x(0) < \delta(\epsilon) \rightarrow \text{norm } x(k) < \epsilon, \forall k \geq 0$$

**asymptotically stable** in  $\Omega \subseteq \mathbb{R}^n$  if it is Lyapunov stable and attractive  $\lim_{k \rightarrow \infty} x(k) = 0, \forall x(0) \in \Omega$ .

**Lyapunov Function**  $V : \mathbb{R}^n \rightarrow \mathbb{R}$  must be continuous at the origin, finite  $\forall x \in \Omega$  and:

$$V(0) = 0 \text{ and } V(x) > 0, \forall x \in \Omega \setminus \{0\}$$

$$V(g(x)) - V(x) \leq -\alpha(x), \forall x \in \Omega \setminus \{0\}$$

where  $\alpha : \mathbb{R}^n \rightarrow \mathbb{R}$  is continuous positive definite, equilibrium at  $x = 0$  and  $\Omega \subset \mathbb{R}^n$  closed and bounded set containing the origin.

**Lyapunov Theorem** If a system admits Lyapunov function  $V(x)$ , then  $x = 0$  is **asymptotically stable** in  $\Omega$  (sufficient but not necessary). If additionally  $\|x\| \rightarrow \infty \Rightarrow V(x) \rightarrow \infty$ , then  $x = 0$  is **globally asymptotically stable**.

To check if  $V(x) = x^T P x$  is valid Lyapunov function of system  $x_{k+1} = A x_k$  check if  $(A^T P A - P)$  has neg. eigen values.

In other words: If eigenvalues of  $A$  inside unit circle (i.e. stable) then  $\exists$  unique  $P > 0$  that solves  $A_{cl}^T P A_{cl} - P = -Q, Q > 0$  and  $V(x) = x^T P x$  is a lyapunov function.

### 1.4 Discretization

Euler:  $A = I + T_s A^c, B = T_s B^c, C = C^c, D = D^c$

$$x_{k+1} = x_k + T_s g^c(x_k, u_k) = g(x_k, u_k)$$

$$y_k = h^c(x_k, u_k) = h(x_k, u_k)$$

Exact: (assume constant  $u(t)$  during  $T_s$ )

$$A = e^{A^c T_s}, B = \int_0^{T_s} e^{A^c(T_s-\tau')} B^c d\tau$$

$$B = (A^c)^{-1} (A - I) B^c, \text{ if } A^c \text{ invertible}$$

### 1.5 Controllability (reachability) and observability

$$C = [B \ AB \ \dots \ A^{n-1}B]$$

$$O = [C^T \ (CA)^T \ \dots \ (CA^{n-1})^T]$$

## 2 Unconstrained Control

### 2.1 Block Approach (used also for $\bar{w}$ substitution)

$$\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} I \\ A \\ \vdots \\ A^N \end{bmatrix} x(0) + \begin{bmatrix} 0 & 0 & \dots & 0 \\ B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & \dots & AB & B \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

$$x = S^x \cdot x(0) + S^u \cdot u \quad \text{size}(S^x) = [n_{\text{states}} \cdot (N+1), N]$$

$$\text{size}(S^u) = [n_{\text{states}} \cdot (N+1), n_{\text{states}}]$$

$$\bar{Q} = \text{diag}(Q, \dots, Q, P) \quad \text{size}(\bar{Q}) = [n_{\text{states}} \cdot (N+1), n_{\text{states}} \cdot (N+1)]$$

$$\bar{R} = \text{diag}(R, \dots, R) \quad \text{size}(\bar{R}) = [n_{\text{input}} \cdot N, n_{\text{input}} \cdot N]$$

$$H = S^{uT} \bar{Q} S^u + R \quad F = S^{xT} \bar{Q} S^u$$

$$Y = S^{xT} \bar{Q} S^x$$

### Optimal cost and control

$$J^*(x_0) = -x_0^T F H F^T x_0 + x_0^T Y x_0$$

$$u^*(x_0) = -H^{-1} F^T x_0 = -\left(S^{uT} \bar{Q} S^u + R\right)^{-1} S^{uT} \bar{Q} S^x x_0$$

### 2.2 Recursive Approach

$$J_k^*(x_k) = \min_{u_k} I(x_k, u_k) + J_{k+1}(x_{k+1})$$

Is a feedback controller as opposed to the Batch Approach. For LQR solve via Riccati Difference Equation (RDE).

$$F_k = -(B^T P_{k+1} B + R)^{-1} B^T P_{k+1} A$$

$$P_k = A^T P_{k+1} A + Q - A^T P_{k+1} B (B^T P_{k+1} B + R)^{-1} B^T P_{k+1} A$$

$$u_k^* = F_k x_k \quad J_k^*(x_k) = x_k^T P_k x_k \quad P_N = P$$

For unconstrained Infinite Horizon Problem, substituting  $P_\infty = P_k = P_{k+1}$  into RDE gives DARE. Uniquely solvable, iff  $(A, B)$  stabilizable and  $(A, G)$  detectable, where  $G G^T = Q$ . Follows from closed-loop system  $x_{k+1} = (A + B F_k) x_k$

