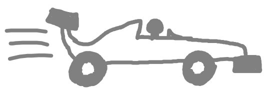
# MPC model predictive control



(Relevant Theory recap)

## CT state-space systems

- **non-linear system:** $\dot{x}(t)=g(x(t),u(t))$ $\quad y(t)=h(x(t),u(t))$ $\quad$ ($x$: state $\;u$: input $\;y$: output)
  - linearization: $A=\begin{bmatrix}\frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n}\\ \vdots & & \vdots\\ \frac{\partial g_n}{\partial x_1} & \cdots & \frac{\partial g_n}{\partial x_n}\end{bmatrix}_{\substack{x=x_e\\u=u_e}}$ $\quad B=\begin{bmatrix}\frac{\partial g_1}{\partial u_1} & \cdots & \frac{\partial g_1}{\partial u_m}\\ \vdots & & \vdots\\ \frac{\partial g_n}{\partial u_1} & \cdots & \frac{\partial g_n}{\partial u_m}\end{bmatrix}_{\substack{x=x_e\\u=u_e}}$ $\quad C=\begin{bmatrix}\frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n}\\ \vdots & & \vdots\\ \frac{\partial h_p}{\partial x_1} & \cdots & \frac{\partial h_p}{\partial x_n}\end{bmatrix}_{\substack{x=x_e\\u=u_e}}$ $\quad D=\begin{bmatrix}\frac{\partial h_1}{\partial u_1} & \cdots & \frac{\partial h_1}{\partial u_m}\\ \vdots & & \vdots\\ \frac{\partial h_p}{\partial u_1} & \cdots & \frac{\partial h_p}{\partial u_m}\end{bmatrix}_{\substack{x=x_e\\u=u_e}}$

  $\boxed{\begin{aligned}\dot{x}&=Ax+Bu\\ y&=Cx+Du\end{aligned}}$

- **linear system:** $\dot{x}(t)=Ax(t)+Bu(t)$ $\quad y(t)=Cx(t)+Du(t)$ $\quad$ ($x$: state $\;u$: input $\;y$: output)
  - discretization:

  $\boxed{\begin{aligned}x(k+1)&=A_d x(k)+B_d u(k)\\ y(k)&=C_d x(k)+D_d u(k)\end{aligned}}$

  direct method: $A_d=e^{AT}=\sum_{i=0}^{\infty}(AT)^k\frac{1}{k!}$ $\quad C_d=C$ $\quad B_d=\int_0^T e^{AT}d\tau\cdot B=A^{-1}(A_d-I)B$ $\quad D_d=D$

  euler forward method: $A_d=I+T\cdot A$ $\quad C_d=C$ $\quad B_d=T\cdot B$ $\quad D_d=D$

  ($k$: timestep $\;T$: sampling t $\;t=kT$)

## DT state-space systems

- **linear system:** $x(k+1)=\underline{A}x(k)+\underline{B}u(k)$ ; $y(k)=\underline{C}x(k)+\underline{D}u(k)$ $\quad$ ($x$: state $\;u$: input $\;y$: output)
  - asymp. stability: $\lim_{k\to\infty}x(k)=0\;\forall x(0)$ $\longleftrightarrow$ $|\lambda_i|<1\;\forall\lambda_i=EW(\underline{A})$ $\quad$ (any $x(0)$ will reach equilibrium ($x=0$) with $u=0$)
  - controllability: $\mathrm{rank}(B\;AB\;\cdots\;A^{n-1}B)=n$ $\quad$ ($n$: sys. order) $\quad$ (any $x^*$ can be reached from any $x(0)$ with correct $u$)
  - observability: $\mathrm{rank}(C^T\;(CA)^T\cdots(CA^{n-1})^T)=n$ $\quad$ ($n$: sys. order) $\quad$ (state $x(0)$ can be deduced from $y$)
  - pot. stabilizability: $\mathrm{rank}((\lambda_i I-A)\,\vdots\,B)=n\quad\forall|\lambda_i|\geq1$ $\quad$ ($\lambda_i=EW(A)$) $\quad$ (uncontrollable modes are stable)
  - detectability: $\mathrm{rank}((A^T-\lambda_i I)\,\vdots\,C^T)=n\quad\forall|\lambda_i|\geq1$ $\quad$ ($\lambda_i=EW(A)$) $\quad$ (unobservable modes are stable)

- **non-linear system:** $x(k+1)=g(x(k),u(k))$ ; $y(k)=h(x(k),u(k))$ $\quad$ (equilibrium at $\bar{x}$: $\bar{x}=g(\bar{x},0)$)
  - asymp. stability: $\lim_{k\to\infty}\|x(k)-\bar{x}\|=0\quad\forall x(0)\in\Omega$ $\quad$ (state goes to equilibrium when starting at $x(0)$ and $u=0$)
  - global asympt. stab.: asympt. stability, but for all $x(0)\in\Omega\in\mathbb{R}^n$
  - lyapunov stability: $\|x(k)-\bar{x}\|<\epsilon\in\mathbb{R}\quad\forall k,x(0)\in\Omega$ $\quad$ (state doesn't go to inf. when starting at $x(0)$ and $u=0$)

## Stability with lyapunov function

lyapunov function: $V(x):\mathbb{R}^n\to\mathbb{R}$ with $V(0)=0$ ; $V(x)>0\;\forall x\in\Omega\setminus\{0\}$ $\quad$ (some sort of "Energy" function)

↳ good candidates to try: $V(x)=x^TPx$ $\quad$ (where $P$ positive definite matrix)

- CT nonlinear: $\dot{x}(t)=f(x(t))$ $\quad$ (eq. at $x=0$) $\to$ $\alpha(x)=\frac{d}{dt}V(x(t))=\sum_{i=0}^{n}\frac{\partial V}{\partial x_i}\cdot\frac{\partial x_i}{\partial t}=(\nabla V(x))^T f(x)$
- DT nonlinear: $x(k+1)=f(x(k))$ $\quad$ (eq. at $x=0$) $\to$ $\alpha(x)=V(f(x(k)))-V(x(k))\approx\dot{V}(x)$ $\quad$ (euler forward discretization)
  - if $\alpha(x)\leq0\quad\forall x\in\Omega\setminus\{0\}$ $\to$ lyapunov stable in $\Omega$
  - if $\alpha(x)<0\quad\forall x\in\Omega\setminus\{0\}$ $\to$ asympt. stable in $\Omega$
  - if $\alpha(x)<0\quad\forall x\in\mathbb{R}^n\setminus\{0\}$ and $\lim_{\|x\|\to\infty}V(x)\to\infty$ $\to$ globally asympt. stable
  - else $\longrightarrow$ no conclusion. Try other $V(x)$ maybe?

