Model Predictive Control

FS25 ETHZ

Juri Pfammatter, Daniel Schweizer and Tobias Meier

0 Notation

x_i	Predicted state at step i
$x(i)$	Real state at step i
A^c	Continuous time matrix
A^k	Matrix A to the power of k
A	Discrete time matrix

1 System Theory

1.1 Models of Dynamic Systems

The basic formulation of a nonlinear, time-invariant, continuous-time, state space model can be stated as

$$\dot{x} = g(x,u)$$
$$y = h(x,u)$$

with

$x \in \mathbb{R}^n$	state vector
$u \in \mathbb{R}^m$	input vector
$y \in \mathbb{R}^p$	output vector
$g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$	system dynamics
$h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$	output function

1.2 Linearization

The first order **Taylor Expansion** around an operating point $\bar{x}$ of $f(x)$ is given by

$$f(x) \approx f(\bar{x}) + \left.\frac{\partial f}{\partial x^\top}\right|_{x=\bar{x}} (x - \bar{x})$$

with

$$\frac{\partial f}{\partial x^\top} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots & \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

The linearized, time-invariant, continuous-time state space model around the stationary operating point x_s, u_s can then be obtained with

$$\dot{x} = \overbrace{\left.\frac{\partial g}{\partial x^\top}\right|_{\substack{x_s\\u_s}}}^{A^c \in \mathbb{R}^{n\times n}} \Delta x + \overbrace{\left.\frac{\partial g}{\partial u^\top}\right|_{\substack{x_s\\u_s}}}^{B^c \in \mathbb{R}^{n\times m}} \Delta u$$
$$\Delta y = \underbrace{\left.\frac{\partial h}{\partial x^\top}\right|_{\substack{x_s\\u_s}}}_{C \in \mathbb{R}^{p\times n}} \Delta x + \underbrace{\left.\frac{\partial h}{\partial u^\top}\right|_{\substack{x_s\\u_s}}}_{D \in \mathbb{R}^{p\times m}} \Delta u$$

where

$$\dot{x}_s = g(x_s, u_s) = 0$$
$$y_s = h(x_s, u_s)$$

and

$$\Delta x = x - x_s$$
$$\Delta u = u - u_s$$
$$\Delta y = y - y_s$$
$$\Delta \dot{x} = \dot{x} - \underbrace{\dot{x}_s}_{=0}$$

Exact Solution for LTI CT SS Models

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

where

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

1.3 Discretization

DT systems are of the general form

$$x(k+1) = g(x(k), u(k))$$
$$y(k) = h(x(k), u(k))$$

or for the LTI case

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k) + Du(k)$$

Euler Discretization (Forward)

$$\dot{x}^c \approx \frac{x^c(t+T_s) - x^c(t)}{T_s}$$

with T_s describing the sampling time, hence

$$x(k) := x^c(t_0 + kT_s)$$
$$u(k) := u^c(t_0 + kT_s)$$

Then the DT model is given by

$$x(k+1) = x(k) + T_s g^c(x(k), u(k)) \qquad = g(x(k), u(k))$$
$$y(k) = h^c(x(k), u(k)) \qquad = h(x(k), u(k))$$

Therefore a LTI system becomes

$$x(k+1) = \overbrace{(\mathbb{I} + T_s A^c)}^{:=A} x(k) + \overbrace{T_s B^c}^{:=B} u(k)$$
$$y(k) = \underbrace{C^c}_{:=C} x(k) + \underbrace{D^c}_{:=D} u(k)$$

Exact Discretization of LTI

If the input u is held constant over a sampling interval (ZOH) one can retrieve an exact discretization of the LTI system

$$x(t_{k+1}) = \underbrace{e^{A^c T_s}}_{=A} x(t_k) + \underbrace{\int_0^{T_s} e^{A^c(T_s-\tau)}B^c d\tau}_{B}\, u(t_k)$$

with

$$B = (A^c)^{-1}(A - \mathbb{I})B^c$$

if A^c is invertible.

Solution of DT LTI System

If the input sequence $\{u(k), \ldots, u(k+N-1)\}$ and the initial state $x(k)$ is known the the solution for the discrete time system at time $k+N$ is given by

$$x(k+N) = A^N x(k) + \sum_{i=0}^{N-1} A^i Bu(k+N-1-i)$$

which is linear in the input sequence and initial condition.

1.4 LTI DT System Analysis

This section only summarizes the most important properties. Further details can be found in `https://github.com/MeierTobias/eth-control-systems-2`.

1.4.1 Coordinate Transformations

$$\tilde{x}(k) = Tx(k); \qquad \det(T) \neq 0$$

$$\tilde{x}(k+1) = \overbrace{TAT^{-1}}^{\tilde{A}}\tilde{x}(k) + \overbrace{TB}^{\tilde{B}}u(k)$$
$$y(k) = \underbrace{CT^{-1}}_{\tilde{C}}\tilde{x}(k) + \underbrace{D}_{\tilde{D}}u(k)$$

1.4.2 Stability

The DT LTI System

$$x(k+1) = Ax(k)$$

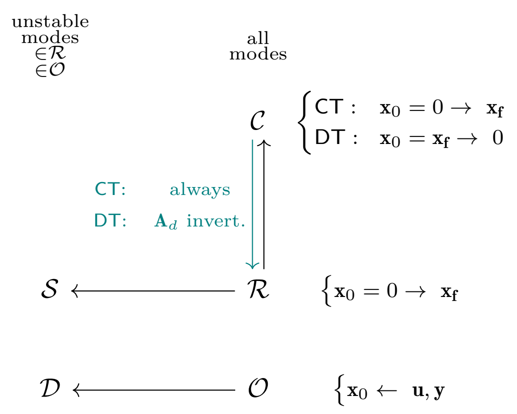
is globally asymptotically stable iff

$$|\lambda_j| < 1$$

for all eigenvalues λ_j of A.

1.4.3 Reachability and Observability

$$\mathcal{R} \subseteq \mathcal{C}, \quad \mathcal{R} \subset \mathcal{S}, \quad \mathcal{O} \subset \mathcal{D}$$


unstable modes $\in\mathcal{R}$ $\in\mathcal{O}$
all modes
$\mathcal{C}$
$\begin{cases} \text{CT}: & \mathbf{x_0} = 0 \to \mathbf{x_f} \\ \text{DT}: & \mathbf{x_0} = \mathbf{x_f} \to 0 \end{cases}$
CT: always
DT: $\mathbf{A}_d$ invert.
$\mathcal{S}$
$\mathcal{R}$
$\{\mathbf{x_0} = 0 \to \mathbf{x_f}$
$\mathcal{D}$
$\mathcal{O}$
$\{\mathbf{x_0} \leftarrow \mathbf{u}, \mathbf{y}$

1.4.4 Controllability

A system is controllable if for any pair of states $x(0), x^*$ there exists a finite time N and a control sequence u such that

$$x(N) = x^*$$
$$= A^N x(0) + \underbrace{\begin{bmatrix} B & AB & \cdots & A^{N-1}B \end{bmatrix}}_{=\mathcal{C} \text{ if } n=N} \begin{bmatrix} u(N-1) \\ u(N-2) \\ \vdots \\ u(0) \end{bmatrix}$$

The system has a unique solution in $u(0)\ldots u(N-1)$ iff the **controllability matrix**

$$\mathcal{C} := \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}$$

has **full row rank**.

Stabilizability iff all uncontrollable modes are stable

- A system is stabilizable if all unstable modes are controllable.
- Controllability always implies stabilizability

Stabilizability can be checked in the modal basis or via PBH test:

$$\text{rank}\left([\lambda_j I - A \mid B]\right) = n \quad \forall \lambda_j \in \Lambda_A^+ \Rightarrow (A,B) \text{ is stabilizable}$$

where Λ_A^+ is the set of unstable eigenvalues.

Reachability is used for CT systems.

1.4.5 Observability

A system is observable if there exists a finite N such that for every $x(0)$ the measurements $y(0), \ldots, y(N-1)$ uniquely distinguish the initial state $x(0)$. The system

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} = \underbrace{\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N-1} \end{bmatrix}}_{=\mathcal{O} \text{ if } n=N} x(0)$$

has a unique solution iff the **observability matrix**

$$\mathcal{O} := \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

has **full column rank**.

Detectability iff all unobservable modes are stable

- A system is detectable if all unstable modes are observable.
- Observability always implies detectability.

Detectability can be checked in modal basis or via

$$\text{if rank}\left(\left[A^\top - \lambda_j I \mid C^\top\right]\right) = n \quad \forall \lambda_j \in \Lambda_A^+ \Rightarrow (A,C) \text{ is detectable}$$

1.5 Nonlinear System Analysis

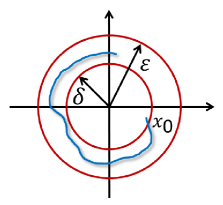
Lyapunov stability analysis is for instance useful if the linearization yields eigenvalues on the imaginary axis or the unit circle respectively. Lyapunov functions are often found via

1. definition of general form of the Lyapunov function.
2. search for parameter values via optimization.

1.5.1 Lyapunov Stability

The equilibrium point $\bar{x}$ of a DT system is **Lyapunov stable** iff for every $\epsilon > 0$ there exists a $\delta(\epsilon)$ such that

$$\|x(0) - \bar{x}\| < \delta(\epsilon) \to \|x(k) - \bar{x}\| < \epsilon, \ \forall k \geq 0$$



Global Asymptotic Stability

An equilibrium point $\bar{x}$ of a system is globally asymptotically stable if it is Lyapunov stable and attractive

$$\lim_{k\to\infty} \|x(k) - \bar{x}\| = 0, \ \forall x(0) \in \mathbb{R}$$

1.5.2 Global Lyapunov Function

Consider the equilibrium point $\bar{x} = 0$. A function $V : \mathbb{R}^n \to \mathbb{R}$, continuous at the origin, finite $\forall x \in \mathbb{R}^n$, and such that

$$\begin{aligned} &\|x\| \to \infty \Rightarrow V(x) \to \infty && \text{(radially unbounded)} \\ &V(0) = 0 \text{ and } V(x) > 0 && \forall x \in \mathbb{R}^n \setminus \{0\} \\ &V(g(x)) - V(x) \leq -\alpha(x) && \forall x \in \mathbb{R}^n \end{aligned}$$

where $\alpha : \mathbb{R}^n \to \mathbb{R}$ is continuous positive definite, is called a **Lyapunov function**, a system theoretic generalization of energy. For mechanical systems: total mechanical energy.

1.5.3 Local Lyapunov Function

If the conditions on asymptotical stability only hold on a closed and bounded, positively invariant set $\Omega \subset \mathbb{R}^n$ and V is finite $\forall x \in \Omega$, then the system is locally asymptotically stable in Ω, where Ω is also called the **region of attraction (ROA)**.

1.5.4 Global Lyapunov Stability

If a system admits a Lyapunov function $V(x)$, then $x = 0$ is **globally asymptotically stable**.

If $V(x)$ satisfies the conditions only with $\alpha(x)$ being positive **semi**definite, then $x = 0$ is **globally Lyapunov stable**.

1.6 Global Lyapunov Stability of LTI DT System

For the LTI DT system

$$x(k+1) = Ax(k)$$

One can take

$$V(x) = x^\top Px$$

as a candidate Lyapunov function with $P \succ 0$ (positive definite). It's easy to see that it satisfies

$$V(0) = 0 \text{ and } V(x) > 0$$
$$\|x\| \to \infty \Rightarrow V(x) \to \infty$$

To construct the $\alpha(x)$ function the "energy decrease" condition is used

$$\begin{aligned} V(Ax) - V(x) &= x^\top A^\top PAx - x^\top Px \\ &= x^\top\left(A^\top PA - P\right)x \leq -\alpha(x) \end{aligned}$$

If $\alpha(x)$ is chosen as

$$\alpha(x) = x^\top Qx, \quad Q > 0$$

the condition can be satisfied if a $P > 0$ can be found that solves the **discrete-time Lyapunov equation**

$$A^\top PA - P = -Q, \quad Q > 0$$

The discrete-time Lyapunov equation has a unique solution $P > 0$ iff A has all eigenvalues inside the unit circle, i.e. iff the system $x(k+1) = Ax(k)$ is stable.

Closed Loop Control

When using LQR, the matrix P that satisfies the DARE equation also satisfies the Lyapunov equation

$$(A+BK)^\top P(A+BK) - P = -Q, \quad Q \succ 0$$

Note that Q is not the stage cost matrix!

1.6.1 Example: Lyapunov Stability for Linear Systems

Given the LTI DT system

$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k)$$

with

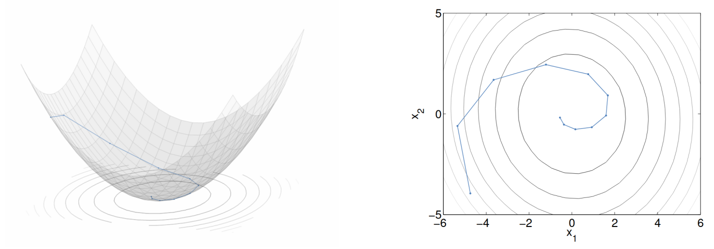
$$u(k) = Kx(k) = -\begin{bmatrix} 0.1160 & 0.5385 \end{bmatrix} x(k)$$

We choose $Q = \mathbb{I}$ and solve the discrete time Lyapunov equation

$$(A+BK)^\top P\,(A+BK) - P = -Q$$

$$P = \begin{bmatrix} 3.3921 & 2.7757 \\ 2.7757 & 7.7136 \end{bmatrix}$$

Since $V(x) = x^\top Px$ is a quadratic function, all level sets are ellipsoids with shape matrix P and the closed loop evolution can be simulated as shown.



1.6.2 Direct and Indirect Lyapunov Methods

1.6.2.1 Direct Lyapunov Method

The direct method involves finding a Lyapunov function $V(x)$ and proving that it satisfies the conditions for stability directly.

1.6.2.2 Indirect Lyapunov Method

The indirect method uses the linearization and Hartman-Grobman theorem to conclude stability of the nonlinear system from the stability of the linearized system in a local neighborhood of the equilibrium point.

2 Unconstrained Linear Quadratic Optimal Control

In this section, DT LTI systems with quadratic cost function are treated, where the goal is to regulate the state to the origin, without any state or input constraints.

2.1 Finite Horizon LQR

The **unconstrained finite horizon control problem** has the objective of finding a sequence of inputs

$$U := \begin{bmatrix} u_0^\top & u_1^\top & \dots & u_{N-1}^\top \end{bmatrix}^\top$$

that minimizes the objective function:

$$\begin{aligned} J^*(x(0)) &:= \min_U J(x_0, U) \\ &= \min_U \left(x_N^\top P x_N + \sum_{i=0}^{N-1} (x_i^\top Q x_i + u_i^\top R u_i) \right) \end{aligned}$$

$$\text{subj. to } x_{i+1} = Ax_i + Bu_i, \quad i = 0, \dots, N-1$$
$$x_0 = x(0)$$

In practice, the optimization variables include also the state $x_0, x_1, \dots, x_N$.