## 3 (Convex) Optimization

**General Problem**  $\min_{x \in \text{dom}(f)} f(x)$  s. t.  $g_i(x) \leq 0$  and  $h_j(x) = 0$ .

**Norm**  $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$

$$f(x) = 0 \Rightarrow x = 0,$$

$$f(x) \geq 0$$

$$f(\alpha \cdot x) = |\alpha| \cdot f(x)$$

$$\text{for scalar } \alpha$$

$$f(x+y) \leq f(x) + f(y)$$

$$\forall x, y \in \mathbb{R}^n$$

### 3.1 Convexity

**Convex set**  $\mathcal{X}$  iff  $\forall \lambda \in [0, 1] \forall x, y \in \mathcal{X} \lambda x + (1 - \lambda)y \in \mathcal{X}$ .

Intersection preserves convexity, union does not.

**Affine set**  $\mathcal{X} = \{x \in \mathbb{R}^n | A x = b\}$  for some  $A, b$

**Subspace** is affine set through origin, i.e.  $b = 0$ , aka Nullspace of  $A$ .

**Hyperplane**  $\mathcal{X} = \{x \in \mathbb{R}^n | a^T x = b\}$  for some  $a, b$ .

**Halfspace**  $\mathcal{X} = \{x \in \mathbb{R}^n | a^T x \leq b\}$  for some  $a, b$ .

**Polyhedron**  $\mathcal{P} = \{x | a_i^T x \leq b_i, i = 1, \dots, n\} = \{x | A x \leq b\}$

**Cone**  $\mathcal{X}$  if for all  $x \in \mathcal{X}$ , and for all  $\theta > 0, \theta x \in \mathcal{X}$ .

**Ellipsoid**  $\mathcal{E} = \{x | (x - x_c)^T A^{-1} (x - x_c) \leq 1\}, x_c$  center point.

**Convex function**  $f : \text{dom}(f) \rightarrow \mathbb{R}$  is convex iff  $\text{dom}(f)$  is convex and  $f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y), \forall \lambda \in (0, 1), \forall x, y \in \text{dom}(f)$ .

**Norm ball** is convex (for all norms).

**Epigraph set**  $f : \text{dom}(f) \rightarrow \mathbb{R}$  is the set

$$\text{epi}(f) := \left\{ \begin{bmatrix} x \\ t \end{bmatrix} \mid x \in \text{dom}(f), f(x) \leq t \right\} \subseteq \text{dom}(f) \times \mathbb{R}$$

**Level set**  $L_a$  of a function  $f$  for value  $a$  is the set of all  $x \in \text{dom}(f)$  for which  $f(x) = a$ :  $L_a = \{x | x \in \text{dom}(f), f(x) = a\}$ .

**Sublevel set**  $C_a$  is defined by  $C_a = \{x | x \in \text{dom}(f), f(x) \leq a\}$ .

### 3.2 Linear Programming (LP)

**Problem statement**  $\min c^T x$  such that  $Gx \leq h$  and  $Ax = b$ .

**Norm  $l_\infty$**   $\min_x \|x\|_\infty = \min_{x \in \mathbb{R}^n} [\max\{x, \dots, x_n, -x_1, \dots, -x_n\}]$ :

$$\begin{aligned} \min_{x,t} t \quad & \text{subject to} \quad x_i \leq t, -x_i \leq t, \quad \mathbf{F}x \leq g \\ \iff \min_{x,t} t \quad & \text{subject to} \quad -\mathbf{1}t \leq x \leq \mathbf{1}t, \quad \mathbf{F}x \leq g. \end{aligned}$$

**Norm  $l_1$**   $\min_x \|x\|_1 = \min_x [\sum_{i=1}^m \max\{x_i, -x_i\}]$ :

$$\begin{aligned} \min_t t_1 + \dots + t_m \quad & \text{subject to} \quad x_i \leq t_i, -x_i \leq t_i, \quad \mathbf{F}x \leq g \\ \iff \min_t \mathbf{1}^T t \quad & \text{subject to} \quad -t \leq x \leq t, \quad \mathbf{F}x \leq g. \end{aligned}$$

Note that for  $\dim x = 1$ ,  $l_1$  and  $l_\infty$  are the same. Note also that  $t$  is scalar for norm  $l_\infty$  and a vector in norm  $l_1$ .

### Piecewise Affine

$$\begin{aligned} \min_x \left[ \max_{i=1, \dots, m} \{c_i^T x + d_i\} \right] \quad & \text{s.t. } \mathbf{G}x \leq h \\ \iff \min_{x,t} t \quad & \text{s.t. } c_i^T x + d_i \leq t, \mathbf{G}x \leq h \end{aligned}$$

### 3.3 Duality

#### Lagrangian Dual Function

$$\begin{aligned} L(x, \lambda, \nu) &= f(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{i=1}^p \nu_i h_i(x) \\ d(\lambda, \nu) &= \inf_{x \in \mathcal{X}} L(x, \lambda, \nu) \quad \text{i.e. } \nabla_x L(x, \lambda, \nu) = 0 \end{aligned}$$

**Dual Problem (always convex)**  $\max_{\lambda, \nu} d(\lambda, \nu)$  s. t.  $\lambda \geq 0$ .

Optimal value is lower bound for primal:  $d^* \leq p^*$ .