(DT linear example: $x(k+1)=A\cdot x(k)$ $\quad$ (eq. at $x=0$) ; $V(x)=x^TPx$ $\to$ $\alpha(x)=(Ax)^TP(Ax)-x^TPx=x^T(A^TPA-P)x$

globally asympt. stable if $-(A^TPA-P)$ positive definite $\leftrightarrow$ $|\lambda_i|<1\;\forall i$ with $\lambda_i=EW(A)$

CT linear example: $\dot{x}(t)=A\cdot x(t)$ $\quad$ (eq. at $x=0$) ; $V(x)=x^TPx$ $\to$ $\alpha(x)=\nabla(x^TPx)^TAx=(2Px)^TAx=x^T((2P)^TA)x$

globally asympt. stable if $-((2P)^TA)$ positive definite $\leftrightarrow$ $Re(\lambda_i)<0\;\forall i$ with $\lambda_i=EW(A)$)

## Optimization Problem

minimize$_x$ $f(x)$ subject to : $\begin{cases} x \in \mathrm{dom}(f) \\ g_i(x) \leq 0 & i=1,\dots,m \\ h_i(x) = 0 & i=1,\dots,p \end{cases}$

$x=(x_1,\dots,x_n)$: optimization variables
$f:\mathrm{dom}(f)\to\mathbb{R}$: objective function
$g_i:\mathbb{R}^n\to\mathbb{R}$: inequality constraints (optional)
$h_i:\mathbb{R}^n\to\mathbb{R}$: equality constraints (optional)

$\mathcal{X}=\{x\in\mathrm{dom}(f) \mid g_i(x)\leq 0, h_i(x)=0\}$: set of feasible points
- $\mathcal{X}=\mathbb{R}^n$: unconstrained problem
- $\bar{\mathcal{X}}=\{\mathcal{X} \mid g_i(x)<0\}$: set of strictly feasible points

$p^*=\min f(x)$ subj to $x\in\mathcal{X}$ : optimal value
- $p^*=-\infty$: unbounded below
- $p^*=\infty$: infeasible $(\mathcal{X}=\{\})$

$x^*=\mathrm{argmin}\, f(x)$ subj to $x\in\mathcal{X}$ : optimizer (multiple points)

if $g_i(x^*)=0$ constraint is active else inactive

a constraint is redundant if it does not change $\mathcal{X}$

### convex optimization problem:

convex set: $\lambda x+(1-\lambda)y\in\mathcal{X}$ , $\forall\lambda\in[0,1], \forall x,y\in\mathcal{X}$ (every point between 2 points $\in\mathcal{X}$ is also $\in\mathcal{X}$)
- open/closed halfspace: $\{x\in\mathbb{R}^n \mid a^Tx \ (\leq \text{ or } <)\ b\}$ (set of all points on one side of a plane $\underline{a}$: normal, $b$: dist. from $\underline{0}$)
- polyhedron: intersection of a finite set of halfspaces. polytope: a bounded polyhedron.
- ellipsoid: $\{x\in\mathbb{R}^n \mid (x-x_c)^TA(x-x_c)\leq 1\}$ $(A>0)$ ($x_c$: center, $\sqrt{EW(A^{-1})}$: semi-axis lengths, $EV(A^{-1})$: semi-axis dir.)
- norm balls: $\{x\in\mathbb{R}^n \mid \|x-x_c\|_p\leq r)\}$ ($x_c$: center, $r$: radius, $p$: norm) ($\|x\|_2=\sqrt{\sum x_i^2}$ ; $\|x\|_1=\sum|x_i|$ ; $\|x\|_\infty=\max(x_i)$ ; ...)
- range: $\{x\in\mathbb{R} \mid x_L\leq x\leq x_h\}=[x_L,x_h]$ ($x_L$: lower bound, $x_h$: higher bound)

convex function: $f(x):\mathrm{dom}(f)\to\mathbb{R}$ convex iff $\mathrm{dom}(f)$: convex and $f(\lambda x+(1-\lambda)y)\leq\lambda f(x)+(1-\lambda)f(y)$ , $\forall\lambda\in[0,1]$, $\forall x,y\in\mathrm{dom}(f)$

strictly convex func.: like convex but with $<$ not $\leq$ concave function: $f(x):\mathrm{dom}(f)\to\mathbb{R}$ concave iff $-f(x)$ is convex

level set: $L_\alpha=\{x \mid x\in\mathrm{dom}(f), f(x)=\alpha\}$ sublevel set: $C_\alpha=\{x \mid x\in\mathrm{dom}(f), f(x)\leq\alpha\}$ (these are convex)
- first-order condition: $f(y)\geq f(x)+\nabla f(x)^T(y-x)$ $\forall x,y\in\mathrm{dom}(f)$ $(\nabla f(x)=[\partial f/\partial x_1,\dots,\partial f/\partial x_n]^T)$
- second-order condition: $\nabla^2 f(x)\geq 0$ $\forall x\in\mathrm{dom}(f)$ $(\nabla^2 f(x)_{ij}=\partial^2 f/\partial x_i\partial x_j)$
- examples: $f(x)=ax+b$ ; $f(x)=e^{ax}$ ; $f(x)=x^a$ $\forall a\leq 0, a\geq 1$ ; $f(x)=\|x\|_p$ $\forall p\geq 1$ ; $f(x)=\frac{1}{2}x^THx+q^Tx$ $\forall H>0$

convex problem: $\min_x f(x)$ subj. to $g_i(x)\leq 0$ $i=1,..,m$ where $f$: convex function
$h_i(x)=0$ $i=1,..,p$ $g_i\leq 0, h_i=0$: convex sets $\leftrightarrow$ $g_i,h_i$: convex functions

(consequence: feasibility set $\mathcal{X}$ is convex ; local optima are global optima)
- general linear program (LP): $\min c^Tx$ subj. to $Gx\leq h$ and $Ax=b$ $\to$ $\mathcal{X}$ is a polyhedron!
- general quadratic program (QP): $\min \frac{1}{2}x^THx+q^Tx$ subj. to $Gx\leq h$ and $Ax=b$ $(H>0)$

### dual optimization problem: (dual problem always convex even if primal problem is not!)

Lagrangian function: $L(x,\lambda,\nu)=f(x)+\sum_{i=1}^m\lambda_ig_i(x)+\sum_{i=1}^p\nu_ih_i(x)$ $(L:\mathrm{dom}(f)\times\mathbb{R}^m\times\mathbb{R}^p\to\mathbb{R})$