Weight Matrices

- $P \succeq 0$, with $P = P^\top$ is the terminal weight.
- $Q \succeq 0$, with $Q = Q^\top$ is the state weight.
- $R \succ 0$, with $R = R^\top$ is the input weight.
- N is the horizon length.

2.1.1 Batch Approach

The batch approach yields a FFW series U of numerical values for the input, as a function of $x(0)$. A constrained problem could theoretically still yield a solution through matrix inversion, which however might be unfeasible.

Problem Setup

Defining

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

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

and the block diagonal weight matrices

$$\overline{Q} := \text{blockdiag}(Q, \dots, Q, P), \; \overline{R} := \text{blockdiag}(R, \dots, R)$$

the finite horizon cost $J(x(0), U) = X^\top \bar{Q} X + U^\top \bar{R} U$ becomes the positive definite quadratic function in U

$$J(x(0), U) = U^\top HU + 2x(0)^\top FU + x(0)^\top {S^x}^\top \overline{Q} S^x x(0)$$

$$\text{where } H := (S^u)^\top \overline{Q} S^u + \overline{R} \succ 0 \text{ and } F := (S^x)^\top \overline{Q} S^u$$

Optimal Control Solution

Setting the gradient to zero yields the optimal control action

$$U^*(x(0)) = -H^{-1}F^\top x(0)$$

Note that

- This is linear in $x(0)$.
- H^{-1} always exists as $H \succ 0$ i.e. full rank.

The **optimal cost** is

$$J^*(x(0)) = x(0)^\top \left((S^x)^\top \overline{Q} S^x - FH^{-1}F^\top \right) x(0)$$

2.1.2 Recursive Approach

The recursive approach uses Dynamic Programming (DP) to compute an optimal policy, i.e. an optimal control action per state.

2.1.2.1 Bellman's Principle of Optimality

Bellman's Principle of Optimality justifies the recursive approach, stating that:

For any solution for steps 0 *to* N *to be optimal, any solution for steps* j *to* N *with* $j \geq 0$*, taken from the* 0 *to* N *solution, must itself be optimal for the* j*-to-*N *problem.*

Hence, for any $j = 0 \dots N$

$$J_j^\star(x_j) = \min_{u_j} J(x_j, u_j) + J_{j+1}^\star(x_{j+1})$$

$$\text{subj. to } x_{j+1} = Ax_j + Bu_j$$

2.1.2.2 Optimal Control Method

Problem Setup

We define the j-step optimal cost-to-go (*minimum* cost after step j) as the optimal cost attainable for the step j problem

$$J_j^\star(x(j)) := \min_{U_{j\to N}} x_N^\top P x_N + \sum_{i=j}^{N-1} (x_i^\top Q x_i + u_i^\top R u_i)$$

$$\text{subj. to } x_{i+1} = Ax_i + Bu_i, i = j, \dots, N-1$$
$$x_j = x(j)$$

Optimal Control Solution

Solving the local subproblem yields the optimal control input

$$u_i^\star = F_i x_i$$

$$F_i := -(B^\top P_{i+1} B + R)^{-1} B^\top P_{i+1} A$$

The optimal cost-to-go is

$$J_i^\star(x_i) = x(i)^\top P_i x(i)$$

with cost matrices found from the Discrete Time Riccati equation / **Riccati Difference equation (RDE)**:

$$P_i = A^\top P_{i+1} A + Q - A^\top P_{i+1} B (B^\top P_{i+1} B + R)^{-1} B^\top P_{i+1} A$$

$$P_N = P \text{ (init. with terminal weight)}$$

and the full-trajectory **optimal cost** $J^*(x(0))$ is

$$x(0)^\top P_0 x(0)$$

2.1.3 Comparison: Batch vs. Recursive Approach

Batch Approach

- open-loop control sequence
- + Allows for constraints
- - large matrix inversions

Recursive Approach

- + feedback policy $\rightarrow$ disturbance robustness

2.1.4 Stability of Finite Horizon LQR

Choice of Terminal Weight

There are three common choices for the terminal weight P in finite horizon LQR to ensure CL stability.

1. Choose P s.t. it matches the infinite horizon LQR solution

$$P = A^\top PA + Q - A^\top PB(B^\top PB + R)^{-1} B^\top PA$$

2. Choose P assuming no control action after the end of the finite horizon

$$x(k+1) = Ax(k), k = N, \dots, \infty$$

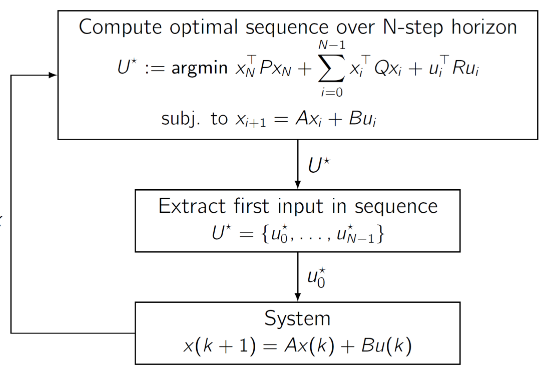
$$A^\top PA + Q = P \text{ (solve LE for } P)$$

This approach only makes sense if the system is asymptotically stable (or no positive definite solution P will exist).

3. Force both state and input both to be zero after the end of the finite horizon. No P is needed but the constraint

$$x_{i+N} = 0$$

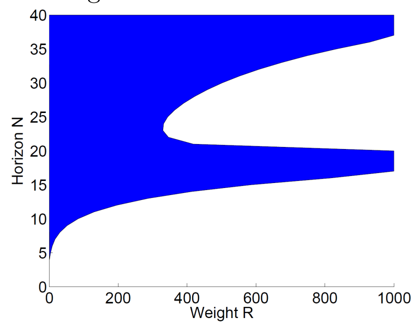
2.2 Receding Horizon LQR



Compute optimal sequence over N-step horizon
$U^\star := \text{argmin } x_N^\top P x_N + \sum_{i=0}^{N-1} x_i^\top Q x_i + u_i^\top R u_i$
subj. to $x_{i+1} = Ax_i + Bu_i$
$U^\star$
Extract first input in sequence
$U^\star = \{u_0^\star, \dots, u_{N-1}^\star\}$
$u_0^\star$
System
$x(k+1) = Ax(k) + Bu(k)$
x

Horizon Length

Stability depends on the choice of the weight matrices Q, R, P and the horizon length N.



Blue = stable, white = unstable

2.2.1 Stability of Receding Horizon LQR

Depending on the specific situation

- receding horizon *can* result in a stable CL trajectory with a horizon length that would yield an unstable trajectory in the OL case. However, receding horizon LQR does **not** guarantee a stable CL in general.
- OL control can be sufficient to achieve stability, i.e. receding horizon is not always needed in theory.

2.3 Infinite Horizon LQR

Problem Setup

$$J_\infty(x(0)) = \min_{u(\cdot)} \sum_{i=0}^{\infty} \left(x_i^\top Q x_i + u_i^\top R u_i \right)$$

$$\text{subj. to } x_{i+1} = Ax_i + Bu_i, \quad i = 0, 1, 2, \dots, \infty,$$
$$x_0 = x(0)$$

Optimal Control Solution

Solving the local subproblem yields the optimal control input

$$u^\star(k) = F_\infty x(k)$$

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

The optimal cost-to-go is

$$J_\infty(x(k)) = x(k)^\top P_\infty x(k)$$

where P_∞ is the solution of the **Discrete Time Algebraic Riccati equation (DARE)** ($P_i = P_{i+1} = P_\infty$)

$$P_\infty = A^\top P_\infty A + Q - A^\top P_\infty B (B^\top P_\infty B + R)^{-1} B^\top P_\infty A$$

Technical Conditions

Assuming

- (A, B) stabilizable
- $(Q^{\frac{1}{2}}, A)$ detectable,

then the RDE (initialized with Q at $i = \infty$ and solved for $i \searrow 0$) converges to the **unique positive definite** solution P_∞.

2.3.1 Stability of Infinite Horizon LQR

If the technical conditions are fulfilled, the optimal value function $J^\star(x) = x^T P_\infty x = V(x)$ is a Lyapunov function for the CL system $x^+ = (A + BF_\infty)x$, i.e. the system is asymptotically stable.

This can be shown by substituting the CL dynamics into the Lyapunov function $J^\star(x)$.

3 Convex Optimization

3.1 Problem Formulation and Terminology

$$\min_{x \in \text{dom}(f)} f(x)$$
$$\text{subj. to } \; g_i(x) \leq 0 \quad i = 1, \dots, m$$
$$h_i(x) = 0 \quad i = 1, \dots, p$$

where

$x = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix}^\top$	Decision/optimization variable
$f: \text{dom}(f) \to \mathbb{R}$	Objective function ($\text{dom}(f) \subseteq \mathbb{R}^n$)
$g_i: \mathbb{R}^n \to \mathbb{R}$	Inequality constraint functions
$h_i: \mathbb{R}^n \to \mathbb{R}$	Equality constraint functions
$\mathcal{X} := \{x \in \text{dom}(f) \| g_i(x) \leq 0, h_i(x) = 0\}$	Feasible set

Terminology

- **Feasible point:** point in the feasible set $\mathcal{X}$

- **Strictly feasible point:** point in the interior of $\mathcal{X}$ (i.e., $g_i(x) < 0$)
- **Optimal value:** $p^* = f(x^*) = \min_{x \in \mathcal{X}} f(x)$
- **Optimizer:** x^* that achieves p^* i.e. $f(x^*) \leq f(x)\ \forall x \in \mathcal{X}$. Not necessarily unique.
- **Unbounded below:** $p^* = -\infty$
- **Infeasible:** $\mathcal{X} = \emptyset$ i.e. $p^* = \infty$
- **Unconstrained:** $\mathcal{X} = \mathbb{R}^n$

3.1.1 Constraints

$g_i(x)$ is **active** at $\bar{x}$ if $g_i(\bar{x}) = 0$. Thus, equality constraints are always active.

A **redundant** constraint does not change the feasible set.

3.1.2 Optimality

$x \in \mathcal{X}$ is **locally optimal** if, for some $R > 0$

$$y \in \mathcal{X},\ \|x - y\| < R \quad \Rightarrow \quad f(x) \leq f(y)$$

$x \in \mathcal{X}$ is **globally optimal** if

$$f(x) \leq f(y) \quad \forall y \in \mathcal{X}$$

3.2 Convex Sets

A set $\mathcal{X}$ is convex **iff** for any pair of points x and y in $\mathcal{X}$:

$$\lambda x + (1-\lambda) y \in \mathcal{X}, \quad \forall \lambda \in [0,1],\ \forall x, y \in \mathcal{X}$$

Convex Combination
Any point x of the form

$$x = \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_k x_k \text{ with } \theta_1 + \ldots + \theta_k = 1, \theta_i \geq 0$$

is called convex combination of $x_1, \cdots, x_k$.

3.2.1 Hyperplanes and Halfspaces

A **hyperplane** is defined by $\{x \in \mathbb{R}^n \mid a^\top x = b\}$ for $a \neq 0$, where $a \in \mathbb{R}^n$ is the normal vector to the hyperplane.

A **halfspace** is everything on one side of a hyperplane $\{x \in \mathbb{R}^n \mid a^\top x \leq b\}$ for $a \neq 0$. It can either be open (strict inequality) or closed (non-strict inequality).

Hyperplanes are affine and convex, halfspaces are convex.

3.2.2 Polyhedra and Polytopes

Polyhedra and polytopes are always convex.

3.2.2.1 Polyhedra

A polyhedron is the intersection of a finite number of closed halfspaces:

$$P := \{x \mid a_i^\top x \leq b_i, i = 1, \ldots, m\} = \{x \mid Ax \leq b\}$$

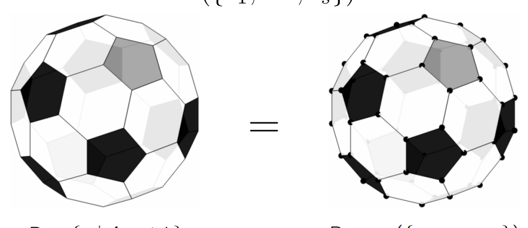
where

$$A := \begin{bmatrix} a_1 & a_2 & \ldots & a_m \end{bmatrix}^\top \quad \text{and} \quad b := \begin{bmatrix} b_1 & b_2 & \ldots & b_m \end{bmatrix}^\top$$

3.2.2.2 Polytopes

A polytope is a bounded polyhedron.

Minkowski-Weyl Theorem
For a set $P \subseteq \mathbb{R}^d$, the following are equivalent:
- P is a polytope $\{x \mid Ax \leq b\}$ (bounded)
- P is finitely generated, i.e., these exist a finite set of vectors $\{v_i\}$ such that $P = \mathrm{co}(\{v_1, \ldots, v_s\})$


$P = \{x \mid Ax \leq b\}$ $\qquad$ $P = \mathrm{co}(\{v_1, \ldots, v_s\})$

3.2.3 Ellipsoids

An ellipsoid is a set defined as

$$\{x \mid (x - x_c)^\top A^{-1} (x - x_c) \leq 1\}$$

where x_c is the centre of the ellipsoid, and $A \succ 0$ (i.e. A is symmetric and positive definite).

3.2.4 Norm Balls

The norm ball, defined by $\{x \mid \|x - x_c\| \leq r\}$ where x_c is the centre of the ball and $r \geq 0$ is the radius, is always convex for any norm.

Euclidian Ball
The Euclidean ball $B(x_c, r)$ is a special case of the ellipsoid, for which $A = r^2 \mathbb{I}$, so that $B(x_c, r) := \{x \mid \|x - x_c\|_2 \leq r\}$

3.2.5 Set Operations

The intersection of convex sets is convex, whereas the union of convex sets is not necessarily convex.

3.2.6 Convex Hull

For any subset S of $\mathbb{R}^d$, the *convex hull* $\mathrm{co}(S)$ is the intersection of all convex sets containing S, i.e. the smallest convex set containing S.

Vertex Representation
Given a set of points $S = \{v_1, v_2, \ldots, v_k\} \in \mathbb{R}^d$, their convex hull is

$$\mathrm{co}(S) = \left\{ x \,\middle|\, x = \sum_i \lambda_i v_i,\ \lambda_i \geq 0,\ \sum_i \lambda_i = 1,\ \forall i = 1, \ldots k \right\}$$

3.3 Convex Functions

A function $f : \mathrm{dom}(f) \to \mathbb{R}$ is convex **iff** $\mathrm{dom}(f)$ is convex and $\forall x, y \in \mathrm{dom}(f)$

$$f(\lambda x + (1-\lambda) y) \leq \lambda f(x) + (1-\lambda) f(y), \quad \forall \lambda \in (0,1)$$

In words, a line connecting any two points on the graph of the function lies above the graph (epigraph).