If primal convex, **Slater condition** (strict feasibility) implies *strong duality*:  $\{x \mid Ax = b, f_i(x) < 0, \} \neq \emptyset \Rightarrow d^* = p^*$

**Karush-Kuhn-Tucker (KKT) Conditions** are necessary for optimality (and sufficient if primal convex).

Primal Feasibility	$f_i(x^*) \leq 0$	$i = 1, \dots, m$
	$h_i(x^*) = 0$	$i = 1, \dots, p$
Dual Feasibility	$\lambda^* \geq 0$	
Complementary slackness	$\lambda_i^* \cdot f_i(x^*) = 0$	$i = 1, \dots, m$
Stationarity	$\nabla_x L(x^*, \lambda^*, \nu^*) = 0$	

### Dual of LP

$$\begin{aligned} \min_x c^T x \quad & \text{s.t. } \mathbf{A}x = b, \mathbf{C}x \leq e \\ \iff \max_{\lambda, \nu} -b^T \nu - e^T \lambda \quad & \text{s.t. } A^T \nu + C^T \lambda + c = 0, \lambda \geq 0 \end{aligned}$$

### Dual of QP

$$\begin{aligned} \min_x \frac{1}{2} x^T \mathbf{Q}x + c^T x \quad & \text{s.t. } \mathbf{C}x \leq e \\ \iff \min_{\lambda, \nu} \frac{1}{2} \lambda^T \mathbf{C} \mathbf{Q}^{-1} \mathbf{C}^T \lambda + (\mathbf{C} \mathbf{Q}^{-1} c + e)^T \lambda + \frac{1}{2} c^T \mathbf{Q}^{-1} c \\ \text{s.t. } \mathbf{Q}x + \nu + c^T \lambda = 0, \lambda \geq 0 \end{aligned}$$

## 4 Constrained Finite Time Optimal Control (CFTOC)

### 4.1 MPC with linear cost

$$J(x_0, u) = \|\mathbf{P}x_N\|_p + \sum_{i=0}^{N-1} \|\mathbf{Q}x_i\|_p + \|\mathbf{R}u_i\|_p.$$

tim: Insert here slide 45, lect 4  $l_g l_\infty$ ??

tim: what is n, k, m?

$$\begin{aligned} \min_z \quad & \epsilon_0^x + \dots + \epsilon_N^x + \epsilon_0^u + \dots + \epsilon_{N-1}^u \\ \text{s.t.} \quad & -\mathbf{I}_n \epsilon_i^x \leq \pm \mathbf{Q} \left[ \mathbf{A}^i x_0 + \sum_{j=0}^{i-1} \mathbf{A}^j \mathbf{B} u_{i-1-j} \right] \\ & -\mathbf{I}_k \epsilon_N^x \leq \pm \mathbf{P} \left[ \mathbf{A}^N x_0 + \sum_{j=0}^{N-1} \mathbf{A}^j \mathbf{B} u_{N-1-j} \right] \\ & -\mathbf{I}_m \epsilon_N^u \leq \pm \mathbf{R} u_i \\ & x_i = \mathbf{A}^i x_0 + \sum_{j=0}^{i-1} \mathbf{A}^j \mathbf{B} u_{i-1-j} \in \mathcal{X} \\ & x_N = \mathbf{A}^N x_0 + \sum_{j=1}^{N-1} \mathbf{A}^j \mathbf{B} u_{N-1-j} \in \mathcal{X} \\ & u_i \in \mathcal{U} \end{aligned}$$

Bring into LP form:

$$\begin{aligned} \min_z \quad & c^T z \quad \text{s.t. } \bar{\mathbf{G}}z \leq \bar{w} + \bar{s}x(k) \\ c &= [1 \quad \dots \quad 1 \quad 1 \quad \dots \quad 1 \quad 0 \quad \dots \quad 0] \\ z &= [\epsilon_0^x \quad \dots \quad \epsilon_N^x \quad \epsilon_0^u \quad \dots \quad \epsilon_{N-1}^u] \\ \bar{\mathbf{G}} &= \begin{bmatrix} \mathbf{G}_\epsilon & ?? \\ 0 & ?? \end{bmatrix}, \quad \bar{w} = \begin{bmatrix} w_\epsilon \\ w \end{bmatrix} \\ \bar{s} &= \begin{bmatrix} s_\epsilon \\ s \end{bmatrix} \end{aligned}$$

### 4.2 QP with substitution (see also Batch approach)

$$\begin{aligned} J^*(x_k) &= \min_u \begin{bmatrix} u^T & x_k^T \end{bmatrix} \begin{bmatrix} \mathbf{H} & \mathbf{F}^T \\ \mathbf{F} & \mathbf{Y} \end{bmatrix} \begin{bmatrix} u \\ x_k \end{bmatrix} \\ \text{s. t. } \mathbf{G} u &\leq w + \mathbf{E} x_k \end{aligned}$$