Lagrange dual func.: $d(\lambda,\nu)=\min_x L(x,\lambda,\nu)$ subj. to: nothing (for some $\lambda,\nu$ take smallest $L$) $\to$ $x^*=\mathrm{func}(\lambda,\nu)$

dual problem: $d^*=\max_{\lambda,\nu} d(\nu,\lambda)$ subj. to: $\lambda\geq 0$ $\to$ $\lambda^*,\nu^*$: Lagrange multipliers ; $\underline{x^*=\mathrm{func}(\lambda^*,\nu^*)}$

- weak duality: $d^*\leq p^*$ always holds
  ↳ $x^*$ = some local minimum
- strong duality: $d^*=p^*$ if convex problem and $\underbrace{\text{strictly feasible points } \bar{\mathcal{X}}\neq\{\}}_{\text{Slater's condition}}$
  ↳ $x^*$ = global minimum

### KKT conditions: (equivalent to solving dual optim. problem)

1) primal feasibility: $g_i(x^*)\leq 0$ $(i=1,..,m)$ and $h_i(x^*)=0$ $(i=1,..,p)$
2) dual feasibility: $\lambda_i^*\geq 0$ $(i=1,..,m)$
3) complementary slackness: $\lambda_i^*g_i(x^*)=0$ $(i=1,..,m)$
4) stationarity: $\nabla_xL(x^*,\lambda^*,\nu^*)=\nabla f(x^*)+\sum_{i=1}^m\lambda_i^*\nabla g(x^*)+\sum_{i=1}^p\nu_i^*\nabla h_i(x^*)=0$

- necessary conditions on $(x^*,\lambda^*,\nu^*)$ given weak duality.
- necessary + sufficient conditions on $(x^*,\lambda^*,\nu^*)$ given strong duality.

# Linear Quadratic Optimal control

( quadratic cost ; linear system → find input to minimize cost)

## • finite horizon:

$$J(x(0),\underline{U}) = x_N^T P x_N + \sum_{i=0}^{N-1} (x_i^T Q x_i + u_i^T R u_i)$$

( $J$: total cost  $Q>0$: state weight  $N$: time horizon
$P>0$: terminal weight  $R>0$: input weight  $U:(u(0),\dots,u(N))$: inputs )

(linear system: $x_{k+1} = Ax_k + Bu_k$)

→ find $U^*$ that minimizes $J(x(0),U)$

### – batch approach:

$$\underbrace{\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix}}_{X} = \underbrace{\begin{bmatrix} I \\ A \\ \vdots \\ A^{N-1} \\ A^N \end{bmatrix}}_{S^x} x(0) + \underbrace{\begin{bmatrix} 0 & 0 & \cdots & 0 \\ B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ A^{N-1}B & \cdots & AB & B \end{bmatrix}}_{S^u} \cdot \underbrace{\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix}}_{U}$$

$$X = S^x \cdot x(0) + S^u \cdot U$$

$J(x(0),U) = X^T\bar{Q}X + U^T\bar{R}U$  $(\bar{Q}=\mathrm{diag}(Q,\dots,Q,P),\ \bar{R}=\mathrm{diag}(R,\dots,R))$

$J(x(0),U) = U^THU + 2x(0)FU + x(0)^TYx(0)$  $\left(\begin{aligned} H &= S^{uT}\bar{Q}S^u + \bar{R} \\ F &= S^{xT}\bar{Q}S^u \\ Y &= S^{xT}\bar{Q}S^x \end{aligned}\right)$

$\underset{U}{\operatorname{argmin}}\ J$: $\nabla_U J = 2HU + 2F^Tx(0) = 0$

↳ $U^*(x(0)) = -H^{-1}F^Tx(0)$

$J^*(x(0)) = \dots = x(0)^T(Y - FH^{-1}F^T)x(0)$

### – recursive approach:

$J_N^*(x_N) = x_N^T P_N x_N$  $(P_N = P)$  ($J_i^*(x_i)$: optimal cost for steps $i$ to $N$)

$J_{N-1}^*(x_{N-1}) = \min\limits_{u_{N-1}} x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1} + x_N^T P_N x_N$

↓ (substit. $x_N = Ax_{N-1} + Bu_{N-1}$ and solve)

$J_{N-1}^*(x_{N-1}) = x_{N-1}^T P_{N-1} x_{N-1}$ ; $u_{N-1}^* = F_{N-1}x_{N-1}$

$J_{N-2}^*(x_{N-2}) = x_{N-2}^T P_{N-2} x_{N-2}$ ; $u_{N-2}^* = F_{N-2}x_{N-2}$

↓ etc.  ↓ etc.

$J_0^*(x_0) = x_0^T P_0 x_0$ ; $u_0^* = F_0 x_0$

$$\begin{cases} P_i = A^TP_{i+1}A + Q - A^TP_{i+1}B(B^TP_{i+1}B+R)^{-1}B^TP_{i+1}A \\ F_i = -(B^TP_{i+1}B+R)^{-1}B^TP_{i+1}A \\ x_{i+1} = Ax_i + Bu_i \end{cases}$$

algorithm:

$$\begin{tikzcd}
P_N \arrow[r] \arrow[dr] & P_{N-1} \arrow[r] \arrow[dr] & P_{N-2} \arrow[r] & \cdots \arrow[r] \arrow[dr] & P_1 \arrow[r] \arrow[dr] & P_0 \\
 & F_{N-1} \arrow[d] & F_{N-2} \arrow[d] & \cdots & F_1 \arrow[d] & F_0 \arrow[d] \\
 & u^*_{N-1} & u^*_{N-2} \arrow[dl] & \cdots & u^*_1 \arrow[dl] & u^*_0 \arrow[dl] \\
 & x_{N-1} \arrow[u] & x_{N-2} \arrow[u] \arrow[l] & \cdots \arrow[l] & x_1 \arrow[u] \arrow[l] & x_0 \arrow[u] \arrow[l]
\end{tikzcd}$$

### – choice of terminal cost $P$:

choose $P = P_\infty$ from infinite horizon LQR below! (=remaining cost to "best" solution)

### – stability:

$\underbrace{x(k+1) = Ax(k) + Bu(k)}_{\text{system model}}$ ; $\underbrace{u(k) = K(x(k),R,Q,N)\cdot x(k)}_{\text{LQR output}}$ → $x(k+1) = (A+BK)x$ is stable for what $R,Q,N$?
↳ hard to tell!