The function $f : \mathrm{dom}(f) \to \mathbb{R}$ is strictly convex if this inequality is strict.

The function f is **concave iff** $\mathrm{dom}(f)$ is convex and $-f$ is convex.

First Order Condition
A differentiable function $f : \mathrm{dom}(f) \to \mathbb{R}$ with a convex domain is convex **iff**

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x), \quad \forall x, y \in \mathrm{dom}(f)$$

i.e. the first-order approximation must be a global underestimator.

Second Order Condition
A twice-differentiable function $f : \mathrm{dom}\,(f) \to \mathbb{R}$ with convex domain $\mathrm{dom}\,(f)$ is convex **iff**

$$\nabla^2 f(x) \geq 0, \quad \forall x \in \mathrm{dom}\,(f), \quad \nabla^2 f(x)_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$$

f is called **strictly convex** if $\mathrm{dom}(f)$ is convex and the Hessian fulfills

$$\nabla^2 f(x) > 0, \quad \forall x \in \mathrm{dom}(f)$$

3.3.1 Level and Sublevel Sets

The **level set** L_α of a function f for value α is the set of all $x \in \mathrm{dom}(f)$ for which $f(x) = \alpha$:

$$L_\alpha := \{x \mid x \in \mathrm{dom}(f),\ f(x) = \alpha\}$$

For $f : \mathbb{R}^2 \to \mathbb{R}$ these are **contour lines** of constant height. The **sublevel set** C_α of a function f for value α is defined by

$$C_\alpha := \{x \mid x \in \mathrm{dom}(f),\ f(x) \leq \alpha\}$$

f is convex $\Rightarrow$ sublevel sets of f are convex $\forall \alpha$.
But, sublevel sets of f are convex $\forall \alpha \not\Rightarrow f$ is convex.

3.3.2 Examples of Convex Functions

Convex
- Affine functions: $ax + b,\ \forall a, b \in \mathbb{R}$
- Exponential functions: $e^{ax}\ \forall a \in \mathbb{R}$
- Powers: x^α on domain $\mathbb{R}_{++}$ for $\alpha \leq 0, \alpha \geq 1$
- Vector norms on $\mathbb{R}^n$:

$$\|x\|_p = \left(\sum_{i=1}^n |x|^p \right)^{1/p}, \text{ for } p \geq 1$$

$$\|x\|_\infty = \max_i |x_i|$$

Convexity Preserving Operations
- Nonnegative weighted sums: $\sum_{i=1}^n \theta_i f_i(x)$ for $\theta_i \geq 0$
- Composition with an affine function: $f(ax + b)$
- Pointwise maximum/supremum: $\max(f_1(x), f_2(x))$
- Partial minimization

Concave
- Affine functions: $ax + b$
- Powers: x^α on domain $\mathbb{R}_{++}$ for $0 \leq \alpha \leq 1$
- Logarithm: $\log(x)$ on domain $\mathbb{R}_{++}$
- Entropy: $-x \log(x)$ on domain $\mathbb{R}_{++}$

3.4 Convex Optimization Problems

$$\begin{aligned} \min_{x \in \mathrm{dom}(f)} \ & f(x) \\ \text{subj. to} \ & g_i(x) \leq 0 \quad i = 1, \ldots, m \\ & a_i^\top x = b_i \quad i = 1, \ldots, p \\ & (Ax = b \quad A \in \mathbb{R}^{p \times n}) \end{aligned}$$

$$\begin{cases} f, g_i & \text{convex functions} \\ \mathrm{dom}(f) & \text{convex set} \\ h_i(x) = a_i^\top x - b & \text{affine functions} \end{cases}$$

As a result, the feasible set $\mathcal{X}$ is convex.

3.4.1 Local and Global Optimality

For any convex optimization problem, any local optimal solution is globally optimal.

3.4.2 Equivalent Optimization Problems

Two problems are called equivalent if the solution to one can be (easily) inferred from the solution to the other, and vice versa.

Introducing Equality Constraints

$$\begin{aligned} & \min f(A_0 x + b_0) \\ & \text{s.t. } g_i(A_i x + b_i) \leq 0 \qquad i = 1, \ldots, m \end{aligned}$$

is equivalent to

$$\begin{aligned} & \min f(y_0) \\ & \text{s.t. } g_i(y_i) \leq 0, && i = 1, \ldots, m \\ & \quad A_i x + b_i = y_i, && i = 0, 1, \ldots, m \end{aligned}$$

Introducing Slack Variables

$$\begin{aligned} & \min f(x) \\ & \text{s.t. } A_i x \leq b_i, && i = 1, \ldots, m \end{aligned}$$

is equivalent to

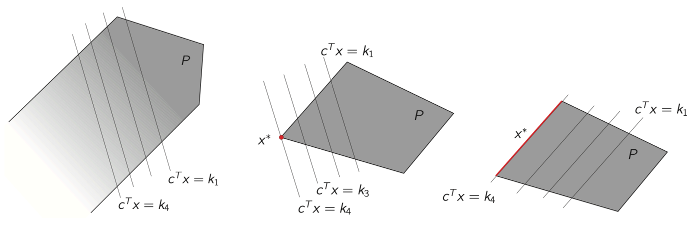
$$\begin{aligned} & \min f(x) \\ & \text{s.t. } A_i x + s_i = b_i, && i = 1, \ldots, m \\ & \quad s_i \geq 0, && i = 1, \ldots, m \end{aligned}$$

3.4.3 Linear Program (LP)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \ & c^\top x \\ \text{s.t. } & Gx \leq h \\ & Ax = b \end{aligned}$$

where the feasible set $\mathcal{P}$ is a polyhedron (convex).

For the set of optimizers $\mathcal{X}_{opt}$, generally, three cases can occur:
1. **Unbounded** $\mathcal{P}$ is unbounded below
2. **Bounded with unique optimizer**: $\mathcal{X}_{opt}$ is a **singleton**. *At least* n constraints are active.
3. **Bounded with multiple optimizers**: $\mathcal{X}_{opt}$ is a (bounded or unbounded) subset of $\mathbb{R}^n$. *At least* one constraint is active. In this case, the solution is sensitive to noise.



(a) Case 1 $\qquad$ (b) Case 2 $\qquad$ (c) Case 3

3.4.4 Quadratic Program (QP)

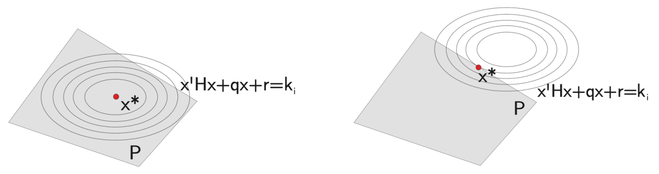
The general QP

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \ & \frac{1}{2} x^\top H x + q^\top x + r \\ \text{s.t. } & Gx \leq h \\ & Ax = b \end{aligned}$$

is convex if $H \succ 0$.

Problems with concave objective $H \prec 0$ are quadratic programs, but hard.

If feasible, in general two cases can occur:
1. **Case 1**: optimizer lies strictly inside the feasible polyhedron. No constraints active.
2. **Case 2**: optimizer lies on the boundary of the feasible polyhedron. At least one constraint active.



3.5 Optimality Conditions

3.5.1 Lagrange Dual Problem

The standard (possibly non-convex) optimization problem

$$\begin{aligned} \min_{x \in \mathrm{dom}(f)} \ & f(x) \\ \text{subject to } \ & g_i(x) \leq 0 \quad i = 1, \ldots, m \\ & h_i(x) = 0 \quad i = 1, \ldots p \end{aligned}$$

with (primal) decision variable x, domain $\mathrm{dom}(f)$ and optimal value p^* can be transformed into a dual problem using the **Lagrangian Function** $L : \mathrm{dom}\,(f) \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ is a weighted sum of the objective and constraint functions

$$L(x, \lambda, \nu) = f(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{i=1}^p \nu_i h_i(x)$$

with

$$\lambda_i \geq 0$$

The **dual function** d is then given by

$$d(\lambda, \nu) = \inf_{x \in \mathrm{dom}(f)} L(x, \lambda, \nu)$$

$L(x, \lambda, \nu)$ is affine. Hence, $d(\lambda, \nu)$ (the pointwise infimum in x) is always a concave function and is a lower bound for p^*

$$d(\lambda, \nu) \leq p^*, \quad \forall (\lambda \geq 0, \nu \in \mathbb{R}^p)$$

The **dual problem**

$$\begin{aligned} \max_{\lambda, \nu} \ & d(\lambda, \nu) \\ \text{subject to } \ & \lambda \geq 0 \end{aligned}$$

- is convex ($\min_{\lambda,\nu} -d(\lambda,\nu)$), even if the primal is not.
- has an optimal value $d^* \leq p^*$.
- the point (λ,ν) is **dual feasible** if $\lambda \geq 0$ and $(\lambda,\nu) \in \mathrm{dom}(d)$.
- Can often impose the constraint $(\lambda,\nu) \in \mathrm{dom}(d)$ explicitly.

3.5.1.1 Dual of a Linear Program (LP)

The primal problem is given by

$$\begin{aligned}\min_{x\in\mathbb{R}^n} \; & c^\top x\\ \text{subject to} \quad & Ax - b = 0\\ & Cx - e \leq 0\end{aligned}$$

With the **dual function**

$$\begin{aligned}d(\lambda,\nu) &= \min_{x\in\mathbb{R}^n}\left[c^\top x + \nu^\top(Ax-b) + \lambda^\top(Cx-e)\right]\\ &= \min_{x\in\mathbb{R}^n}\left[\left(A^\top\nu + C^\top\lambda + c\right)^\top x - b^\top\nu - e^\top\lambda\right]\\ &= \begin{cases}-b^\top\nu - e^\top\lambda & \text{if } A^\top\nu + C^\top\lambda + c = 0\\ -\infty & \text{otherwise}\end{cases}\end{aligned}$$

the **dual problem** is then given by

$$\begin{aligned}\max_{\lambda,\nu} \; & -b^\top\nu - e^\top\lambda\\ \text{subject to} \quad & A^\top\nu + C^\top\lambda + c = 0\\ & \lambda \geq 0\end{aligned}$$

this optimal value d^* is then a lower bound for the primal problem.

$$d^* \leq p^*$$

The dual of a linear program is also a linear program.

The dual of a mixed integer linear program (MILP) is a linear program.

3.5.1.2 Dual of a Quadratic Program (QP)

The primal problem of a QP is

$$\begin{aligned}\min_{x\in\mathbb{R}^n} \; & \frac{1}{2}x^\top Qx + c^\top x\\ \text{subject to} \quad & Cx - e \leq 0\\ & Q \succ 0\end{aligned}$$

the **dual function** is therefore

$$\begin{aligned}d(\lambda) &= \min_{x\in\mathbb{R}^n}\left[\frac{1}{2}x^\top Qx + c^\top x + \lambda^\top(Cx-e)\right]\\ &= \min_{x\in\mathbb{R}^n}\left[\frac{1}{2}x^\top Qx + (c + C^\top\lambda)^\top x - e^\top\lambda\right]\end{aligned}$$

for which the unconstrained minimization over x is convex for every λ. To get the optimal x the dual function is differentiated and set equal to zero (requires $Q > 0$)

$$\begin{aligned}Qx + c + C^\top\lambda = 0 \Leftrightarrow\\ x = -Q^{-1}(c + C^\top\lambda)\end{aligned}$$

This can then be substituted into the dual function

$$d(\lambda) = -\frac{1}{2}(c + C^\top\lambda)^\top Q^{-1}\left(c + C^\top\lambda\right) - e^\top\lambda$$

hence the **dual problem** is to maximize $d(\lambda)$ over $\lambda \geq 0$, or equivalently

$$\begin{aligned}\min_{\lambda} \; & \frac{1}{2}\lambda^\top CQ^{-1}C^\top\lambda + \left(CQ^{-1}c + e\right)^\top + \frac{1}{2}c^\top Q^{-1}c\\ \text{subject to} \quad & \lambda \geq 0\end{aligned}$$

The **dual problem** of a QP is another QP.

3.5.2 Weak and Strong Duality

Duality Gap:

$$p^* - d^*$$

Weak Duality
It is always true that

$$d^* \leq p^*.$$

Strong Duality
It is sometimes true that

$$p^* = d^*.$$

For non-convex problems this is usually not given.

Slater Condition
If for an optimization problem f and all g_i are convex, i.e. the problem is **convex**, then the Slater condition states:

If there exists at least one **strictly feasible point**, i.e.

$$\{x | Ax = b, \; g_i(x) < 0, \; \forall i \in \{1,\dots,m\}\} \neq \emptyset$$

this **always implies strong duality**.

3.5.3 Karush-Kuhn-Tucker (KKT) Conditions

Assume that f, all g_i and h_i are differentiable.

1. Primal Feasibility:

$$\begin{aligned}g_i(x^*) \leq 0 \quad & i = 1,\dots,m\\ h_i(x^*) = 0 \quad & i = 1,\dots,p\end{aligned}$$

2. Dual Feasibility:

$$\lambda^* \geq 0$$

3. Complementary Slackness:

$$\lambda_i^* g_i(x^*) = 0 \quad i = 1,\dots,m$$

implying

$$\begin{aligned}\lambda_i^* = 0 \text{ for every } g_i(x^*) < 0.\\ g_i(x^*) = 0 \text{ for every } \lambda_i^* > 0\end{aligned}$$

4. Stationarity:

$$\begin{aligned}\nabla_x L(x^*,\lambda^*,\nu^*) = \nabla f(x^*) + \sum_{i=1}^m \lambda_i^*\nabla g_i(x^*)\\ \dots + \sum_{i=1}^p \nu_i^*\nabla h_i(x^*) = 0\end{aligned}$$

Convex Problems
Iff (x^*,λ^*,ν^*) satisfy the KKT conditions, then they are optimal and Slater's condition holds, hence strong duality holds

$$p^* = d^*.$$

General Problems
For a general optimization problem there is only a necessary condition that states:
If x^* and (λ^*,ν^*) are primal and dual optimal solutions and strong duality holds, then x^* and (λ^*,ν^*) satisfy the KKT conditions.