Latter gives three sets (same for without substitution)

$$\begin{aligned} \mathcal{X} &= \{x \mid A_x x \leq b_x\} \\ \mathcal{U} &= \{u \mid A_u u \leq b_u\} \\ \mathcal{X}_f &= \{x \mid A_f x \leq b_f\} \end{aligned}$$

State equations are in cost matrix, usually

$$\mathbf{A}_x = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, b_x = \begin{bmatrix} b_{\max} \\ -b_{\min} \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{A}_u & 0 & \dots & 0 \\ 0 & \mathbf{A}_u & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{A}_u \\ 0 & 0 & \dots & 0 \\ \mathbf{A}_x \mathbf{B} & 0 & \dots & 0 \\ \mathbf{A}_x \mathbf{A} \mathbf{B} & \mathbf{A}_x \mathbf{B} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_x \mathbf{A}^{N-2} \mathbf{B} & \mathbf{A}_x \mathbf{A}^{N-3} \mathbf{B} & \dots & 0 \\ \mathbf{A}_f \mathbf{A}^{N-1} \mathbf{B} & \mathbf{A}_f \mathbf{A}^{N-2} \mathbf{B} & \dots & \mathbf{A}_f \mathbf{B} \end{bmatrix} \quad \mathbf{E} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -\mathbf{A}_x \\ -\mathbf{A}_x \mathbf{A} \\ -\mathbf{A}_x \mathbf{A}^2 \\ \vdots \\ -\mathbf{A}_x \mathbf{A}^{N-1} \\ -\mathbf{A}_f \mathbf{A}^N \end{bmatrix} \quad \mathbf{W} = \begin{bmatrix} b_{uu} \\ b_{uu} \\ \vdots \\ b_{ux} \\ b_{ux} \\ \vdots \\ b_{ux} \\ b_{fx} \\ b_{fx} \end{bmatrix}$$

### 4.3 QP without substitution

State equations represented in equality constraints ( $k$  fixed, usually  $k = 0$ ).

$$\begin{aligned} J^*(x_k) &= \min_z \begin{bmatrix} z^T & x_k^T \end{bmatrix} \begin{bmatrix} \bar{\mathbf{H}} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} \end{bmatrix} \begin{bmatrix} z \\ x_k \end{bmatrix} \\ \text{s.t. } \mathbf{G} z &\leq w + \mathbf{E} x_k \\ \mathbf{G}_{\text{eq}} z &= \mathbf{E}_{\text{eq}} x_k, \quad \text{system dynamics} \end{aligned}$$

$$\bar{\mathbf{H}} = \text{diag}(\mathbf{Q}, \dots, \mathbf{Q}, \mathbf{P}, \mathbf{R}, \dots, \mathbf{R})$$

$$\begin{aligned} z &= \begin{bmatrix} x_1 \\ x_N \\ u_0 \\ \vdots \\ u_{N-1} \end{bmatrix} \quad \mathbf{G}_{\text{eq}} = \left[ \begin{array}{c|c} \mathbf{I} & -\mathbf{B} \\ -\mathbf{A} & \mathbf{I} \end{array} \middle| \begin{array}{c} -\mathbf{B} \\ -\mathbf{B} \end{array} \right] \quad \mathbf{E}_{\text{eq}} = \begin{bmatrix} \mathbf{A} \\ 0 \\ 0 \end{bmatrix} \\ w &= \begin{bmatrix} b_x \\ b_f \\ b_u \\ \vdots \\ b_u \end{bmatrix} \quad \mathbf{G} = \left[ \begin{array}{c|c} \mathbf{A}_x & \\ \hline & \mathbf{A}_d \end{array} \right] \quad \mathbf{E} = \begin{bmatrix} -\mathbf{A}_x^T \\ 0 \\ 0 \end{bmatrix} \end{aligned}$$

### 4.4 Invariance

Def.:  $x(k) \in O \Rightarrow x(k+1) \in O \forall k$ .

$$\text{pre}(S) := \{x \mid g(x) \in S\} = \{x \mid Ax \in S\}$$

Max invariant set calculation:  $\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$ , terminating when  $\Omega_{i+1} = \Omega_i$ .

tim: We need more here, pos. inv. set, max. pos.inv  $O_\infty$

### 4.5 Stability and Feasibility

Main Idea: Choose  $\mathcal{X}_f$  and  $\mathbf{P}$  to mimic infinite horizon. LQR control law  $\kappa(x) = \mathbf{F}_\infty x$  from solving DARE. Set terminal weight  $\mathbf{P} = \mathbf{P}_\infty$ , terminal set  $\mathcal{X}_f$  as maximal invariant set:

$$\begin{aligned} x_{k+1} &= \mathbf{A}x_k + \mathbf{B}\mathbf{F}_\infty x_k \in \mathcal{X}_f \quad \forall x_k \in \mathcal{X}_f \text{ terminal set invariant} \\ \mathcal{X}_f &\subseteq \mathcal{X}, \quad \mathbf{F}_\infty x_k \in \mathcal{U} \quad \forall x_k \in \mathcal{X}_f \text{ constraint satisfied} \end{aligned}$$

We get: 1. Positive stage cost function, 2. invariant terminal set by construction, 3. Terminal cost is Lyapunov function with

$$x_{k+1}^T \mathbf{P} x_{k+1} - x_k^T \mathbf{P} x_k = -x_k^T (\mathbf{Q} + \mathbf{F}_\infty^T \mathbf{R} \mathbf{F}_\infty) x_k$$

Extension to non-linear (time-invariant) MPC possible since terminal set and cost do not rely on linearity.