## • infinite horizon ($N=\infty$)

$J_\infty(x(k)) = x(k)^T P_\infty x(k)$  ($J_\infty$: optimal cost to $\infty$)

$u^*(k) = F_\infty \cdot x(k)$  ($u^*(k)$: optimal input)

from $P_{i+1} = P_i$ at $\infty$ if converging (Riccati eq.):

$P_\infty = A^TP_\infty A + Q - A^TP_\infty B(B^TP_\infty B+R)^{-1}B^TP_\infty A$ → solve for $P_\infty$

$F_\infty = -(B^TP_\infty B+R)^{-1}B^TP_\infty A$

### – stability:

$x(k+1) = (A+BF_\infty)\cdot x(k)$

can be shown to be **globally asympt. stable** with Lyapunov func $V(x)$:
$V(x) = J_\infty(x) = x^TP_\infty x$ → $V(x)>0\ \forall x\neq 0$ ; $\lim\limits_{\|x\|\to\infty} V(x) \to \infty$ ; $V(x(k+1)) - V(x(k)) < 0\ \forall x \neq 0$
**if $A,B$ are stabilizable and detectable!**

# Constrained Finite Time Optimal control

$$J^*(x_0) = \min_U \; l_f(x_N) + \sum_{i=0}^{N-1} l(x_i, u_i)$$

subj. to: $x_{i+1} = A \cdot x_i + B u_i$

$x_i \in \mathcal{X} \leftarrow \mathcal{X} = \{x \mid A_x x \le b_x\}$

$u_i \in \mathcal{U} \leftarrow \mathcal{U} = \{u \mid A_u u \le b_u\}$

$x_N \in \mathcal{X}_f \leftarrow \mathcal{X}_f = \{x \mid A_f x \le b_f\}$

$J^*$: optimal total cost

$l(x,u)$: stage cost (s.t: $l(0,0)=0$ and $l(x,u)>0$ for $x\neq 0, u \neq 0$)

$l_f(x)$: terminal cost (s.t: $l_f(0)=0$ and $l(x)>0$ for $x \neq 0$)

$\mathcal{X}, \mathcal{U}$: constraint spaces for $x, u$ $\quad \mathcal{X}_f$: terminal $x$ constraint space

$\rightarrow U^* = \{u_0^*, \dots, u_N^*\}$: optimal input $\quad u_0^*$: optimal control action

$\mathcal{X}_0$: **feasible set** (all $x_0$ for which the optim. prob. is feasible)

## • quadratic cost: ($l_f(x_N) = x_N^T P x_N$ ; $l(x_i,u_i) = x_i^T Q x_i + u_i^T R u_i$ ; $P \succ 0$ ; $Q \succ 0$ ; $R \succ 0$)

– with substitution: $J^*(x_0) = \min_U \begin{bmatrix} U \\ x_0 \end{bmatrix}^T \begin{bmatrix} H & F^T \\ F & Y \end{bmatrix} \begin{bmatrix} U \\ x_0 \end{bmatrix} = U^T H U + 2 x_0^T F U + x_0^T Y x_0$ subj. to: $GU \le w + E x_0$

$$U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} \quad G = \begin{bmatrix} A_u & 0 & \cdots & 0 \\ 0 & A_u & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_u \\ \hline 0 & 0 & \cdots & 0 \\ A_x B & 0 & \cdots & 0 \\ A_x A B & A_x B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_f A^{N-1} B & A_f A^{N-2} B & \cdots & A_f B \end{bmatrix} \quad w = \begin{bmatrix} b_u \\ b_u \\ \vdots \\ b_u \\ \hline b_x \\ b_x \\ b_x \\ \vdots \\ b_f \end{bmatrix} \quad E = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \hline -A_x \\ -A_x A \\ -A_x A^2 \\ \vdots \\ -A_f A^N \end{bmatrix}$$

$$\underbrace{H = [\;] \quad F = [\;] \quad Y = [\;]}_{\text{see LQR batch approach for } H, F, Y}$$

– without substitution: $J^*(x_0) = \min_z \begin{bmatrix} z \\ x_0 \end{bmatrix}^T \begin{bmatrix} \bar{H} & 0 \\ 0 & Q \end{bmatrix} \begin{bmatrix} z \\ x_0 \end{bmatrix} = z^T \bar{H} z + x_0^T Q x_0$ subj. to: $G_{in} z \le w_{in} + E_{in} x_0$ ; $G_{eq} z = E_{eq} x_0$

$$z = \begin{bmatrix} x_1 \\ \vdots \\ x_N \\ u_0 \\ \vdots \\ u_{N-1} \end{bmatrix} \quad G_{eq} = \left[\begin{array}{cccc|ccc} I & & & & -B & & \\ -A & I & & & & \ddots & \\ & -A & I & & & & \\ & & \ddots & \ddots & & & \\ & & -A & I & & & -B \end{array}\right] \quad E_{eq} = \begin{bmatrix} A \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad G_{in} = \left[\begin{array}{cccc|ccc} 0 & A_x & & & & & \\ & & \ddots & & & & \\ & & A_x & & & & \\ & & & A_f & & & \\ \hline & & & & A_u & & \\ & & & & & \ddots & \\ & & & & & & A_u \end{array}\right] \quad w_{in} = \begin{bmatrix} b_x \\ \vdots \\ b_x \\ b_u \\ \vdots \\ b_u \end{bmatrix} \quad E_{in} = \begin{bmatrix} -A_x \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \bar{H} = \left[\begin{array}{ccc|ccc} Q & & & & & \\ & \ddots & & & & \\ & & Q & & & \\ & & & P & & \\ \hline & & & & R & \\ & & & & & \ddots \\ & & & & & & R \end{array}\right]$$

## • ∞-norm cost: ($l_f(x_N) = \|P x_N\|_\infty$ ; $l(x_i,u_i) = \|Q x_i\|_\infty + \|R u_i\|_\infty$)

$J^*(x_0) = \min_z c^T z$ subj. to: $\bar{G} z \le \bar{w} + \bar{S} x_0$

$z = [\varepsilon_0^x, \dots, \varepsilon_N^x, \varepsilon_0^u, \dots, \varepsilon_{N-1}^u, u_0^T, \dots, u_{N-1}^T]^T$

$c^T = [1, \dots, 1, 1, \dots, 1, 0, \dots, 0]$

$\pm Q x_i \le [\varepsilon_i^x, \dots, \varepsilon_i^x]^T \quad (i = 0, \dots, N-1)$

$\pm P x_N \le [\varepsilon_N^x, \dots, \varepsilon_N^x]^T$

$\pm R u_i \le [\varepsilon_i^u, \dots, \varepsilon_i^u]^T \quad (i = 0, \dots, N-1)$

$GU \le w + E x_0$ (like for quad. cost)

substit. $X = S^x x_0 + S^u U$ and rewrite:

$\bar{G} z \le \bar{w} + \bar{S} x_0$

$\bar{G} = \begin{bmatrix} G_\varepsilon & G_u \\ 0 & G \end{bmatrix}, \; \bar{w} = \begin{bmatrix} w_\varepsilon \\ w \end{bmatrix}, \; \bar{S} = \begin{bmatrix} S_\varepsilon \\ E \end{bmatrix}$

$G_\varepsilon, G_u, w_\varepsilon, S_\varepsilon = \dots ?$

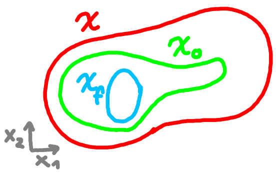
– example ∞-norm: $\min_{\underline{b}} \|\underline{A}\underline{b}\|_\infty \Leftrightarrow \min_{\varepsilon, \underline{b}} \varepsilon$ subj. to: $\pm \underline{A}\underline{b} \le [\varepsilon, \dots, \varepsilon]^T$

## • 1-norm cost: ($l_f(x_N) = \|P x_N\|_1$ ; $l(x_i,u_i) = \|Q x_i\|_1 + \|R u_i\|_1$)

formula derivation analog to ∞-norm...