3.5.3.1 Example: KKT Conditions for a QP

Given the following QP

$$\begin{aligned}\min_{x\in\mathbb{R}^n} \; & \frac{1}{2}x^\top Qx + c^\top x\\ \text{subject to} \quad & Ax = b\\ & x \geq 0\\ & Q \succeq 0\end{aligned}$$

The Lagrangian is

$$L(x,\lambda,\nu) = \frac{1}{2}x^\top Qx + c^\top x + \nu^\top(Ax-b) - \lambda^\top x$$

The KKT conditions are:

$$\begin{aligned}\nabla_x L(x,\lambda,\nu) = Qx + A^\top\nu - \lambda + c = 0 \quad & \text{KKT 4}\\ Ax = b \quad & \text{KKT 1}\\ x \geq 0 \quad & \text{KKT 1}\\ \lambda \geq 0 \quad & \text{KKT 2}\\ x_i\lambda_i = 0 \quad i = 1\dots n \quad & \text{KKT 3}\end{aligned}$$

3.5.4 Sensitivity Analysis

To analyze the sensitivity of a solution one can add a perturbation term to the constraints. The perturbed primal becomes

$$\begin{aligned}\min_x \; & f(x)\\ \text{subject to} \quad & g_i(x) \leq u_i\\ & h_i(x) = v_i\end{aligned}$$

and the corresponding dual

$$\begin{aligned}\max_{\lambda,\nu} \quad & d(\lambda,\nu) - u^\top\lambda - v^\top\nu\\ \text{subject to} \quad & \lambda \geq 0\end{aligned}$$

where u and v represent the perturbation.

In the case where strong duality holds for the unperturbed problem, the weak duality of the perturbed problem implies

$$\begin{aligned}p^*(u,v) &\geq d^*(\lambda^*,\nu^*) - u^\top\lambda^* - v^\top\nu^*\\ &= p^*(0,0) - u^\top\lambda^* - v^\top\nu^*\end{aligned}$$

Global Sensitivity Analysis
If a (large) Lagrangian multiplier and the corresponding perturbation have opposite sign, the problem is sensitive to the perturbation. In particular,

$$\begin{aligned}&\lambda_i^* \text{ large and } u_i < 0 && \Rightarrow \text{(i)}\\ &\lambda_i^* \text{ small and } u_i > 0 && \Rightarrow \text{(ii)}\\ &\left\{\begin{array}{l}\nu^* \text{ large and positive and } v_i < 0\\ \nu^* \text{ large and negative and } v_i > 0\end{array}\right\} && \Rightarrow \text{(i)}\\ &\left\{\begin{array}{l}\nu^* \text{ small and positive and } v_i > 0\\ \nu^* \text{ small and negative and } v_i < 0\end{array}\right\} && \Rightarrow \text{(ii)}\end{aligned}$$

(i) $p^*(u,v)$ increases greatly
(ii) $p^*(u,v)$ does not decrease much

Local Sensitivity Analysis
If in addition $p^*(u,v)$ is differentiable at $(0,0)$, then

$$\lambda_i^* = -\frac{\partial p^*(0,0)}{\partial u_i}, \qquad \nu_i^* = -\frac{\partial p^*(0,0)}{\partial v_i},$$

where
- λ_i^* is the sensitivity of p^* relative to i^{th} inequality.
- ν_i^* is the sensitivity of p^* relative to i^{th} equality.

4 Constrained Finite Time Optimal Control (CFTOC)

4.1 Constrained Linear Optimal Control

For a linear model the CFTOC problem is given by

$$\begin{aligned}J^*(x(k)) = \min_U \; & I_f(x_N) + \sum_{i=0}^{N-1} I(x_i,u_i)\\ \text{subject to} \quad & x_{i+1} = Ax_i + Bu_i, \quad i = 0,\dots,N-1\\ & x_i \in \mathcal{X}, u_i \in \mathcal{U}, \quad i = 0,\dots,N-1\\ & x_N \in \mathcal{X}_f\\ & x_0 = x(k)\end{aligned}$$

where $I(x_i,u_i)$ is the stage cost, $I_f(x_N)$ represents an approximation of the tail cost and $\mathcal{X}_f$ of the tail constraints, respectively. N is the time horizon and $\mathcal{X},\mathcal{U},\mathcal{X}_f$ are polyhedral regions.

Feasible Set
The feasible set describes a set of initial states for which the optimal control problem is feasible. It is defined only by the constraints and not by the cost:

$$\begin{aligned}\mathcal{X}_N = \{&x_0 \in \mathbb{R}^n | \exists(u_0,\dots,u_{N-1}) \text{ such that } x_i \in \mathcal{X}, u_i \in \mathcal{U}\\ & i = 0,\dots,N-1, x_N \in \mathcal{X}_f, \text{where } x_{i+1} = Ax_i + Bu_i\}\end{aligned}$$

Or in other words, the feasible set describes initial conditions from which it is possible to get into the final set $\mathcal{X}_f$ within $N-1$ steps.

In the convex case, the convex hull of a set of feasible solutions can be used as approximation of $\mathcal{X}_N$.

4.1.1 Quadratic Cost CFTOC

The cost function that incorporates a squared euclidean norm is composed of

$$\begin{aligned}I_f(x_N) &= x_N^T P x_N\\ I(x_i,U_i) &= x_i^\top Qx_i + u_i^\top Ru_i\end{aligned}$$

with $P \succeq 0, Q \succeq 0, R \succ 0$.

The **quadratic cost CFTOC problem**

$$\begin{aligned}J^*(x(k)) = \min_U \; & x_N^T P x_N + \sum_{i=0}^{N-1} x_i^\top Qx_i + u_i^\top Ru_i\\ \text{subject to} \quad & x_{i+1} = Ax_i + Bu_i, \quad i = 0,\dots,N-1\\ & x_i \in \mathcal{X}, u_i \in \mathcal{U}, \quad i = 0,\dots,N-1\\ & x_N \in \mathcal{X}_f\\ & x_0 = x(k)\end{aligned}$$

can be transformed into a QP problem

$$\begin{aligned}\min_{z\in\mathbb{R}^n} \; & \frac{1}{2}z^\top Hz + q^\top z + r\\ \text{subject to} \quad & Gz \leq h\\ & Az = b\end{aligned}$$

either with (4.1.1.2) or without (4.1.1.1) substitution of the future states X.

Notation
In the following we use
- n_x: dimension of state space
- n_u: dimension of input
- n_z: dimension of z for 4.1.1.1
- n_w: dimension of w for 4.1.1.2
- $n_{in,x}$: number of state inequality constraints
- $n_{in,u}$: number of input inequality constraints

4.1.1.1 Construction of QP Without Substitution

This approach keeps the state equations as equality constraints. The CFTOC problem is transformed into the QP problem parametrized in the initial condition $x(k)$

$$\begin{aligned}J^*(x(k)) = \min_z \; & \begin{bmatrix}z^\top & x(k)\end{bmatrix}\underbrace{\begin{bmatrix}\bar{H} & 0\\ 0 & Q\end{bmatrix}}_{\tilde{H}}\begin{bmatrix}z^\top & x(k)^\top\end{bmatrix}^\top\\ \text{subject to} \quad & G_{in}z \leq w_{in} + E_{in}x(k)\\ & G_{eq}z = E_{eq}x(k)\end{aligned}$$

where

$$z = \begin{bmatrix}x_1^\top & \cdots & x_N^\top & u_0^\top & \cdots & u_{N-1}^\top\end{bmatrix}^\top \in \mathbb{R}^{n_z = N(n_x+n_u)}$$

and $\tilde{H} \in \mathbb{R}^{(n_z+n_x)\times(n_z+n_x)} \geq 0$ as $J(x(k),U) \geq 0$ by assumption.

Cost
The cost submatrix $\bar{H}$ is

$$\bar{H} = \left[\begin{array}{cccc|ccc}Q & & & & & & \\ & \ddots & & & & & \\ & & Q & & & & \\ & & & P & & & \\ \hline & & & & R & & \\ & & & & & \ddots & \\ & & & & & & R\end{array}\right]$$

This can be constructed using the Matlab function

```
barH = blkdiag(kron(eye(N-1),Q),P,kron(eye(N),R))
```

Equality Constraints
The equalities are given from the system dynamics

$$x_{i+1} = Ax_i + Bu_i$$

can be rewritten as

$$G_{eq}z = E_{eq}x(k)$$

with

$$G_{eq} = \left[\begin{array}{cccc|cccc} I & & & & -B & & & \\ -A & I & & & & -B & & \\ & & \ddots & & & & \ddots & \\ & & -A & I & & & & -B \end{array}\right] \in \mathbb{R}^{N\cdot n_x \times n_z}$$

$$E_{eq} = \begin{bmatrix} A^\top & 0 & \cdots & 0 \end{bmatrix}^\top \in \mathbb{R}^{N\cdot n_x \times n_x}$$

Inequality Constraints
The inequality constraints

$$\mathcal{X} = \{x | A_x x \le b_x\}$$
$$\mathcal{U} = \{u | A_u u \le b_u\}$$
$$\mathcal{X}_f = \{x | A_f x \le b_f\}$$

can be rewritten as

$$G_{in}z \le w_{in} + E_{in}x(k)$$

with

$$G_{in} \in \mathbb{R}^{(n_{in,x}+n_{in,u})\times n_z}$$

$$= \left[\begin{array}{cccc|cccc} 0 & & & & 0 & & & \\ \hline A_x & & & & 0 & & & \\ & \ddots & & & & \ddots & & \\ & & A_x & & & & 0 & \\ & & & A_f & & & & 0 \\ \hline 0 & & & & A_u & & & \\ & \ddots & & & & A_u & & \\ & & 0 & & & & \ddots & \\ & & & 0 & & & & A_u \end{array}\right]$$

$$w_{in} \in \mathbb{R}^{(n_{in,x}+n_{in,u})\times 1}, \quad E_{in} \in \mathbb{R}^{(n_{in,x}+n_{in,u})\times n_x}$$

$$w_{in} = \left[\begin{array}{c} b_x \\ \hline b_x \\ \vdots \\ b_x \\ b_f \\ \hline b_u \\ b_u \\ \vdots \\ b_u \\ b_u \end{array}\right], \quad E_{in} = \left[\begin{array}{c} -A_x \\ \hline 0 \\ \vdots \\ 0 \\ 0 \\ \hline 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{array}\right]$$

4.1.1.2 Construction of QP With Substitution

The future states are only dependent on the current state and the applied input sequence. Hence they can be uniquely constructed from these and can therefore be substituted. To do so the same method as for the LQR batch approach (2.1.1) is used:

$$\underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}}_{X\in\mathbb{R}^{N\cdot n_x}} = \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}}_{S^x} x(k) + \underbrace{\begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \ddots & \ddots & 0 \\ A^{N-1}B & \cdots & AB & B \end{bmatrix}}_{S^u\in\mathbb{R}^{(N\cdot n_x)\times(N\cdot n_u)}} \underbrace{\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}}_{U}$$

$$X = S^x x(k) + S^u U$$

Cost
By substituting the X vector in the cost function of 4.1.1.1, one can rewrite the cost only depending on U and $x(k)$ (parametrized in the initial condition)

$$J^*(x(k),U) = \min_U \begin{bmatrix} U^\top & x(k)^\top \end{bmatrix} \underbrace{\begin{bmatrix} H & F^\top \\ F & Y \end{bmatrix}}_{\widetilde{H}} \begin{bmatrix} U^\top & x(k)^\top \end{bmatrix}^\top$$

$$\text{subject to} \quad GU \le w + Ex(k)$$

where $\widetilde{H} \in \mathbb{R}^{(N\cdot n_u+n_x)\times(N\cdot n_u+n_x)} \ge 0$ since $J(x(k),U) \ge 0$ by assumption.

Inequality Constraints
The inequality constraints

$$\mathcal{X} = \{x | A_x x \le b_x\}$$
$$\mathcal{U} = \{u | A_u u \le b_u\}$$
$$\mathcal{X}_f = \{x | A_f x \le b_f\}$$

can be rewritten as

$$GU \le w + Ex(k)$$

with

$$G \in \mathbb{R}^{(n_{in,u}+n_{in,x})\times(N\cdot n_u)}$$

$$= \left[\begin{array}{cccc} A_u & 0 & \cdots & 0 \\ 0 & A_u & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_u \\ \hline 0 & 0 & \cdots & 0 \\ A_xB & 0 & \cdots & 0 \\ A_xAB & A_xB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_fA^{N-1}B & A_fA^{N-2}B & \cdots & A_fB \end{array}\right]$$

$$w \in \mathbb{R}^{(n_{in,u}+n_{in,x})\times 1}, \quad E \in \mathbb{R}^{(n_{in,u}+n_{in,x})\times n_x}$$

$$w = \left[\begin{array}{c} b_u \\ b_u \\ \vdots \\ b_u \\ \hline b_x \\ b_x \\ b_x \\ \vdots \\ b_f \end{array}\right], \quad E = \left[\begin{array}{c} 0 \\ 0 \\ \vdots \\ 0 \\ \hline -A_x \\ -A_xA \\ -A_xA^2 \\ \vdots \\ -A_fA^N \end{array}\right]$$

Note that w is structurally different to w_{in} from 4.1.1.1.

4.1.1.3 Comparison: Substitution vs. No Substitution

	Substitution	No Substitution
# opt. vars.	Nn_u	$N(n_x + n_u)$
benefits	less opt. vars.	
	less constraints	sparse constr. ($\propto N$)
drawbacks	complicated constr.	more opt. vars.
	more constr. ($\propto N^2$)	

Note that substitution transforms input constraints into state constraints, making them more involved in general. Hence, no substitution is often preferable. For small N and large n_x however, substitution can be more efficient.

4.1.1.4 State Feedback Solution

As $n_x >= 1$, the CFTOC problem is a **multiparametric quadratic program (mp-QP)** in general with the following solution properties:

- u_0^* is of the form (nonlinear feedback policy)

  $$u_0^* = \kappa(x(k)), \quad \forall x(k) \in \mathcal{X}_0$$

  with $\kappa : \mathbb{R}^n \to \mathbb{R}^m$ cont., **piecewise affine** on polyhedra

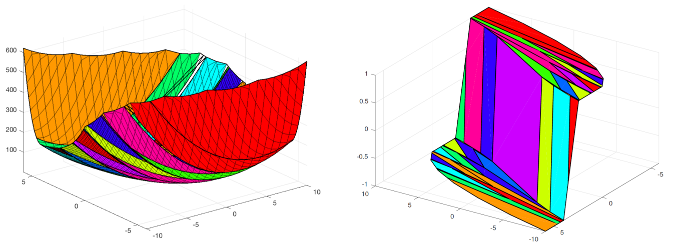
  $$\kappa = \kappa(x) = F^jx + g^j, \quad \text{if} \quad x \in CR^j, \quad j = 1, \cdots, N^r$$

- The polyhedral sets for the individual control laws

  $$CR^j = \{x \in \mathbb{R}^n | H^jx \le K^j\}, j = 1, \cdots, N^r$$

  are a partition of the feasible polyhedron $\mathcal{X}_0$.
- $J^*(x(k))$ is **convex**, **piecewise quadratic** on polyhedra.

Explicit MPC addresses how to compute this solution.