## 5 Practical Issues

### 5.1 MPC for tracking

Target steady-state conditions  $x_s = \mathbf{A}x_s + \mathbf{B}u_s$  and  $y_s = \mathbf{C}x_s = r$  and constraints give:

$$\min_{x_s, u_s} u_s^T \mathbf{R} u_s \text{ subj. to } \begin{bmatrix} \mathbf{I} - \mathbf{A} & -\mathbf{B} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ r \end{bmatrix}, x_s \in \mathcal{X}, u_s \in \mathcal{U}$$

Usually assume  $x_s, u_s$  unique and feasible. If no solution exists, compute closest steady-state  $\min(\mathbf{C}x_s - r)^T \mathbf{Q}(\mathbf{C}x_s - r)$  s. t.  $x_s = \mathbf{A}x_s + \mathbf{B}u_s$ .

MPC problem to drive  $y \rightarrow r$  is:

$$\min_u \|y_N - \mathbf{C}x_s\|_{P_y}^2 + \sum_{i=0}^{N-1} \|y_i - \mathbf{C}x_s\|_{Q_y}^2 + \|u_i - u_s\|_R^2$$

### 5.2 Delta formulation

Reference  $r$ ,  $\Delta x_k = x_k - x_s$ ,  $\Delta u_k = u_k - u_s$ :

$$\min V_f(\Delta x_N) + \sum_{i=0}^{N-1} \Delta x_i^T \mathbf{Q} \Delta x_i + \Delta u_i^T \mathbf{R} \Delta u_i$$

$$\text{s.t. } \Delta x_0 = \Delta x_k$$

$$\Delta x_{k+1} = \mathbf{A} \Delta x_k + \mathbf{B} \Delta u_k$$

$$\mathbf{H}_x x \leq k_x \Rightarrow \mathbf{H}_x \Delta x \leq k_x - \mathbf{H}_x x_s$$

$$\mathbf{H}_u u \leq k_u \Rightarrow \mathbf{H}_u \Delta u \leq k_u - \mathbf{H}_u u_s$$

$$\Delta x_N \in \mathcal{X}_f \text{ adjusted accordingly, shift (and scaled)}$$

$$x_s \oplus \mathcal{X}_f \subseteq \mathcal{X}$$

$$\mathbf{K} \Delta x + u_s \in \mathcal{U}$$

Control given by  $u_0^* = \Delta u_0^* + u_s$ .

### 5.3 Offset free tracking

$$x_{k+1} = \mathbf{A}x_k + \mathbf{B}u_k + \mathbf{B}_d d_k$$

$$d_{k+1} = d_k$$

$$y_k = \mathbf{C}x_k + \mathbf{C}_d d_k$$

$$\begin{bmatrix} \mathbf{I} - \mathbf{A} & -\mathbf{B} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} \mathbf{B}_d \hat{d} \\ r - \mathbf{C}_d \hat{d} \end{bmatrix}$$

Choice of  $\mathbf{B}_d, \mathbf{C}_d$  requires that  $(\mathbf{A}, \mathbf{C})$  is observable and

$\begin{bmatrix} \mathbf{A} - \mathbf{I} & \mathbf{B}_d \\ \mathbf{C} & \mathbf{C}_d \end{bmatrix}$  has full  $(n_x + n_d)$  column rank (i.e.  $\det \neq 0$ ).

Intuition: for fixed  $y_s$  at steady-state,  $d_s$  is uniquely determined. If plant has no integrator we can choose  $\mathbf{B}_d = \mathbf{0}$  since  $\det(\mathbf{A} - \mathbf{I}) \neq 0$ .

$$\begin{bmatrix} \hat{x}_{k+1} \\ \hat{d}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B}_d \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \hat{d}_k \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} u_k + \begin{bmatrix} \mathbf{L}_x \\ \mathbf{L}_d \end{bmatrix} (-y_k^m + \mathbf{C} \hat{x}_k + \mathbf{C}_d \hat{d}_k)$$

where  $y_k^m$  measured output; choose  $\begin{bmatrix} \mathbf{L}_x \\ \mathbf{L}_d \end{bmatrix}$  s.t. error dynamics stable and converge to zero.

tim: Target condition here

If 1) number of dist. = number of outputs, 2) target steady-state problem feasible and no constraints active at steady-state, 3) closed-loop system converges, then target achieved without offset.

### 5.4 Soft-constraints via slack variables

$$\min_x f(z) + l_\epsilon(\epsilon) \quad \text{s.t. } g(z) \leq \epsilon, \epsilon \geq 0$$

Requirement: Softened problem has same minimiser as original problem if feasible.