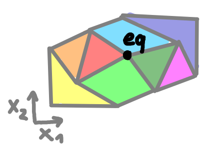
– example 1-norm: $\min_{\underline{b}} \|\underline{A}\underline{b}\|_1 \Leftrightarrow \min_{\underline{\varepsilon}, \underline{b}} (1, \dots, 1) \cdot \underline{\varepsilon}$ subj. to: $\pm \underline{A}\,\underline{b} \le \underline{\varepsilon}$

## • resulting control laws: ($u_0^* = \text{func}(x_0)$)



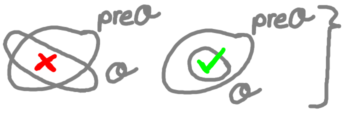

– quadratic program:
$u_0^*$ continuous and piecewise linear on polyhedra
(• = unconstrained LQR)


– linear program:
$u_0^*$ piecewise constant on polyhedra
(min. 1 constr. always active)



## Invariance (sets of states that are guaranteed to respect constraints)

- **positive invariant set**: $\mathcal{O} \subseteq \mathcal{X}$ for which: $x \in \mathcal{O} \Rightarrow g(x) \in \mathcal{O}$ $\left(x(k+1)=g(x(k)) \leftarrow \text{NO INPUT!}\right)$ (state stays inside $\mathcal{O}$ forever)




↳ same as: $\mathcal{O} \subseteq \text{pre}(\mathcal{O})$ (pre-set: $\text{pre}(S) = \{x \mid g(x) \in S\}$ set that becomes $S$ in 1 step)
(maximal positive invariant set: $\mathcal{O}_\infty$ = union of all $\mathcal{O}$)

**pre-set with linear sys. $x(k+1)=Ax(k)$:**
- polytopic $\mathcal{X} = \{x \mid Fx \leq f\}$:
  ↳ $\text{pre}(\mathcal{X}) = \{x \mid FAx \leq f\}$
- ellipsoid $\mathcal{X} = \{x \mid x^T P x \leq \alpha\}$:
  ↳ $\text{pre}(\mathcal{X}) = \{x \mid x^T A^T P A x \leq \alpha\}$
- range $\mathcal{X} = [a,b] \rightarrow \text{pre}(\mathcal{X}) = \frac{1}{A} \cdot [a,b]$

**algorithm for finding $\mathcal{O}_\infty$:**
```
Ω₀ ← X
while
    Ω_{i+1} ← pre(Ω_i) ∩ Ω_i
    if Ω_{i+1} = Ω_i then
        return O_∞ = Ω_i
    endif
endwhile
```

**algorithm for finding largest ellipsoid $\mathcal{O}_e$:**
(works if ∃ Lyapunov func. $V(x) = x^T P x$ and $\mathcal{X}$ polytopic set $\{x \mid Fx \leq f\}$)

$\max\limits_{\alpha} \alpha$ s.t. $\{x \mid x^T P x \leq \alpha\} \subset \{x \mid Fx \leq f\}$

↳ equiv. to: $\max\limits_{\alpha} \alpha$ s.t. $h(F_i) \leq f_i \quad \forall \text{row } i$

$\quad h(F_i) = \max\limits_{x} F_i^T x \text{ s.t. } x^T P x \leq \alpha$
$\quad = \ldots = \|P^{-1/2} F_i\| \cdot \sqrt{\alpha}$

↳ equiv. to: $\boxed{\alpha^* = \min f_i^2 / (F_i P^{-1} F_i^T) \ \forall i \quad ↳ \mathcal{O}_e = \{x \mid x^T P x \leq \alpha^*\}}$

size of $\mathcal{O}_e$ depends on choice of $P$!
(can do better by optimizing $\alpha$ and $P$)

- **control invariant set**: like positive invariant, but allow inputs.
  $\mathcal{C} \subseteq \mathcal{X}$ for which: $x \in \mathcal{C} \Rightarrow \exists u \in \mathcal{U}$ s.t. $g(x,u) \in \mathcal{C}$ HARD!!
  (given the correct input, the state stays inside $\mathcal{C}$ forever)
  ↳ same as: $\mathcal{C} \subseteq \text{pre}(\mathcal{C})$ (pre-set: $\text{pre}(S) = \{x \mid \exists u \in \mathcal{U} \text{ s.t. } g(x,u) \in S\}$)

- **control invariant set under const. control law**: $u(k)=Kx(k) \rightarrow x(k+1) = \underline{(A+BK)}\,x(k) = A_K x(k)$ ; $\mathcal{X}_K = \left\{x \,\middle|\, \begin{bmatrix} A_x \\ A_u K \end{bmatrix} x \leq \begin{bmatrix} b_x \\ b_u \end{bmatrix}\right\}$ $\begin{matrix} \leftarrow \mathcal{X} \\ \leftarrow \mathcal{U} \end{matrix}$
  ↳ $\mathcal{C}_K$ = e.g. $\mathcal{O}_\infty$ alg. with $A_K, \mathcal{X}_K$ ↳ $\mathcal{C}_K|_{K=F_\infty}$ = e.g. $\mathcal{O}_e$ alg. with $K=F_\infty \rightarrow P=P_\infty$ = Lyap. func for $(A+BK)$, $\mathcal{X}_K|_{K=F_\infty}$

## recursive feasibility and stability → pick $\mathcal{X}_f, l_f(x)$ such that: (for constrained finite time optimal control)

1) **recursive feasibility**: optimizer will always find solution that satisfies all constraints when applying $u_0^*$ indefinitely
(motivation: finite time horizon could lead state $x$ to a value where no $u \in \mathcal{U}$ can keep $x \in \mathcal{X}$ in the next timesteps)
↳ prove that: if $x(k)$ **feasible** $(\in \mathcal{X}_0)$ then $x(k+1)$ **feasible** $(\in \mathcal{X}_0)$