Value function   Optimal control input

4.1.2 1-Norm and ∞-Norm Cost CFTOC

The 1-norm and ∞-norm cost CFTOC can be transformed into LPs of the form

$$\min_{z\in\mathbb{R}^n} C^Tz$$
$$\text{subj. to } Gz \le h$$
$$Az = b$$

The key method is to reformulate the problem in **epigraph form** using auxiliary variables.

4.1.2.1 l_∞ Minimization

The constrained ℓ_∞ (Chebyshev) minimization problem is:

$$\min_{x\in\mathbb{R}^n} \|x\|_\infty$$
$$\text{subj. to } Fx \le g$$

The peak absolute value in x is the largest value of all $\{-x_i, x_i\}$. Hence, the cost can be rewritten as maximum of linear functions

$$\min_{x\in\mathbb{R}^n} [\max\{x_1, \ldots, x_n, -x_1, \ldots, -x_n\}]$$

Auxiliary Variable Formulation
In order to obtain a LP, one uses an equivalent formulation with *one* auxiliary variable t:

$$\min_{x,t} t$$
$$\text{subj. to } -\mathbf{1}t \le x \le \mathbf{1}t$$
$$Fx \le g$$

where $\mathbf{1}$ indicates a vector of ones and $-\mathbf{1}t \le x \le \mathbf{1}t$ bounds the absolute value of every element of x with a *common* scalar variable t.

Application to CFTOC
The ∞-norm minimization CFTOC reformulation introduces a *scalar* auxiliary variable for each state of the trajectory

$$z := \{\varepsilon_0^x, \ldots, \varepsilon_N^x, \varepsilon_0^u, \ldots, \varepsilon_{N-1}^u, u_0^\top, \ldots, u_{N-1}^\top\} \in \mathbb{R}^s,$$
$$s := (m+1)N + N + 1$$

This yields the ∞-norm minimization CFTOC **cost**

$$\min_z \varepsilon_0^x + \cdots + \varepsilon_N^x + \varepsilon_0^u + \cdots + \varepsilon_{N-1}^u$$

Using

$$x_i = A^ix_0 + \sum_{j=0}^{i-1} A^jBu_{i-1-j}$$

$$x_N = A^Nx_0 + \sum_{j=0}^{N-1} A^jBu_{N-1-j}$$

the state and input **norm constraints** amount to

$$\begin{aligned} \text{subj. to} \quad -\mathbf{1}_n\varepsilon_{x_i} &\le \pm Qx_i, \\ -\mathbf{1}_r\varepsilon_{x_N} &\le \pm Px_N, \\ -\mathbf{1}_m\varepsilon_{u_i} &\le \pm Ru_i \end{aligned}$$

The problem results in the following **standard LP**:

$$\min_z \ c^\top z$$
$$\text{subj. to } \bar{G}z \le \bar{w} + \bar{S}x(k),$$
$$\bar{G} = \begin{bmatrix} G_\varepsilon & G_u \\ 0 & G \end{bmatrix}, \bar{S} = \begin{bmatrix} S_\varepsilon \\ S \end{bmatrix}, \bar{w} = \begin{bmatrix} w_\varepsilon \\ w \end{bmatrix}$$

Using this formulation, given $x(k)$, the optimal control sequence U^* can be obtained via an LP solver.

General Formulation

$$y = \begin{bmatrix} x \\ z \end{bmatrix} \in \mathbb{R}^{(n+1)}$$

$$\min_x \|Ax\|_\infty \quad \Leftrightarrow \quad \min_y \begin{bmatrix} 0 & 1 \end{bmatrix} y$$
$$\text{s.t.} \begin{bmatrix} A & -\mathbf{1} \\ -A & -\mathbf{1} \end{bmatrix} y \le 0$$

4.1.2.2 l_1 Minimization

The constrained ℓ_1 minimization problem is defined as:

$$\min_{x\in\mathbb{R}^n} \|x\|_1$$
$$\text{subj. to } Fx \le g$$

again using the maximum of linear functions method yields

$$\min_{x\in\mathbb{R}^n} \left[\sum_{i=1}^m \max\{x_i, -x_i\}\right]$$

Auxiliary Variable Formulation The equivalent formulation introduces a *vector* of auxiliary variables $t \in \mathbb{R}^n$, yielding

$$\min_{x\in\mathbb{R}^n, t\in\mathbb{R}^n} \mathbf{1}^\top t,$$
$$\text{subj. to } -t \le x \le t,$$
$$Fx \le g$$

The constraint $-t \le x \le t$ bounds the absolute value of each component of x with each *component* of the vector variable t.

Application to CFTOC

$$z := \{(\boldsymbol{\varepsilon}_0^x)^\top, \ldots, (\boldsymbol{\varepsilon}_N^x)^\top, (\boldsymbol{\varepsilon}_0^u)^\top, \ldots, (\boldsymbol{\varepsilon}_{N-1}^u)^\top, u_0^\top, \ldots, u_{N-1}^\top\}$$
$z \in \mathbb{R}^s$ with $s := n(N+1) + 2mN$

$$\begin{aligned} \min_z \quad & \mathbf{1}^\top\boldsymbol{\varepsilon}_0^x + \cdots + \mathbf{1}^\top\boldsymbol{\varepsilon}_N^x + \mathbf{1}^\top\boldsymbol{\varepsilon}_0^u + \cdots + \mathbf{1}^\top\boldsymbol{\varepsilon}_{N-1}^u \\ \text{subj. to} \quad & -\boldsymbol{\varepsilon}_{x_i} \le Qx_i \le \boldsymbol{\varepsilon}_{x_i} \\ & -\boldsymbol{\varepsilon}_{x_N} \le Px_N \le \boldsymbol{\varepsilon}_{x_N} \\ & -\boldsymbol{\varepsilon}_{u_i} \le Ru_i \le \boldsymbol{\varepsilon}_{u_i} \end{aligned}$$

General Formulation

$$y = \begin{bmatrix} x \\ \mathbf{z} \end{bmatrix} \in \mathbb{R}^{(n+N)}$$

$$\min_x \|Ax\|_1 \quad \Leftrightarrow \quad \min_y \begin{bmatrix} 0 & \mathbf{1}^\top \end{bmatrix} y$$
$$\text{s.t.} \begin{bmatrix} A & -\mathbb{I} \\ -A & -\mathbb{I} \end{bmatrix} y \le 0$$

4.1.2.3 l_1, l_∞ State Feedback Solution

The CFTOC problem

$$\min_{z\in\mathbb{R}^n} c^Tz$$
$$\text{subj. to } \bar{G}z \le \bar{w} + \bar{S}x(k)$$

is again a mp-LP with the following solution properties:

- u_0^* has the form:

  $$u_0^* = \kappa(x(0)), \quad \forall x(0) \in \mathcal{X}_0,$$

where $\kappa : \mathbb{R}^n \to \mathbb{R}^m$ is cont. piecewise affine on polyhedra:

$$\kappa(x) = F^j x + g^j, \quad \text{if } x \in CR^j, \quad j = 1, \dots, N^r$$

- The polyhedral sets

$$CR^j = \{x \in \mathbb{R}^n \mid H^j x \le K^j\}, \quad j = 1, \dots, N^r$$

are a partition of the feasible polyhedron $\mathcal{X}_0$.
- In case of multiple optimizers, a **piecewise affine** control law exists.
- The value function $J^*(x(0))$ is **convex** and **piecewise affine** on polyhedra.

Location of Optimizer in State Space

The type of cost function influences the location of the optimizer in state space. See
- 3.4.3 for LP
- 3.4.4 for QP

4.2 Common Constraints

4.2.1 Polytopic Constraints

Input Constraints

$$u_{\min} \le u \le u_{\max} \quad \Leftrightarrow \quad \begin{bmatrix} -\mathbb{I} \\ \mathbb{I} \end{bmatrix} u \le \begin{bmatrix} -u_{\min} \\ u_{\max} \end{bmatrix}$$

Rate Constraints

$$\|x_K - x_{k+1}\|_\infty \le \alpha \quad \Leftrightarrow \quad \begin{bmatrix} \mathbb{I} & -\mathbb{I} \\ -\mathbb{I} & \mathbb{I} \end{bmatrix} \begin{bmatrix} x_k \\ x_{k+1} \end{bmatrix} \le \mathbf{1}\alpha$$

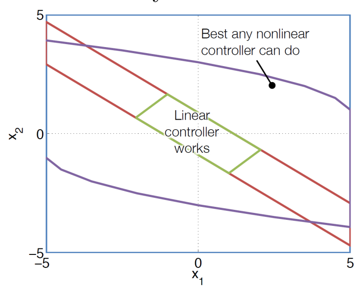
Magnitude Constraints

$$\|Cx_k\|_\infty \le \alpha \quad \Leftrightarrow \quad \begin{bmatrix} C \\ -C \end{bmatrix} x_k \le \mathbf{1}\alpha$$

5 Invariance

Limitations of Linear Controllers

The region in which a linear controller *never* violates state and input constraints is very limited.



Nonlinear control (MPC) can be used to increase the region, for which constraints can always be satisfied.

5.1 Invariance

The concept of invariance is used to access constraint satisfaction, for
- an autonomous system $x(k+1) = g(x(k))$
- or a CL system $x(k+1) = g(x(k), \kappa(x(k)))$, given a controller κ.

5.1.1 Invariant Sets

Positively Invariant Set

A set $\mathcal{O}$ is said to be a positively invariant set for an autonomous system if

$$x(k) \in \mathcal{O} \Rightarrow x(k+1) \in \mathcal{O}, \quad \forall k \in \{0, 1, \dots\}$$

Hence, if $\mathcal{O}$ is within the constraints, it provides a set of initial states from which the trajectory will **never** violate the system constraints.

Maximal Positively Invariant Set $\mathcal{O}_\infty$

The set $\mathcal{O}_\infty \subset \mathcal{X}$ is the maximal positively invariant set with respect to $\mathcal{X}$ if $\mathcal{O}_\infty$ is positively invariant and $\mathcal{O}_\infty$ contains all positively invariant sets.

$\mathcal{O}_\infty$ is the set of **all** states for which the system will remain feasible once in $\mathcal{O}_\infty$.

Geometric Condition for Invariance

A set $\mathcal{O}$ is a positively invariant set if and only if

$$\mathcal{O} \subseteq \operatorname{pre}(\mathcal{O})$$

Note that

$$\mathcal{O} \subseteq \operatorname{pre}(\mathcal{O}) \iff \operatorname{pre}(\mathcal{O}) \cap \mathcal{O} = \mathcal{O}$$

5.1.2 Pre-Sets

Given a set S and the dynamic system $x(k+1) = g(x(k))$, the pre-set of S is the set of states that evolve into the target set S in **one** time step:

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

Linear Autonomous Systems

Given $x(k+1) = Ax(k)$, the condition is

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

In the constraint case $S := \{x \mid Fx \le f\}$, this yields

$$\operatorname{pre}(S) = \{x \mid FAx \le f\}$$

5.1.3 Computing Invariant Sets

A conceptual algorithm to calculate invariant sets is

```
Input: $g$, $X$
Output: $\mathcal{O}_\infty$
$\Omega_0 \leftarrow X$
while true do
    $\Omega_{i+1} \leftarrow \operatorname{pre}(\Omega_i) \cap \Omega_i$
    if $\Omega_{i+1} = \Omega_i$ then
        return $\mathcal{O}_\infty = \Omega_i$
```

This algorithm generates the set sequence $\{\Omega_i\}$ satisfying $\Omega_{i+1} \subseteq \Omega_i$ for all $i \in \mathbb{N}$ and terminates when $\Omega_{i+1} = \Omega_i$, where Ω_i is the maximal positively invariant set $\mathcal{O}_\infty$ for $x(k+1) = g(x(k))$.

5.2 Control Invariance

5.2.1 Control Invariant Sets

A set $\mathcal{C} \subseteq \mathcal{X}$ is said to be a control invariant set if

$$x(k) \in \mathcal{C} \Rightarrow \exists u(k) \in \mathcal{U}$$
$$\text{such that } g(x(k), u(k)) \in \mathcal{C} \text{ for all } k \in \mathbb{N}^+$$

If this set is feasible (wrt. constraints) then we can be sure that the constraints will be satisfied for all time under the given control law.

Maximal Control Invariant Set $\mathcal{C}_\infty$

The set $\mathcal{C}_\infty$ is said to be the maximal control invariant set for the system $x(k+1) = g(x(k), u(k))$ subject to the constraints $(x, u) \in \mathcal{X} \times \mathcal{U}$ if it is control invariant and **contains all control invariant sets** contained in $\mathcal{X}$.

For all states contained in the maximal control invariant set $\mathcal{C}_\infty$, there exists a control law such that the system constraints are never violated.

5.2.2 Conceptual Calculation of Control Invariant Sets

The concept of a pre-set extends to systems with exogenous inputs:

$$\operatorname{pre}(S) := \{x \mid \exists u \in \mathcal{U} \text{ s.t. } g(x, u) \in S\}$$

A set $\mathcal{C}$ is a control invariant set if and only if $\mathcal{C} \subseteq \operatorname{pre}(\mathcal{C})$.

The same algorithm as for positively invariant sets can be used but the calculation of the pre-set is much more complicated.

Pre-set Computation for Constrained LTI System

Consider the system $x(k+1) = Ax(k) + Bu(k)$ under the constraints

$$u(k) \in \mathcal{U} := \{u \mid Gu \le g\}$$
$$S := \{x \mid Fx \le f\}$$

The preset can be computed as

$$\begin{aligned} \operatorname{pre}(S) &= \{x \mid \exists u \in \mathcal{U},\ Ax + Bu \in S\} \\ &= \{x \mid \exists u \in \mathcal{U},\ FAx + FBu \le f\} \\ &= \left\{ x \mid \exists u, \begin{bmatrix} FA & FB \\ 0 & G \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \le \begin{bmatrix} f \\ g \end{bmatrix} \right\} \end{aligned}$$

which is a **projection operation**.

Properties
- An entire set of states can map into each point
- The pre-set is a lot larger (positive), but much more difficult to compute
- The maximum control invariant set is the best any controller can do.

5.2.2.1 Usefulness

If one can compute a control invariant set, it provides a direct method for synthesizing a control law that will satisfy constraints. However, calculating the maximal control invariant set is often too complex in the case of linear systems and even more difficult for nonlinear ones. MPC implicitly describes a (suboptimal) control invariant set such that it's easy to represent and compute.

Linear System / Polyhedral Constraints

In this special case one can e.g. choose
- Polyhedral invariant set (can represent maximum control invariant set, resulting QP can be very complex due to many inequalities)
- Ellipsoidal invariant set (smaller than polyhedral, but easier to compute, resulting quadratically constrained QP has fixed complexity)

5.2.2.2 Control Laws from Control Invariant Sets

Let $\mathcal{C}$ be a control invariant set for the system $x(k+1) = g(x(k), u(k))$.