Quadratic error function  $l_\epsilon(\epsilon) = v\epsilon + w\epsilon^2$ ,  $w > 0$  gives smoothness, choose  $v > \lambda^* \geq 0$  for exact penalty (above requirement fulfilled).

### 5.5 Move Blocking

Main idea to set a number of inputs as the same,  $u_2 = u_3 = \dots = u_N$ , to reduce computational burden, at the slight cost of sub-optimality.

## 6 Robust MPC

Enforcing terminal constraints by robust invariance:

$$x \in O^W \Rightarrow g(x, w) \in \Omega^W \quad \forall w \in \mathcal{W}$$

$$\text{pre}^W(\Omega) = \{x | g(x, w) \in \Omega \quad \forall w \in \mathcal{W}\}$$

tim: Maybe an example from exercises to compute  $O_\infty^W$

Enforcing sequential constraints for uncertain system  $\phi$ :

$$\phi_i(x_0, u, w) = \left\{ x_i + \sum_{j=0}^{i-1} \mathbf{A}^j w_j \mid w \in \mathcal{W}^i \right\} \subseteq \mathcal{X}$$

$$\phi_N(x_0, u, w) \in \mathcal{X}_f \text{ as well}$$

The uncertain system evolves with the summation of all the disturbances up to time  $i$ , hence we have to restrict the open-loop (determine control before disturbance is measured):

$$\mathbf{A}_x x \leq b_x \text{ becomes } \mathbf{A}_x x_i + \mathbf{A}_x \sum_{j=0}^{i-1} \mathbf{A}^j w_j \leq b_x :$$

$$\begin{aligned} x_i &\in \mathcal{X} \ominus (\mathcal{W} \oplus \mathbf{A}\mathcal{W} \oplus \dots \oplus \mathbf{A}^{i-1}\mathcal{W}) \\ &= \left( \bigoplus_{j=0}^{i-1} \mathbf{A}^j \mathcal{W} \right) = [\mathbf{A}^0 \quad \dots \quad \mathbf{A}^{i-1}] \mathcal{W}^i \end{aligned}$$

For example: Robust invariant set calculation of  $x_{k+1} = 0.5x_k + w_k$  under  $-10 \leq x \leq 10$  and  $-1 \leq w \leq 1$ .

$$\Omega_0 = [-10, 10]$$

$$\begin{aligned} \text{pre}^W(\Omega_0) &= \{x | -10 \leq 0.5x + w \leq 10 \text{ for } -1 \leq w \leq 1\} \\ &= \{x | -20 - 2w \leq x \leq 20 + 2w \text{ for } -1 \leq w \leq 1\} \\ &= \{x | -18 \leq x \leq 18\} \end{aligned}$$

$$\Omega_1 = [-10, 10] \cap [-18, 18] = [-10, 10] = \mathcal{O}_\infty^W$$

**Tube-MPC** We want nominal system  $z_k = \mathbf{A}z_k + \mathbf{B}v_k$  with "tracking" controller  $u_k = \mathbf{K}(x_k - z_k) + v_k$  i.e. closed-loop,  $\mathbf{K}$  found offline.

Step 1: Compute  $\mathcal{E} = \bigoplus_{j=1}^\infty \mathbf{A}_{cl}^j \mathcal{W}$ .

Step 2: Shrink Constraints:

$$\begin{aligned} \{z_i\} \oplus \mathcal{E} &\subseteq \mathcal{X} && \Rightarrow z_i \in \mathcal{X} \ominus \mathcal{E} \\ u_i \in \mathbf{K}\mathcal{E} \oplus \{v_i\} &\subset \mathcal{U} && \Rightarrow v_i \in \mathcal{U} \ominus \mathbf{K}\mathcal{E} \\ z_n \in \mathcal{X}_f \ominus \mathcal{E} &&& \end{aligned}$$

## 7 Explicit MPC

$z^*(x_k)$  is continuous and polyhedral piecewise affine over feasible set.

### 7.1 Quadratic Cost

$J^*(x_k)$  is continuous, convex and polyhedral piecewise quadratic. Using the formulation with substitution,

$$J(x_k) = \min z^T H z - x_k^T (Y - F H^{-1} F^T) x_k$$

$$\text{s.t. } Gz \leq w + Sx_k$$

$$z(x_k) = U + H^{-1} F^T x_k$$

$$S = E + G H^{-1} F^T$$

$$U^* = z^*(x_k) - H^{-1} F^T x_k$$

The first solution gives  $u^*(x_k) = \kappa(x_k)$ , which is continuous and piecewise affine on polyhedra  $\kappa(x) = F_j x + g_j$ .

### 7.2 $1/\infty$ -norm

$J^*(x_k)$  is continuous, convex and polyhedral piecewise affine.

Optimal solution:  $u_0^* = [0 \quad \dots \quad 0 \quad I_m \quad 0 \quad \dots \quad 0] z^*(x_k)$ , and is in the same form as the QP case above.

### 7.3 Explicit Example

- Write out KKT conditions and Lagrangian.
- Determine infeasible regions from primal feasibility constraints. For example,  $x_1 < 10$ .
- From primal and dual feasibility, and complementary slackness conditions, list out all cases that can occur.
 