2) **recursive stability**: $\lim\limits_{k\to\infty} x(k) = 0$ = "equilibrium" when reapplying $u_0^*$ indefinitely (motivation: will get to eq. eventually)
↳ prove that: optimal cost function $J^*(x_0)$ is a Lyapunov function $\Leftrightarrow J^*(x(k+1)) < J^*(x(k))$

- $\mathcal{X}_f = \{0\}$, $l_f(x) = ?$: (force state to $0$ = "equilibrium" at end of horizon)
  - feasibility proof: if $\{u_0^*, \ldots, u_N^*\} \rightarrow \{x_0^*, \ldots, x_N^* = 0\}$ feasible, then $\{u_1^*, \ldots, u_N^*, 0\} \rightarrow \{x_1^*, \ldots, x_N^* = 0, x_N^* = 0\}$ feasible → ✓
  - stability proof: $J^*(x(k)) = J(\{u_0^*, \ldots, u_N^*\}, \{x_0^*, \ldots, x_N^* = 0\})$ ⟶ $J^*(x(k+1)) \leq \tilde{J}(x(k+1)) < J^*(x(k))$
    $\tilde{J}(x(k+1)) = J(\{u_1^*, \ldots, u_N^*, 0\}, \{x_1^*, \ldots, x_N^* = 0, x_N^* = 0\}) = J^*(x(k)) - l(x(k), u_0^*)$
    ↳ $\boxed{l_f(x) = \text{anything}}$

- $\mathcal{X}_f \subseteq \mathcal{C}$, $l_f(x) = ?$: (force state to be in some control invariant set at end of horizon)
  - feasibility proof: ~ (similar to $\mathcal{X}_f = \{0\}$ proof) ~ → ✓
  - stability proof: ~ (similar to $\mathcal{X}_f = \{0\}$ proof) ~ → guaranteed if: $\boxed{l_f(x_{i+1}) - l_f(x_i) \leq -l(x_i, u_0^*(x_i)) \quad \forall x_i \in \mathcal{X}_f}$

- **practical choices**: (for linear system, quadratic cost)
  1) choose desired constraints $x \in \mathcal{X}, u \in \mathcal{U}$
  2) design unconstrained LQR: $F_\infty = \ldots, P_\infty = \ldots$
  3) choose terminal weight: $P = P_\infty \rightarrow l_f(x) = x^T P x$
  4) choose terminal set: $\mathcal{X}_f : x \in \mathcal{X}_f \Rightarrow (A+BF_\infty)x \in \mathcal{X}_f \ \forall x$ and $F_\infty x \in \mathcal{U}$
  (control invar. under $u = F_\infty x$)

  $A_{F_\infty} = A + BF_\infty$ ; $\mathcal{X}_{F_\infty} = \left\{x \,\middle|\, \begin{bmatrix} A_x \\ A_u F_\infty \end{bmatrix} x \leq \begin{bmatrix} b_x \\ b_u \end{bmatrix}\right\}$
  option 1: $\mathcal{X}_f = \mathcal{O}_e$ largest positive invariant ellipsoid using $P_\infty ; \mathcal{X}_{F_\infty}$, since $x^T P_\infty x$ is Lyapunov func of closed-loop controller $x_{k+1} = (A+BF_\infty)x_k$
  option 2: $\mathcal{X}_f = \mathcal{O}_\infty$ max. pos. inv. set for $A_{F_\infty}, \mathcal{X}_{F_\infty}$ system

# Practical issues

- **reference tracking** (track some non-zero set-point) (only for linear systems!)

system model: $x(k+1) = Ax(k) + Bu(k)$ ; $x \in \mathcal{X} = \{x \mid G_x x \le h_x\}$ ; $u \in \mathcal{U} = \{u \mid G_u u \le h_u\}$

goal: $r \approx y = Cx$ ($r$: ref. val. of tracked states, $y$: actual value of tracked states, $C$: selection matrix)

target state: $x_s, u_s$ such that $r = Cx_s$ (match reference) and $x_s = Ax_s + Bu_s$ (steady state)

↳ if multiple solutions exist (pick cheapest one): $\min_{x_s u_s} u_s^T R u_s$ subj. to: $\begin{bmatrix} I-A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix}$ ; $x_s \in \mathcal{X}$ ; $u_s \in \mathcal{U}$

↳ if no solutions exist (pick closest one): $\min_{x_s u_s} (Cx_s - r)^T Q_y (Cx_s - r)$ subj. to: $x_s = Ax_s + Bu_s$ ; $x_s \in \mathcal{X}$ ; $u_s \in \mathcal{U}$

optimization problem: $\min_{U} (y_N - Cx_s)^T P_y (y_N - Cx_s) + \sum_{i=0}^{N-1} (y_i - Cx_i)^T Q_y (y_i - Cx_i) + (u_i - u_s)^T R (u_i - u_s)$

$\min_{\Delta U} \Delta x^T P \Delta x^T + \sum_{i=0}^{N-1} \Delta x^T Q \Delta x + \Delta u^T R \Delta u$

constraints: subj. to: $\Delta x_{i+1} = A\Delta x_i + B\Delta u_i$ ; $G_x \Delta x_i \le h_x - G_x x_s$ ; $G_u \Delta u_i \le h_u - G_u u_s$ ; $\Delta x_N \in \mathcal{X}_f^{new}$

[ $R$: input cost
$Q_y$: ref. state cost
$P_y$: final ref. st. c.
$\Delta x = x - x_s$
$\Delta u = u - u_s$
$P = C^T P_y C$
$Q = C^T Q_y C$ ]

recursive feasibility: $\mathcal{X}_f$ changes for $r \neq 0$ because constraints change! → recompute $\mathcal{X}_f^{new}$, or scale $\mathcal{X}_f^{new} = \alpha \mathcal{X}_f$ to fit inside new constr.

stability: remains guaranteed if stable for $r = 0$ (e.g.: $P = P_\infty$ cost to inf of LQR)

receipt: **select target $x_s, u_s$ → run optimizer for $\Delta x, \Delta u$ → recover $u_0^* = \Delta u_0^* + u_s$**

- **offset-free reference tracking** (like before, but can deal with offsets)

system model: $x(k+1) = Ax(k) + Bu(k) + B_d \cdot d$    $d$: disturbance (assumed const.)
$y(k) = Cx(k) + C_d \cdot d$    $B_d, C_d$: some matrices

[ conditions on $B_d, C_d$:
$\begin{bmatrix} x(k+1) \\ d(k+1) \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} x(k) \\ d(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k)$
$y(k) = \begin{bmatrix} C \\ C_d \end{bmatrix} \cdot \begin{bmatrix} x(k) \\ d(k) \end{bmatrix}$
needs to be stabilizable + detectable! ]

linear state estimator: $\begin{bmatrix} \hat{x}(k+1) \\ \hat{d}(k+1) \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} \hat{x}(k) \\ \hat{d}(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) + \begin{bmatrix} L_x \\ L_d \end{bmatrix} (-y(k) + C\hat{x}(k) + C_d \hat{d}(k))$

($\hat{x}, \hat{d}$: estimates of $x, d$ ; pick $L_x, L_d$ such that $\begin{bmatrix} x(k) - \hat{x}(k) \\ d(k) - \hat{d}(k) \end{bmatrix}$ is stable!)

new target: $x_s, u_s$ such that $x_s = Ax_s + Bu_s + B_d \hat{d}$ (steady state) and $H(Cx_s + C_d \hat{d}) = r$ (match reference)
→ rest is the same!