A control law $\kappa(x(k))$ guarantees that the system $x(k+1) = g(x(k), \kappa(x(k)))$ will satisfy the state constraints for all time if it respects the control invariant set, i.e.:

$$g(x, \kappa(x)) \in \mathcal{C} \quad \forall x \in \mathcal{C}$$

Therefore, we can synthesize a control law from a control invariant set by solving the following **optimization problem**:

$$\kappa(x) := \arg\min\{f(x, u) \mid g(x, u) \in \mathcal{C},\ u \in \mathcal{U}\},$$

where f is any function (including $f(x, u) = 0$).

Note: This ensures that the system will satisfy the constraints, but it does not guarantee convergence.

5.3 Computation of Simple Invariant Sets

This subsection specifies how to compute the intersection of two sets and the equality test needed in 5.1.3.

5.3.1 Polytopes

5.3.1.1 Intersection

Computing polytope intersections in vertex form is difficult. In inequality form however, one directly gets

$$S \cap T = \left\{ x \,\middle|\, \begin{bmatrix} C \\ D \end{bmatrix} x \le \begin{bmatrix} c \\ d \end{bmatrix} \right\}$$

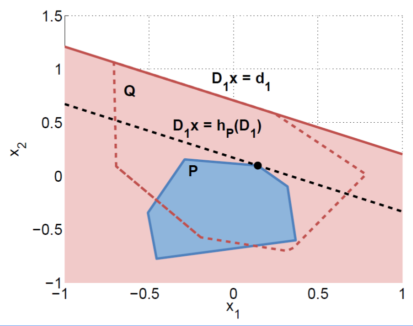
5.3.1.2 Equality Test (Subset Test)

$P = \{x \mid Cx \le c\}$ is a subset of $Q = \{x \mid Dx \le d\}$ if for each row, the **support function** is a subset of D:

$$h_P(D_i) \le d_i$$

where the support (extremum of P in direction D_i) is defined as

$$h_p(D_i) := \max_x D_i x$$
$$\text{s.t. } Cx \le c$$



5.3.2 Ellipsoids

Invariant ellipsoidal sets with its center in the origin, can directly be computed from Lyapunov functions.

If $V : \mathbb{R}^d \to \mathbb{R}$ is a Lyapunov function for the system $x(k+1) = g(x(k))$, then the sublevel set

$$Y_\alpha = \{x \mid V(x) \le \alpha\}$$

is an invariant set for all $\alpha \ge 0$.

5.3.2.1 Linear Systems

For linear systems $x(k+1) = Ax(k)$, the Lyapunov function $V(x) = x^\top Px$ (with $P \succ 0$ and $A^\top PA - P \prec 0$) yields the ellipsoidal invariant set

$$Y_\alpha = \{x \mid x^\top Px \le \alpha\} \subset \mathcal{X} = \{x \mid Fx \le f\}$$

Maxmium Invariant Set

If we want to find the largest such Y_α, we must solve

$$\max_\alpha \alpha$$
$$\text{s.t. } h_{Y_\alpha}(F_i) \le f_i$$
$$\forall i \in \{1, ..., n\}$$

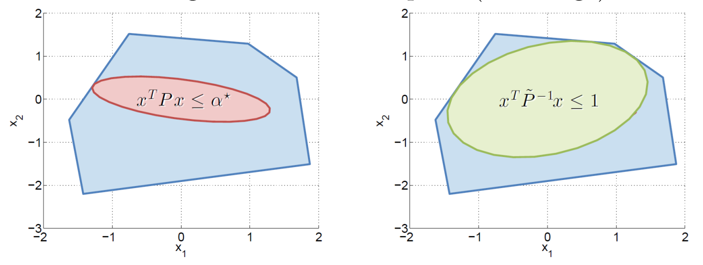
with the support of an ellipse

$$h_{Y_\alpha}(F_i) = \|P^{-1/2} F_i^T\| \sqrt{\alpha}$$

The largest ellipse Y_α in a polytope $\mathcal{X}$ can now be computed with a 1D optimization problem

$$\alpha^* = \min_{i \in \{1, \dots, n\}} \frac{f_i^2}{F_i P^{-1} F_i^\top}$$

Note that this only computes the largest ellipsoid with the center in the origin and a fixed shape P (left image).



6 Feasibility and Stability

Using LQR in constrained systems can e.g. lead to input saturation and instability. However, even MPC does not guarantee feasibility and stability per se. In particular, in finite horizon MPC
- Decrease in the prediction horizon can cause loss of stability properties.
- Depending on the initial condition, the CL trajectory may lead to states for which the optimization problem is infeasible after some steps, even without disturbance or model mismatch.

If we (could) solve the ∞-horizon control problem:
- problem is feasible $\to$ closed loop trajectories will be always feasible.
- the cost is finite $\to$ states and inputs will converge asymptotically to the origin.

i.e. OL and CL would be identical.

The goal of this section is hence to approximate the infinite horizon MPC to get feasibility and stability guarantees.

6.1 Zero Terminal Constraint

Given the terminal constraint $x_N = 0$.

6.1.1 Feasibility

Assume feasibility for $x(k)$ with optimal solution

$$\{u_0^*, u_1^*, \dots, u_{N-1}^*\}, \quad \{x(k), x_1^*, x_2^*, \dots, x_N^*\}$$

Applying the first control input u_0^* to the system, the state will evolve as

$$x(k+1) = Ax(k) + Bu(k) = Ax(k) + Bu_0^* = x_1^*$$

it follows directly that the sequence

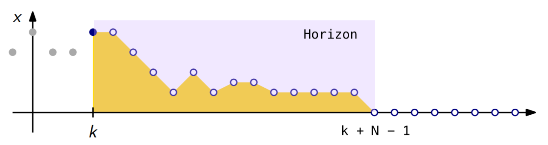
$$\widetilde{U} = \{u_1^*, \dots, u_{N-1}^*, 0\} \;\rightarrow\; \widetilde{X} = \{x_1^*, \dots, x_N^*, \underbrace{Ax_N^* + Bu_N}_{0}\}$$

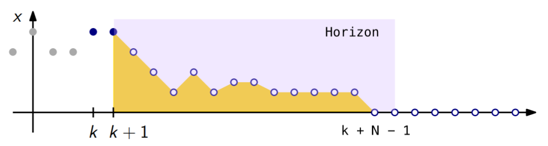
is feasible too. And therefore, recursive feasibility is guaranteed and the feasible set is **control invariant**.

6.1.2 Stability

Using the same (suboptimal) sequence $\widetilde{x}$ as above, the cost is given by

$$\begin{aligned} J^*(x(k+1)) &\le \widetilde{J}(x(k+1)) \\ &= \sum_{i=1}^{N-1} I(x_i^*, u_i^*) + I(x_N^*, 0) \\ &= \underbrace{\sum_{i=0}^{N-1} I(x_i^*, u_i^*)}_{J^*(x(k))} - \underbrace{I(x_0^*, u_0^*)}_{\ge 0} + \underbrace{I(x_N^*, 0)}_{0} \\ &\le J^*(x(k)) \end{aligned}$$





6.2 Terminal Subset Constraint

The terminal constraint $x_N = 0$ reduces the size of the feasible set. By using a convex set $\mathcal{X}_f$ as terminal constraints for which a local control law is known (can be constructed), that has a finite cost to stay in this set, one can increase the region of attraction and hence the feasible set.

6.2.1 Stability and Recursive Feasibility: Main Result

To guarantee stability and recursive feasibility the following three properties have to hold:

S1.1 Stage cost is positive definite, i.e. it is strictly positive and only zero at the origin.

S1.2 The terminal set $\mathcal{X}_f$ is **invariant** under the local control law $\kappa_f(x_i)$:

$$x_{i+1} = Ax_i + B\kappa_f(x_i) \in \mathcal{X}_f, \quad \forall\, x_i \in \mathcal{X}_f$$

All state and input **constraints are satisfied** in $\mathcal{X}_f$:

$$\mathcal{X}_f \subseteq \mathcal{X},\; \kappa_f(x_i) \in \mathcal{U}, \quad \forall\, x_i \in \mathcal{X}_f$$

S1.3 The terminal cost is a continuous **Lyapunov function** in the terminal set $\mathcal{X}_f$ and satisfies:

$$I_f(x_{i+1}) - I_f(x_i) \le -I(x_i, \kappa_f(x_i)), \quad \forall\, x_i \in \mathcal{X}_f$$

where 1. & 2. ensure recursive feasibility, 3. ensures stability.

Under those three assumptions:
The closed-loop system under the MPC control law $u_0^*(x)$ is asymptotically stable and the set $\mathcal{X}_N$ (region of attraction) is positive invariant (and equal to the feasible set) for the system

$$x(k+1) = Ax(k) + Bu_0^*(x(k)).$$

This also generalizes to **nonlinear** system dynamics

$$x(k+1) = g(x(k), u_0^*(x(k)))$$

but there the computation of the terminal set $\mathcal{X}_f$ and the function I_f can be very difficult.

6.2.2 Choice of Terminal Sets and Cost

1. Design a local control law κ_f around $x = 0$.
2. Determine the terminal cost to stay in that set under the local control law.
3. Compute the maximum invariant set for the closed loop system under the local control law hence $\mathcal{X}_f$.

6.2.2.1 Linear System, Quadratic Cost

For a linear system with quadratic cost

$$\begin{aligned} J^*(x(k)) = \min_U\; & x_N^\top P x_N + \sum_{i=0}^{N-1} x_i^\top Q x_i + u_i^\top R u_I \\ \text{s.t. }\; & x_{i+1} = Ax_i + Bu_i, \quad i = 0, \dots, N-1 \\ & x_i \in \mathcal{X}, u_k \in \mathcal{U}, \quad i = 0, \dots, N-1 \\ & x_N \in \mathcal{X}_f \\ & x_0 = x(k) \end{aligned}$$

one can design an unconstrained LQR controller as the local control law $\kappa_f(x_k) = F_\infty x_k$

$$F_\infty = -\left(B^\top P_\infty B + R\right)^{-1} B^\top P_\infty A$$

where the solution to the DARE P_∞ can be chosen as the terminal weight P.
Then compute the terminal set $\mathcal{X}_f$ to be the maximum invariant set for the closed-loop system $x_{k+1} = (A + BF_\infty)x_k$:

$$x_{k+1} = Ax_k + BF_\infty(x_k) \in \mathcal{X}_f, \quad \forall\, x_k \in \mathcal{X}_f$$

where all state and input constraints are satisfied in $\mathcal{X}_f$:

$$\mathcal{X}_f \subseteq \mathcal{X}, F_\infty x_k \in \mathcal{U}, \quad \forall\, x_k \in \mathcal{X}_f$$

This means, compute $\mathcal{X}_f$ for the CL system $(A + BF_\infty)$, with constraints

$$\mathcal{X}_{cl} := \left\{ x \,\middle|\, \begin{bmatrix} A_x \\ A_u F_\infty \end{bmatrix} x \le \begin{bmatrix} b_x \\ b_u \end{bmatrix} \right\}$$

This is the invariant terminal set used in the MPC.

Note that this setup fulfills the conditions from the stability and Recursive Feasibility Theorem 6.2.1.

6.2.2.2 Horizon Length Versus Feasible Set

For an MPC **without** terminal constraint, increasing N shrinks the feasible set, as more constraints are added. One has

$$\mathcal{X}_{N+j} \subseteq \mathcal{X}_N$$

For an MPC **with** terminal constraint, with larger N, the feasible set and region of attraction **grow**. The region of attraction ultimately approaches the maximum control invariant set. One has

$$\mathcal{X}_N \subseteq \mathcal{X}_{N+j}$$

Intuition: Given a trajectory with horizon length N, we can extend the trajectory arbitrarily using the control law $\kappa_l(x)$, while still obtaining feasible solutions.

Practical Note
In practice, one can enlarge N and check stability by sampling.

7 Practical MPC

7.1 Reference Tracking

A common task is to track a non-zero output set-point.
For a linear system with constraints, one can define the tracking task of the reference r as

$$z(k) = Hx(k) \to r \in \mathbb{R}^{n_r} \text{ as } k \to \infty.$$

where $z(k)$ are the outputs that have to follow r.

Note: The following concepts apply to piecewise constant references and not to general time-varying ones.

7.1.1 Steady-State Target Problem

The reference is achieved by the target state x_s if

$$z_s = Hx_s = r.$$

The target state should be a steady state, such that there exists an input that keeps the system at the target, i.e.

$$x_s = Ax_s + Bu_s$$

hence the target conditions become

$$\begin{array}{rcl} x_s & = & Ax_s + Bu_s \\ Hx_s & = & r \end{array} \;\Leftrightarrow\; \underbrace{\begin{bmatrix} I - A & -B \\ H & 0 \end{bmatrix}}_{(n_x+n_r)\times(n_x+n_u)} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix}$$

which can be easily solved for (x_s, u_s).

Multiple Solutions
It is possible that there exist multiple feasible (x_s, u_s) for a given reference r. Then the cheapest steady state can be used

$$\begin{aligned} \min\; & u_s^\top R_s u_s \\ \text{s.t. }\; & \begin{bmatrix} I - A & -B \\ H & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix} \\ & x_s \in \mathcal{X}, \quad u_s \in \mathcal{U}. \end{aligned}$$

No Solution
If no solution exists, then the reachable set point that is closest to r can be computed

$$\begin{aligned} \min\; & (Hx_s - r)^\top Q_s\, (Hx_s - r) \\ \text{s.t. }\; & x_s = Ax_s + Bu_s \\ & x_s \in \mathcal{X}, \quad u_s \in \mathcal{U}. \end{aligned}$$

Note that u_s is unique in this case.

7.1.2 Tracking Formulation

A commonly-used formulation for tracking MPC cost is

$$\min_U \|z_N - Hx_s\|_{P_z}^2 + \sum_{i=0}^{N-1} \|z_i - Hx_s\|_{Q_z}^2 + \|u_i - u_s\|_R^2$$

7.1.3 Delta-Formulation for Tracking

By introducing the delta formulation the tracking problem is transformed to the origin and therefore can be treated like a normal MPC problem.