$\lambda_1 = 0$	$\lambda_1 \geq 0$
$g_1(x) < 0$	$g_1(x) = 0$
- Solve for each case:  $z^*(x_1, x_2)$  and  $J^*(x_1, x_2)$ , listing the active constraints, and range of validity.

## 8 Hybrid MPC

### 8.1 Piecewise Affine (PWA) Systems

Affine dynamics and output for each region:

$$\begin{cases} x_{k+1} &= \mathbf{A}^i x_k + \mathbf{B}^i u_k + \mathbf{f}^i \\ y_k &= \mathbf{C}^i x_k + \mathbf{D}^i u_k + \mathbf{g}^i \end{cases} \text{ if } x_t \in \mathcal{X}_j$$

Polyhedral partition of the  $(x, u)$ -space:

$$\{\mathcal{X}_i\}_{j=1}^s = \{x, u | H_j x + J_j u \leq K_j\}$$

## 8.2 Mixed Logical Dynamical Hybrid Model (MLD)

Idea: associate boolean to binary:  $p_j \iff \delta_i = 1, \neg p_j \iff \delta_i = 0$ .

### 8.2.1 Translate logic rules to Linear Integer Inequalities

AND	$p_1 \wedge p_2$	$\delta_1 \geq 1, \delta_1 \geq 1$ also $\delta_1 + \delta_2 \geq 2$
OR	$p_1 \vee p_2$	$\delta_1 + \delta_2 \geq 1$
NOT	$\neg p_1$	$1 - \delta_1 \geq 1$ also $\delta_1 = 0$
XOR	$p_1 + p_2$	$\delta_1 + \delta_2 = 1$
IMPLY	$p_1 \rightarrow p_2$	$\delta_1 - \delta_2 \leq 0$
IFF	$p_1 \leftrightarrow p_2$	$\delta_1 - \delta_2 = 0$
ASSIGNMENT	$p_3 \leftrightarrow p_1 \wedge p_2$	$\delta_1 + (1 - \delta_3) \geq 1,$ $\delta_2 + (1 - \delta_3) \geq 1,$ $(1 - \delta_1) + (1 - \delta_2) + \delta_3 \geq 1$
CNF-Clause	$\neg p_1 \vee \neg p_2 \vee p_3$	$\delta_1 + \delta_2 + \delta_3 \leq 1$

### Logic Equality Rules

$$\begin{aligned} \neg(A \wedge B) &= \neg A \vee \neg B \\ A \wedge (B \vee C) &= (A \wedge B) \vee (A \wedge C) \\ A \vee (B \wedge C) &= (A \vee B) \wedge (A \vee C) \end{aligned}$$

### 8.2.2 Translate continuous and logical components into Linear Mixed-Integer Relations

Event generator:  $\delta_e(k) = f_{EG}(x_c(k), u_c(k), t)$ .

Consider:  $p \iff a^T x \leq b, \mathcal{X} = \{x | a^T x - b \in [m, M]\}$ .

Translated to linear inequalities:  $m\delta < a^T x - b \leq M(1 - \delta)$ , where  $[m, M]$  are lower and upper bounds.

### Representing Switched Affine Dynamics as "IF-THEN-ELSE" relations

IF p THEN  $z_k = a_1^T x_k + b_1$  else  $z_k = a_2^T x_k + b_2 \iff$

$$\begin{aligned} (m_2 - M_1)\delta + z_k &\leq a_2^T x_k + b_2 \leq -(m_1 - M_2)\delta + z_k \\ (m_1 - M_2)(1 - \delta) + z_k &\leq a_1^T x_k + b_1 \leq -(m_1 - M_2)(1 - \delta) + z_k \end{aligned}$$

This results in a linear MLD model

$$\begin{aligned} x_{k+1} &= Ax_k + B_1 u_k + B_2 \delta_k + B_3 z_k \\ y_k &= Cx_k + D_1 u_k + D_2 \delta_k + D_3 z_k \\ E_2 \delta_k + E_3 z_k &\leq E_4 x_k + E_1 u_k + E_5 \end{aligned}$$

where the last equation describes the relationship between the continuous and integer variables. Physical constraints on cont.

variables:  $\{\mathcal{C}\} = \left\{ \begin{bmatrix} x_c \\ u_c \end{bmatrix} \in \mathcal{R}^{n_c+m_c} | Fx_c + Gu_c \leq H \right\}$

### 8.3 CFTOC for Hybrid Systems

$$\begin{aligned} J^*(x) &= \min_U l_N(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k, \delta_k, z_k) \\ \text{s.t. } x_{k+1} &= Ax_k + B_1 u_k + B_2 \delta_k + B_3 z_k \\ E_2 \delta_k + E_3 z_k &\leq E_4 x_k + E_1 u_k + E_5 \\ x_N &\in \mathcal{X}_f, x_0 = x(0) \end{aligned}$$

## 8.4 MILP/QP

$$\begin{aligned} \min \quad & c_c z_c + c_b z_b + d \quad \text{OR} \quad zHz + qz + d \\ \text{s.t.} \quad & G_c z_c + G_b z_b \leq W \\ & z_c \in R^{s_c}, z_b \in \{0, 1\}^{s_b} \end{aligned}$$

Branch and bound method can be used to efficiently solve the problem. Explicit solution is a time varying fb law for both problems:  $u_k^*(x_k) = F_k^j x_k + G_k^j$  if  $x_k \in \mathcal{R}_k^j$ .