- **MPC without terminal set $\mathcal{X}_f$**: recursive stability + feasibility is still guaranteed for all $x_0$ that land in $x_N \in \mathcal{X}_f$ without that constraint being part of the optimization problem. (these $x_0 \neq \mathcal{X}_0$! set is hard to compute)

- **soft state constraints** (allow state constr. to be broken for a cost, so problem will never become unfeasible)

$\epsilon_i$: amount each state constr. is broken at time $i$. ($\epsilon_i > 0$)
$l_\epsilon(\epsilon_i)$: penalty cost for breaking a state constr.
↳ pick such that: if orig. problem had a feasible solution $U^*$, then the softened problem has the same solution $U^*$ and $\epsilon_i = 0 \; \forall i$

→ option 1: $l_\epsilon(\epsilon) = v \cdot \|\epsilon\|_1 + \epsilon^T S \epsilon$ and $v > \|\lambda^*\|_\infty$ ; $S \ge 0$
→ option 2: $l_\epsilon(\epsilon) = v \cdot \|\epsilon\|_\infty + \epsilon^T S \epsilon$ and $v > \|\lambda^*\|_1$ ; $S \ge 0$
→ option 3: pick something and dont care for $U^*_{HARD} = U^*_{SOFT}$

($\lambda^*$: Lagrange multiplier of "hard" problem)

$$\min_U x_N^T P x_N + l_\epsilon(\epsilon_N) + \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i + l_\epsilon(\epsilon_i)$$

subj. to: $x_{i+1} = A \cdot x_i + B u_i$
$x_i \in \mathcal{X} \leftarrow \mathcal{X} = \{x \mid A_x x \le b_x + \epsilon_i\}$
$u_i \in \mathcal{U} \leftarrow \mathcal{U} = \{u \mid A_u u \le b_u\}$
$x_N \in \mathcal{X}_f \leftarrow \mathcal{X}_f = \{x \mid A_f x \le b_f + \epsilon_N\}$
$\epsilon_i \ge 0 \; \forall i$

## Robust MPC

*uncertainty model:* $x(k+1) = Ax(k) + Bu(k) + w(k)$ (*$w \in W$: random noise*)

- **robust positive invariant set** $\mathcal{O}^W \subseteq \mathcal{X}$ *for which* $x \in \mathcal{O}^W \to Ax + w \in \mathcal{O}^W \ \forall w \in W$ (*state in $\mathcal{O}^W$ forever, despite noise*)
  - ↳ *same as* $\mathcal{O}^W \subseteq \text{pre}^W(\mathcal{O}^W)$ (*robust pre-set:* $\text{pre}^W(\Omega) = \text{pre}(\Omega \ominus W) = \{x \mid Ax + w \in \Omega \ \ \forall w \in W\}$)

  [ • *robust pre-set with polytopic* $\mathcal{X}$: $\text{pre}^W(\Omega) = \{x \mid FAx + Fw \leq f \ \ \forall w \in W\} = \{x \mid FAx \leq f - [h_W(F_1^T), .., h_W(F_m^T)]^T\}$ (*rows of F*)

  • *robust pre-set with range* $\mathcal{X}, W$: $\text{pre}^W(\Omega) = \{x \mid ax + w \in [\mathcal{X}_L, \mathcal{X}_h] \ \forall w \in [W_L, W_h]\} = 1/a \cdot ([\mathcal{X}_L, \mathcal{X}_h] \ominus [W_L, W_h])$ ]

**algorithm for** $\mathcal{O}_\infty^W$ *is the same as for non-robust* $\mathcal{O}_\infty$, *but with* $\text{pre} \to \text{pre}^W$ ; **algorithm for** $\mathcal{O}_e^W$ *doesn't exist..?*

- **robust control invariant set under const. control law:** $C_K^W$ *analog to non robust* $C_K$

- **robust "open-loop" MPC** (*find trajectories U that are rec. feasible for any noise w*) (*often unfeasible problem*)

$$\min_{U} \sum_{i=0}^{N-1} l(x_i, u_i) + l_f(x_N)$$

subj. to: $x_{i+1} = Ax_i + Bu_i$

$x_i \in \mathcal{X} \ominus (W \oplus AW \oplus .. \oplus A^{i-1}W)$

$u_i \in \mathcal{U}$

$x_N \in \mathcal{X}_f \ominus (W \oplus AW \oplus .. \oplus A^{N-1}W)$

*shrink allowable $\mathcal{X}$ of no-noise system at every step such that noisy system is guaranteed to be in original $\mathcal{X}$ at every step!*

(e.g.: $l = x_i^T Q x_i + u_i^T R u_i$ ; $l_f = x_N^T P_\infty x_N$ ; $\mathcal{X}_f = C_{F_\infty}^W$)

- **tube MPC** ("*closed loop*") (*include a simple stabilizer in optim. prob.*) (*less often unfeasible*)

(*real*) *noisy system*: $x_{i+1} = Ax_i + Bu_i + w_i$

(*fake*) *noiseless system*: $z_{i+1} = Az_i + Bv_i$ ($z$: *state* $\hat{=}\, x$; $v$: *input* $\hat{=}\, u$)

*pick stabilizer*: ($u$ *s.t.* $x$ *is pulled to* $z$) $u_i = v_i + K(x_i - z_i) = v_i + Ke_i$ ($K$: *some matrix*)

↳ *state error*: $e_i = x_i - z_i$ ⟶ *state error dynamics*: $e_{i+1} = x_{i+1} - z_{i+1} = \ldots = (A+BK)e_i + w_i$ ⟶

*minimum robust invariant state error* $\mathcal{E}$: ($e \in \mathcal{E}$ *always*)

$e_0 = 0 \to e_1 = w_1 \in W \to e_i = \ldots \in W \oplus (A+BK)W \oplus \ldots \oplus (A+BK)^{i-1}W$

↳ $e_\infty \in \mathcal{E} = W \oplus (A+BK)W \oplus \ldots \oplus (A+BK)^\infty W$