The deviation variables are given by

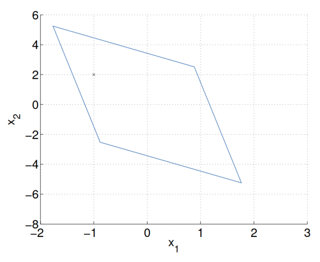
$$\begin{aligned} \Delta x &= x - x_s \\ \Delta u &= u - u_s \end{aligned}$$

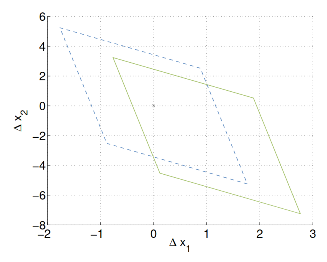
which results for a linear system in a system model

$$\begin{aligned} \Delta x_{k+1} &= x_{k+1} - x_s \\ &= Ax_k + Bu_k - (Ax_s + Bu_s) \\ &= A\Delta x_k + B\Delta u_k \end{aligned}$$

and the constraints become

$$\begin{aligned} G_x x \le h_x &\Rightarrow \quad G_x \Delta x \le h_x - G_x x_s \\ G_u u \le h_u &\Rightarrow \quad G_u \Delta u \le h_u - G_u u_s. \end{aligned}$$

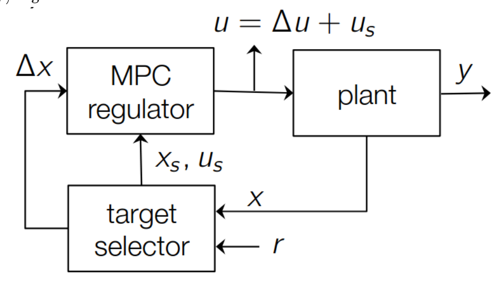




Resulting Optimization Problem
The complete problem in Delta-Formulation then looks like

$$\begin{aligned} \Delta U^* = \arg\min_{\Delta U}\; & \sum_{i=0}^{N-1} \Delta x_i^\top Q \Delta x_i + \Delta u_i^\top R \Delta u_i + V_f(\Delta x_N) \\ \text{s.t. }\; & \Delta x_0 = \Delta x(k) = x(k) - x_s \\ & \Delta x_{i+1} = A\Delta x_i + B\Delta u_i \\ & G_x \Delta x_i \le h_x - G_x x_s \\ & G_u \Delta u_i \le h_u - G_u u_s \\ & \Delta x_N \in \mathcal{X}_f \end{aligned}$$

where x_s, u_s is assumed to be determined beforehand by 7.1.1.



where the final input to the system is

$$u_0^* = \Delta u_0^* + u_s.$$

Note that if the target steady-state is uniquely defined by the reference, we can also include the target condition as a constraint in the MPC problem.

7.1.4 Convergence

Assume target is feasible with $x_s \in \mathcal{X}, u_s \in \mathcal{U}$ and choose terminal weight $V_f(x)$ and constraint $\mathcal{X}_f$ as in the regulation case satisfying (i.e. we found $V_f(x), \mathcal{X}_f$ only for the regulation case):

$$\begin{aligned} &\mathcal{X}_f \subseteq \mathcal{X}, \quad Kx \in \mathcal{U} && \forall x \in \mathcal{X}_f \\ &V_f\big(x(k+1)\big) - V_f\big(x(k)\big) \le -I\big(x(k), Kx(k)\big) && \forall x \in \mathcal{X}_f \end{aligned}$$

As the dynamics are identical in the Delta-formulation and the original system, the same Lyapunov function V_f can be used.

However, the terminal set constraint must be modified. In the Delta-formulation we require

$$\Delta x_N \in \mathcal{X}_f$$

which imposes

$$x_N \in \{x_s + \Delta x_N, \Delta x_N \in \mathcal{X}_f\} \subseteq \mathcal{X}$$

on the true system. Therefore the following condition is required: If in addition the target reference x_s, u_s is such that

$$x_s \oplus \mathcal{X}_f \subseteq \mathcal{X},\; K\Delta x + u_s \in \mathcal{U}, \quad \forall \Delta x \in \mathcal{X}_f$$

then the closed-loop system converges to the target reference, i.e.

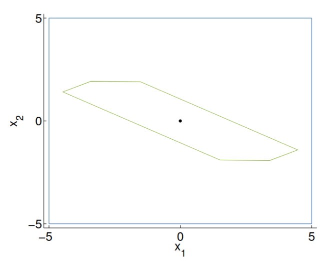
$$x(k) \to x_s \text{ and therefore } z(k) = Hx(k) \to r \text{ for } k \to \infty.$$

7.1.5 Terminal Set

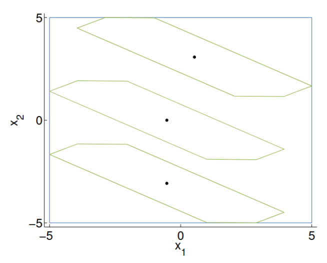
Due to the transformation the set of feasible targets may be significantly reduced.

In the following example the blue line represents the state constraints and the green line the terminal set.

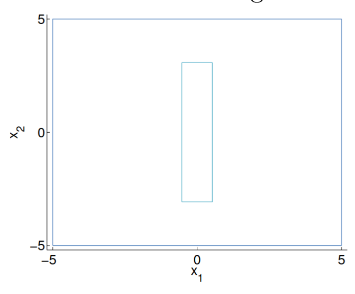
Regulation case:



Tracking using a shifted terminal set:



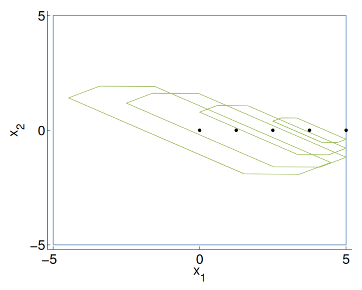
Set of feasible targets:



Terminal Set Scaling

To enlarge the set of feasible targets one can scale the terminal set.

$$\mathcal{X}_f^{\text{scaled}} = \alpha \mathcal{X}_f$$



The scaling factor α has to be chosen such that the state and input constraints are still satisfied. Targets at the boundary of the constraints hence $x_N = x_s$, correspond to a zero terminal set in the regulation case.

Note that scaling preserves invariance.

7.1.6 Offset-Free Reference Tracking

Goal: Track constant reference r, i.e. $z(k) = Hy(k) \to r$ for $k \to \infty$. If system is stabilized in the presence of the disturbance then it converges to set point with zero offset.

Algorithm

At each sampling time:

1. Estimate state and disturbance $\hat{x}$, $\hat{d}$
2. Solve steady-target problem using $\hat{d}$
3. Solve MPC problem:

$$\begin{aligned}\min_U \; & \sum_{i=0}^{N-1} (x_i - x_s)^T Q (x_i - x_s) \\ & + (u_i - u_s)^T R (u_i - u_s) + V_f(x_N - x_s) \\ \text{s.t.} \; & x_0 = \hat{x}(k), \quad d_0 = \hat{d}(k) \\ & x_{i+1} = Ax_i + Bu_i + B_d d_i, \quad i = 0, \dots, N \\ & d_{i+1} = d_i, \quad i = 0, \dots, N \\ & x_i \in \mathcal{X}, \quad u_i \in \mathcal{U}, \quad x_N - x_s \in \mathcal{X}_f\end{aligned}$$

7.1.6.1 Augmented Model

The constant disturbance can be included in to the dynamic system model as follows:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + B_d d_k \\ d_{k+1} &= d_k \\ y_k &= Cx_k + C_d d_k \end{aligned}$$

with $d \in \mathbb{R}^{n_d}$.

Observability of Augmented System

The only restriction on the choice of B_d, C_d is the observability of the augmented model.

The augmented system is observable if and only if (A, C) is observable and

$$\begin{bmatrix} A - I & B_d \\ C & C_d \end{bmatrix}$$

has full column rank, i.e. rank $= n_x + n_d$.

Note that

- The number of measured outputs must be large enough: $n_d \le n_y$.
- This is known as Hautus observability condition (or PBH test).

Intuitively, the condition can be explained as follows: At steady-state

$$\begin{bmatrix} A - I & B_d \\ C & C_d \end{bmatrix} \begin{bmatrix} x_s \\ d_s \end{bmatrix} = \begin{bmatrix} 0 \\ y_s \end{bmatrix}$$

and given y_s, d_s must be uniquely defined.

7.1.6.2 Linear State Estimation

To improve the estimate of the disturbance and to account for non-measurable states, an observer can be used. For the augmented model we get

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

where the last summand corrects the state **and** disturbance estimate. On convergence, this term vanishes, revealing the steady-state equations.

The **error dynamics** are:

$$\begin{aligned}\begin{bmatrix} \eta(k+1) \\ \eta_d(k+1) \end{bmatrix} &= \begin{bmatrix} x(k+1) - \hat{x}(k+1) \\ d(k+1) - \hat{d}(k+1) \end{bmatrix} \\ &= \left(\begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} + \begin{bmatrix} L_x \\ L_d \end{bmatrix} \begin{bmatrix} C & C_d \end{bmatrix} \right) \begin{bmatrix} x(k) - \hat{x}(k) \\ d(k) - \hat{d}(k) \end{bmatrix}\end{aligned}$$

To obtain zero estimation error, the estimator gain $L = \begin{bmatrix} L_x & L_d \end{bmatrix}^\top$ must be chosen such that the error dynamics are asymptotically stable.

Lemma: Offset-Free Steady-State Estimation

Suppose the observer is asymptotically stable and $n_y = n_d$. The observer steady state satisfies:

$$\begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_\infty \\ u_\infty \end{bmatrix} = \begin{bmatrix} -B_d \hat{d}_\infty \\ y_\infty - C_d \hat{d}_\infty \end{bmatrix}$$

In words: The observer output $C\hat{x}_\infty + C_d \hat{d}_\infty$ tracks y_∞ without offset.

7.1.6.3 Steady-State Selection

The knowledge of the disturbance estimate allows us to reformulate the steady-state selection problem as

$$\begin{aligned} x_s &= Ax_s + Bu_s + B_d \hat{d}_\infty \\ z_s &= H(Cx_s + C_d \hat{d}_\infty) = r \end{aligned}$$

which shows that both, the steady-state and the target are modified to account for the disturbance.

As we don't have access to $\hat{d}_\infty$, we use the best possible estimate for $\hat{d}_\infty$, namely $\hat{d}$ for the target selection problem, yielding

$$\begin{bmatrix} A - I & B \\ HC & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} -B_d \hat{d} \\ r - HC_d \hat{d} \end{bmatrix}$$

For regulation, we can simply set $r = 0$.

7.1.6.4 Offset-free Tracking: Main Result

Let $\kappa(\hat{x}(k), \hat{d}(k), r) = u_0^\star$ be the estimation-based control law. Assume

- $n_d = n_y$
- the RHC is recursively feasible and unconstrained for $k \ge j$
- and the closed-loop system

$$\begin{aligned} x(k+1) =& Ax(k) + B\kappa\big(\hat{x}(k), \hat{d}(k), r\big) + B_d d \\ \hat{x}(k+1) =& (A + L_x C)\hat{x}(k) + (B_d + L_x C_d)\hat{d}(k) \\ & + B\kappa\big(\hat{x}(k), \hat{d}(k), r\big) - L_x y(k) \\ \hat{d}(k+1) =& L_d C \hat{x}(k) + (I + L_d C_d)\hat{d}(k) - L_d y(k) \end{aligned}$$

  converges $(\hat{x}(k) \to \hat{x}_\infty,\ \hat{d}(k) \to \hat{d}_\infty,\ y(k) \to y_\infty$ for $k \to \infty)$.

Then $z(k) = Hy(k) \to r$ as $k \to \infty$, i.e. we track the reference in presence of the disturbance.

7.2 Enlarging the Feasible Set

7.2.1 MPC Without Terminal Set

Using a terminal set constraint is attractive due to stability and recursive feasibility guarantees. However,

- it reduces the feasible set
- potentially adds large number of extra constraints
- adds state constraints to problems with only input constraints

Hence, it would be beneficial to formulate an MPC without terminal constraint but yet with guaranteed stability.

Conditions

We can remove the terminal constraint while maintaining stability if

- initial state lies in sufficiently small subset of feasible set
- N is sufficiently large

such that the terminal state satisfies terminal constraint without enforcing it in the optimization.

In that case, the solution of the finite horizon MPC problem corresponds to the infinite horizon solution.

Properties

+ Controller defined in a larger feasible set
- Characterization of ROA or specification of required N extremely difficult

Method

Enlarge horizon and check stability by **sampling**.

With larger horizon length N, region of attraction approaches maximum control invariant set.

7.2.2 Soft Constrained MPC

In practice, input constraints are often "hard" whereas state constraints can usually be relaxed.

If we decide to tolerate state constraint violations, we can either minimize their

- duration
- or size

These goals can be conflicting and their relative importance depends on the application. The optimum lies on a Bambachian pareto curve. Operating exactly on this curve is usually difficult and only approximated.

Properties

- Standard methods do **not provide a stability guarantee** for OL unstable systems

7.2.2.1 Problem Setup

In soft constrained MPC we

- relax state constraints using slack variables $\epsilon_i \in \mathbb{R}^p$
- penalize constraint violation in cost using $l_\epsilon(\epsilon_i)$

The MPC can then be formulated as

$$\begin{aligned}\min_u \; & \sum_{i=0}^{N-1} x_i^\top Q x_i + u_i^\top R u_i + \ell_\epsilon(\epsilon_i) + x_N^\top P x_N + \ell_\epsilon(\epsilon_N) \\ \text{s.t.} \; & x_{i+1} = Ax_i + Bu_i \\ & H_x x_i \le k_x + \epsilon_i \\ & H_u u_i \le k_u \\ & \epsilon_i \ge 0 \end{aligned}$$

7.2.2.2 Penalty Function Choices

Exact Penalty

The penalty function $\ell_\epsilon(\epsilon_i)$ should be as follows: If the original problem has a feasible solution $z^\star$, then the softened problem should have the same solution and $\epsilon = 0$.

Main Result

$$\ell_\epsilon(\epsilon) = v \cdot \epsilon$$

satisfies the requirement for any $v > \lambda^\star \ge 0$, where $\lambda^\star$ is the optimal Lagrange multiplier for the original problem.

Practical Penalty

Combine linear and quadratic terms for tuning:

$$\ell_\epsilon(\epsilon) = v \cdot \epsilon + s \cdot \epsilon^2$$

with $v > \lambda^\star$ and $s > 0$.

Extension to Multiple Constraints

Assume multiple constraints $g_j(z) \le 0$, $j = 1, \dots, r$. The penalty then reads

$$\ell_\epsilon(\epsilon) = v \cdot \|\epsilon\|_{1/\infty} + \epsilon^\top S \epsilon$$

where

- $\epsilon = [\epsilon_1, \dots, \epsilon_r]^\top$,
- $v > \|\lambda^\star\|_D$,
- and $S \succ 0$ can be used to weight violations differently.

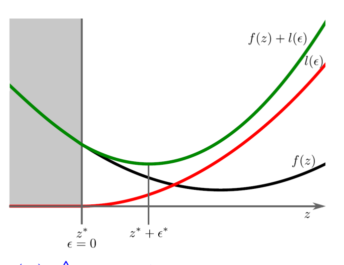
Note that $\|\cdot\|_D$ denotes the dual norm.