## 9 Numerical Optimization – Iterative Methods

### 9.1 Gradient descent

$x_{i+1} = x_i - h_i \nabla f(x_i)$  with step-size  $h_i = \frac{1}{L}$  for  $L$ -smooth  $f(x)$ :

$$\begin{aligned} \exists L : \|\nabla f(x) - \nabla f(y)\| &\leq L\|x - y\| \quad \forall x, y \in \mathbb{R}^n \\ \iff \nabla f &\text{ is Lipschitz continuous} \\ \iff f &\text{ can be upperbounded by a quadratic function} \end{aligned}$$

### 9.2 Newton's Method

$x_{i+1} = x_i - h_i \delta x_{nt} := x_i - h_i (\nabla^2 f(x_i))^{-1} \nabla f(x_i)$   
Choose  $h_i > 0$  s.t.  $f(x_i + h_i \delta x_{nt}) \leq f(x_i)$  **Line-search**.  
For given equality constraint  $Ax = b$  solve:

$$\begin{bmatrix} \nabla^2 f(x_i) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \delta x \\ \lambda \end{bmatrix} = \begin{bmatrix} -\nabla f(x_i) \\ 0 \end{bmatrix}$$

### 9.3 Constrained optimization

$g_i(x) \leq 0$  with

**Gradient method**  $x_{i+1} = \pi_Q(x_i - h_i \nabla f(x_i))$  where  $\pi_Q$  is a projection  $\pi_Q = \arg \min_x \frac{1}{2} \|x - y\|_2^2$ . Projection can be solved directly if simple enough, else solve the dual.

### 9.4 Interior-Point methods

Assumptions  $f(x^*) < \infty, \tilde{x} \in \text{dom}(f)$ .

**Barrier method**  $\min f(x) + \kappa \phi(x)$ . Approximate  $\phi$  using diff'able log barrier (instead of indicator function):

$$\begin{aligned} \phi(x) &= \sum_{i=1}^m I_{-}(g_i(x)) = - \sum_{i=1}^m \log(-g_i(x)) \\ \lim_{\kappa \rightarrow 0} x^*(\kappa) &= x^* \end{aligned}$$

Analytic center:  $\arg \min_x \phi(x)$ , central path  $\{x^*(\kappa) | \kappa > 0\}$ .

### Path following method

- Centering  $x^*(\kappa) = \arg \min_x f(x) + \kappa \phi(x)$  with newton's method:
  - $\Delta x_{nt} = [\nabla^2 f(x) + \kappa \nabla^2 \phi(x)]^{-1} (-\nabla f(x) - \kappa \nabla \phi(x))$ .
  - Line search:

$$\begin{aligned} \text{retain feasibility: } & \arg \max_{h>0} \{h|g_i(x + h\Delta x) < 0\} \\ \text{Find } h^* &= \arg \min_{h \in [0, h_{\max}]} \{f(x + h\Delta x) + \kappa \phi(x + h\Delta x)\} \end{aligned}$$

- Update step  $x_i = x^*(\kappa_i)$
- Stop if  $m\kappa_i \leq \epsilon$
- Decrease  $\kappa_{i+1} = \kappa_i / \mu, \mu > 1$ .

## Centering step with equality constraints

$$\begin{bmatrix} \nabla^2 f(x) + \kappa \nabla^2 \phi(x) & c^T \\ c & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{nt} \\ \nu \end{bmatrix} = - \begin{bmatrix} \nabla f(x) + \kappa \nabla \phi(x) \\ 0 \end{bmatrix}$$

### Relaxed KKT

$$\begin{aligned} Cx^* &= d & g_i(x^*) + s_i^* &= 0 \\ \nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) + c^T \nu &= 0 & \lambda_i^* &= \kappa \frac{\partial \phi}{\partial g_i} = -\frac{\kappa}{g_i} \\ \lambda_i^* g_i(x^*) &= -\kappa & \lambda_i^*, s_i^* &\geq 0 \end{aligned}$$

### Primal Dual Search Direction Computation

$$\begin{bmatrix} H(x, \lambda) & c^T & G(x)^T & 0 \\ c & 0 & 0 & 0 \\ G(x) & 0 & 0 & I \\ 0 & 0 & S & \Lambda \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta u \\ \Delta \lambda \\ \Delta s \end{bmatrix} = - \begin{bmatrix} \nabla f(x) + c^T \nu + G(x)^T \lambda \\ cx - d \\ g(x) + s \\ s\lambda - \nu \end{bmatrix}$$

$S = \text{diag}(s_1, \dots, s_m), \Lambda = \text{diag}(\lambda_1, \dots, \lambda_m)$  and  $\nu$  is a vector for choosing centering parameters.