[ *algorithm for* $\mathcal{E}$: $\Omega_0 = \{0\}$

$\Omega_{i+1} = \Omega_i \oplus (A+BK)^i W$

*if* $\Omega_{i+1} = \Omega_i$ *return* $\mathcal{E} = \Omega_i$ ]

[ $\mathcal{E}$ *with range* $W = [a, b]$:

$\mathcal{E} = [a, b] \cdot 1/1-(A+BK)$ ]

→ *concept*: *run optim. problem with noiseless system, but*:

- *always leave enough buffer for* $\mathcal{X}$ *to fit worst state error* $\mathcal{E}$

  $x_i \in z_i \oplus \mathcal{E} \subseteq \mathcal{X} \to z_i \in \mathcal{X} \ominus \mathcal{E}$

- *always leave control authority to run stabilizer for error* $\mathcal{E}$

  $u_i \in v_i \oplus K\mathcal{E} \subseteq \mathcal{U} \to v_i \in \mathcal{U} \ominus K\mathcal{E}$

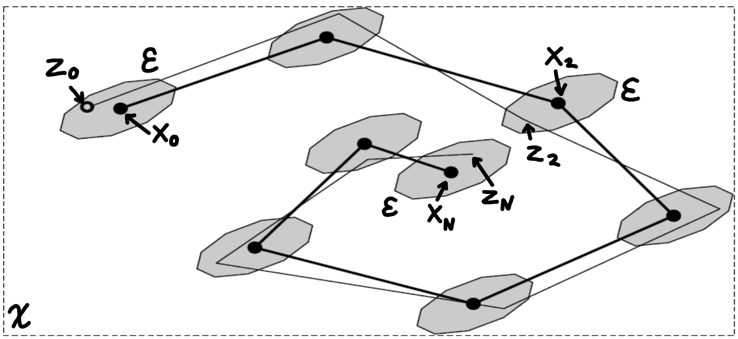
↳ $$\min_{V} \sum_{i=0}^{N-1} l(z_i, v_i) + l_f(z_N)$$

subj. to: $z_{i+1} = Az_i + Bv_i$ ; $z_i \in \mathcal{X} \ominus \mathcal{E}$ ; $z_N \in \mathcal{X}_f \ominus \mathcal{E}$ ; $v_i \in \mathcal{U} \ominus K\mathcal{E}$ ; $z_0 \in x_0 \oplus \mathcal{E}$

NOT $z_0 = x_0$

↳ $u_0^* = v_0^* + K(x_0 - z_0^*)$

(e.g. $l = z^T Q z + v^T R v$ ; $l_f = z^T F_\infty z$ ; $\mathcal{X}_f = C_{F_\infty}^W$ ; $K = F_\infty$)




- *recursive feasibility*: ✓ (*same argument as non robust MPC*)
- *recursive stability*: ✓ *state x will stay in $\mathcal{E}$ arround equilibrium for $t \to \infty$* (*same argument as non robust MPC*)

- **nominal MPC** (*use normal MPC and ignore noise even thoug it's there*)

- *recursive feasibilit*: ✗ *no guarantees on constraint satisfaction can be made!*
- *recursive stability*: ∼ *state can sometimes be shown to converge to some set* $\Omega$ (*not in this course*)

## Extensions

- **solution methods for optimization problem:** see convex optimization lecture
- **hybrid MPC:** add discrete dynamics (e.g. mode switches) to MPC problem
- **nonlinear MPC:** nonlinear systems, nonlinear constraints, non-convex optimization, etc.

---

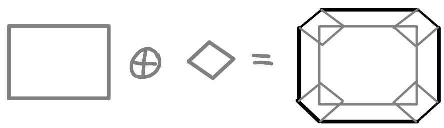
**MPC requirements** (ordered for importance)

1) satisfy constraints: $x(k) \subset \mathcal{X}$, $u(k) \subset \mathcal{U}$ $\forall k$
2) is stable: $\lim_{k\to\infty} x(k) = 0$
3) optimize "performance"
4) maximize set of possible $x(0)$

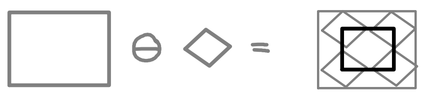
---

## Usefull set theory

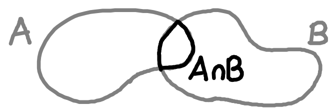
- **Minkowsky sum $\oplus$:** $A \oplus B = \{x+y \mid x \in A, y \in B\}$
  - range: $[a,b] \oplus [c,d] = [a+c, b+d]$



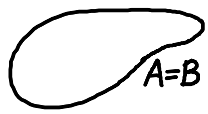
- **Pontryagin difference $\ominus$:** $A \ominus B = \{x \mid x+e \in A \ \forall e \in B\}$
  - range: $[a,b] \ominus [c,d] = [a-c, b-d]$



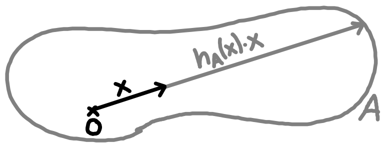
- **set intersection $\cap$:** $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$
  - halfspaces: $\{x \mid Ax \le a\} \cap \{x \mid Bx \le b\} = \{x \mid \begin{bmatrix} A \\ B \end{bmatrix} x \le \begin{bmatrix} a \\ b \end{bmatrix}\}$
  - ranges: $[a,b] \cap [c,d] = [\max(a,c), \min(b,d)]$




- **set equality $=$:** $x \in A$ iff $x \in B \rightarrow A = B$ (equivalent to $A \subseteq B$ AND $B \subseteq A$)
  - ranges: $[a,b] = [c,d]$ iff. $a=c$ and $c=d$




- **support function:** $h_A(x) = \sup\{x \cdot a \mid a \in A\} = \max_a x \cdot a$ subj. to: $a \in A$
  - ellipsoid: $A = \{x \mid x^T P x \le \alpha\} \rightarrow h_A(x) = \|P^{-0.5} x\|_2 \cdot \sqrt{\alpha}$
  - halfspace: $A = \{x \mid a^T x \le b\} \rightarrow h_A(x) = b / a^T x$
  - range: $A = [a,b] \rightarrow h_A(x) =$ if $x>0$: $x \cdot b$ else: $-x \cdot a$




## Usefull other math

- **vector derivatives:** $\nabla_x x^T A x = 2Ax$ ; $\nabla_x a^T x = \nabla_x x^T a = a$ ; $\nabla_x a = 0$
- **geometric series sum:** $\sum_{k=0}^{\infty} a x^k = a / 1-x$ $\left(\sum_{k=0}^{n} a x^k = a \cdot 1-x^n / 1-x\right)$
- **quadratic equation:** $ax^2 + bx + c = 0 \rightarrow x_{1,2} = -b \pm \sqrt{b^2 - 4ac} / 2a$