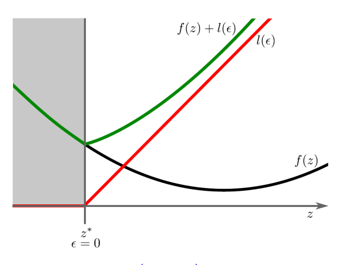
Comparison of Penalty Functions

The following properties hold for quadratic and linear penalties respectively

- Quadratic:
  + Well-posed QP (positive definite Hessian)
  - Increase in S hardens soft constraints
  - **Not** exact for any choice of $s > 0$
- Linear:
  + Allows for exact penalties if v large enough
  + If v is large enough, constraints satisfied if possible
  - Large v makes tuning hard and causes numerical issues

7.2.2.3 Comparison of Penalty Functions





$g(z) \triangleq z - z^\star \le 0$ induces feasible region (grey). Hence, the minimizer of the original problem is $z^\star$

- Quadratic penalty:

$$\begin{aligned} l_\epsilon(\epsilon) &= s \cdot \epsilon^2 \text{ for } \epsilon \ge 0, \\ l_\epsilon(\epsilon) &= 0, \text{ otherwise} \end{aligned}$$

  $\to$ minimizer of $f(z) + l_\epsilon(\epsilon)$ is $(z^\star + \epsilon^\star, \epsilon^\star)$ instead of $(z^\star, 0)$

- Linear penalty:

$$\begin{aligned} l_\epsilon(\epsilon) &= v \cdot \epsilon \text{ for } \epsilon \ge 0, \\ l_\epsilon(\epsilon) &= 0, \text{ otherwise,} \end{aligned}$$

  with v such that $v + \lim_{z \to z^\star} f'(z) > 0$

  $\to$ minimizer of $f(z) + l_\epsilon(\epsilon)$ is $(z^\star, 0)$

7.2.2.4 Simplification: Separation of Objectives

Step 1: Minimize Violation

$$\begin{aligned}\min_{u,\epsilon}\ & \sum_{i=0}^{N-1} \epsilon_i^\top S\epsilon_i + v^\top \epsilon_i\\ \text{s.t.}\ & x_{i+1} = Ax_i + Bu_i\\ & H_x x_i \le k_x + \epsilon_i\\ & H_u u_i \le k_u\\ & \epsilon_i \ge 0\end{aligned}$$

Step 2: Optimize Performance

$$\begin{aligned}\min_{u}\ & \sum_{i=0}^{N-1} x_i^\top Q x_i + u_i^\top R u_i + x_N^\top P x_N\\ \text{s.t.}\ & x_{i+1} = Ax_i + Bu_i\\ & H_x x_i \le k_x + \epsilon_i^{\min}\\ & H_u u_i \le k_u\end{aligned}$$

Properties

+ Simplifies tuning, constraints satisfied if possible
- Requires solving two optimization problems

8 Robust MPC

We assume an uncertain constrained system of the form:

$$x(k+1) = g\big(x(k), u(k), w(k); \theta\big),$$
$$x, u \in \mathcal{X}, \mathcal{U}, \quad w \in \mathcal{W}, \quad \theta \in \Theta$$

where

· w is random noise, changing with time, influencing system evolution

· θ are unknown, constant or slowly-varying parameters that impact the dynamics

Goals of Robust Constrained Control

Design control law $u(k) = \kappa(x(k))$ such that the system:

1. Satisfies constraints: $\{x(k)\} \subset \mathcal{X}, \{u(k)\} \subset \mathcal{U}$ for all disturbance realizations
2. Is stable: converges to a neighborhood of the origin
3. Optimizes (expected/worst-case) "performance"
4. Maximizes the set $\{x(0) \mid$ Conditions 1-3 are met $\}$

8.1 Modeling Uncertainty

8.1.1 Common Uncertainty Models

Measurement / Input Bias

$$g\big(x(k), u(k), w(k); \theta\big) = \widetilde{g}\big(x(k), u(k)\big) + \theta$$

θ unknown, but constant. Note that we called $\theta = d$ in 7.1.6.1.

Linear Parameter Varying System

$$g\big(x(k), u(k), \theta(k)\big) = \left(\sum_{j=0}^{n_\theta} \theta_j(k) A_j\right) x(k) + \left(\sum_{j=0}^{n_\theta} \theta_j(k) B_j\right) u(k)$$

time-varying parameters $\theta(k)$ describe a convex combination of A_j, B_j with $\mathbf{1}^\top \theta(k) = 1$, $\theta(k) \ge 0$.

Additive Stochastic Noise

$$g\big(x(k), u(k), w(k); \theta\big) = Ax(k) + Bu(k) + w(k)$$

where w comes from a known distribution. Used in stochastic MPC.

Additive Bounded Noise

In this course, the following noise models are considered:

$$g\big(x(k), u(k), w(k); \theta\big) = Ax(k) + Bu(k) + w(k), \quad w \in \mathcal{W}$$

where A, B are known, w is unknown but **bounded** and changing at each sampling instance. However, in contrast to the stochastic noise case, w now comes from a certain discrete set and not from a distribution.

Note that

· we may model many nonlinearities in this fashion, but often a conservative model

· the noise is *persistent*, i.e., it does not converge to zero in the limit

8.1.2 Robust Invariant Sets

Robust Positive Invariant Set

A set $\mathcal{O}_\mathcal{W}$ is said to be a robust positive invariant set for the autonomous system $x(k+1) = g\big(x(k), w(k)\big)$ if

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

The same concept is used for CL systems $x(k+1) = g\big(x(k), \kappa(x(k), w(k))\big)$.

A robust positive invariant set exists only for asymptotically stable systems.

Robust Invariant Set Conditions

Theorem: Geometric condition for robust invariance

A set $\mathcal{O}_\mathcal{W}$ is a robust positive invariant set if and only if

$$\mathcal{O}_\mathcal{W} \subseteq \text{pre}^\mathcal{W}(\mathcal{O}_\mathcal{W})$$

Computing Robust Invariant Sets

Similar to 5.1.3:

```
Input: $g, \mathcal{X}, \mathcal{W}$
Output: $\mathcal{O}_\infty^\mathcal{W}$
$\Omega_0 \leftarrow X$
while true do
    $\Omega_{i+1} \leftarrow \text{pre}^\mathcal{W}(\Omega_i) \cap \Omega_i$
    if $\Omega_{i+1} = \Omega_i$ then
        return $\mathcal{O}_\infty^\mathcal{W} = \Omega_i$
```

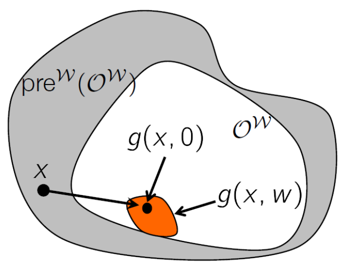
where Ω_0 is chosen so that it is as large as possible, choosing any $w \in \mathcal{W}$.

8.1.2.1 Robust Pre-Sets

Given a set Ω and dynamics $x(k+1) = g(x(k), w(k))$, the pre-set of Ω is is the set of states that evolve into the target set Ω in one time step **for all values** of the disturbance $w \in \mathcal{W}$:

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

Each state is now mapped into an entire possible set of next states:



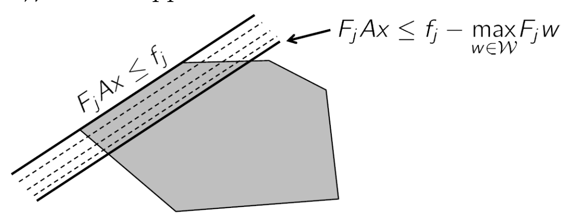
$\text{pre}^\mathcal{W}(\mathcal{O}^\mathcal{W})$
$g(x,0)$
$\mathcal{O}^\mathcal{W}$
x
$g(x,w)$

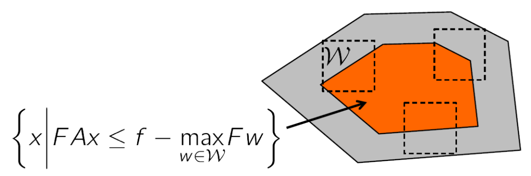
Computing Robust Pre-Sets for Linear Systems

Let $g(x(k), w(k)) = Ax(k) + w(k)$ and $\Omega := \{x \mid Fx \le f\}$. Then, fulfilling the state constraints can be seen as parallel displacement of the constraint boundaries

$$\begin{aligned}\text{pre}_\mathcal{W}(\Omega) &= \{x \mid Ax + w \in \Omega, \ \forall w \in \mathcal{W}\}\\ &= \{x \mid F_j Ax \le f_j - F_j w, \ \forall w \in \mathcal{W}\}\\ &= \{x \mid F_j Ax \le f_j - \max_{w\in\mathcal{W}} F_j\}\\ &= \{x \mid FAx \le f - h_\mathcal{W}(F)\}\end{aligned}$$

where $h_\mathcal{W}$ is the support function.





$\mathcal{W}$
$\left\{x \middle| FAx \le f - \max_{w\in\mathcal{W}} Fw\right\}$

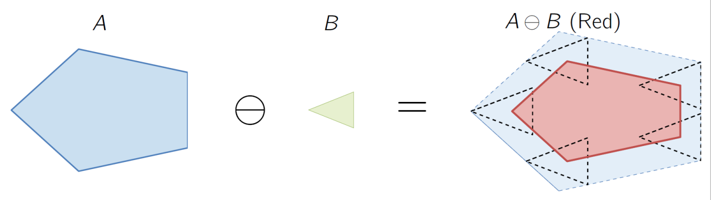
8.1.2.2 Minkowski Sum and Pontryagin Difference

Let A, B be subsets of $\mathbb{R}^n$. The Minkowski Sum is:

$$A \oplus B := \{x + y \mid x \in A, y \in B\}$$

The Pontryagin Difference is:

$$A \ominus B := \{x \mid x + e \in A, \forall e \in B\}$$



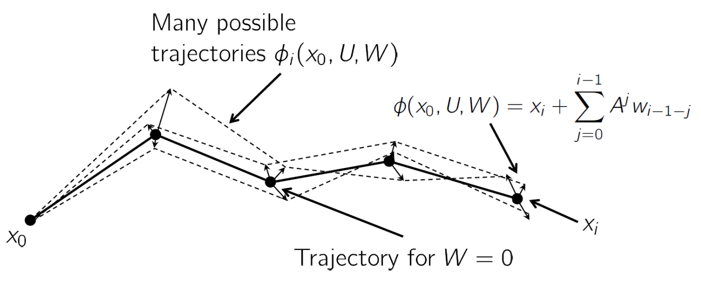
For scalar sets

$$[a, b] \oplus [c, d] = [a + c, b + d]$$
$$[a, b] \ominus [c, d] = [a - c, b - d]$$

8.2 Impact of Bounded Additive Noise

Define $\phi_i(x_0, U, W)$ as the state that the system will be in at time i if the state at time zero is x_0, we apply the input $U := \{u_0, \dots, u_{N-1}\}$ and we observe the disturbance $W := \{w_0, \dots, w_{N-1}\}$. Assume $0 \in \mathcal{W}$ (nominal trajectory).

8.2.1 Uncertain State Evolution



Many possible trajectories $\phi_i(x_0, U, W)$
$\phi(x_0, U, W) = x_i + \sum_{j=0}^{i-1} A^j w_{i-1-j}$
x_0
x_i
Trajectory for $W = 0$

In a system with bounded additive noise, the state evolution is given by:

$$\begin{aligned}\phi_1 &= Ax_0 + Bu_0 + w_0\\ \phi_2 &= A^2 x_0 + ABu_0 + Bu_1 + Aw_0 + w_1\\ \phi_i &= A^i x_0 + \sum_{j=0}^{i-1} A^j B u_{i-1-j} + \sum_{j=0}^{i-1} A^j w_{i-1-j}\\ &= x_i + \sum_{j=0}^{i-1} A^j w_{i-1-j}\end{aligned}$$

8.2.2 Cost

Generally, the cost function would contain a term for the unknown disturbance sequence $W \in \mathcal{W}^N$:

$$J(x_0, U, W) := \sum_{i=0}^{N-1} l(\phi_i(x_0, U, W), u_i) + l_f(\phi_N(x_0, U, W))$$

There are multiple methods to achieve **independence** of W:

· Minimize the expected value

$$J_N(x_0, U) = \mathbb{E}_w\Big[J(x_0, U, W)\Big]$$

· Consider the worst-case

$$J_N(x_0, U) = \max_{W\in\mathcal{W}^{N-1}} J(x_0, U, W)$$

· Consider the nominal case

$$J_N(x_0, U) = J(x_0, U, 0)$$

· Use an estimate of the disturbance (certainty equivalence)

$$J_N(x_0, U) = J(x_0, U, \widehat{W})$$

8.2.3 Robust Constraint Satisfaction

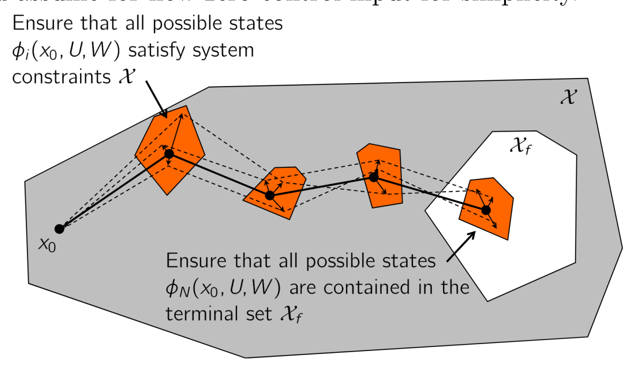
The dynamics and robust constraints along trajectories are:

$$\begin{aligned}\phi_{i+1} &= A\phi_i + Bu_i + w_i & i &= 0, \dots, N-1\\ u_i &\in \mathcal{U} & i &= 0, \dots, N-1\\ \phi_i &\in \mathcal{X}, \ \forall W \in \mathcal{W}^N & i &= 0, \dots, N-1\end{aligned}$$

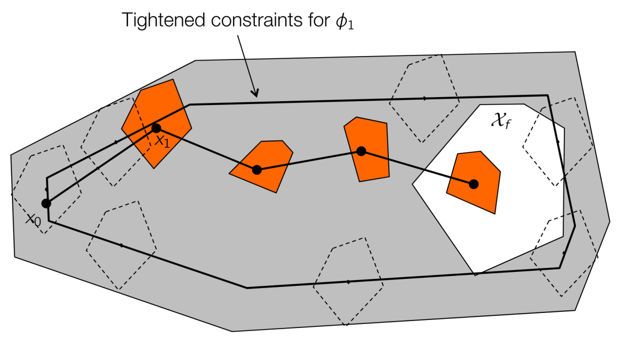
For the terminal set we have:

$$\begin{aligned}\phi_N &\in \mathcal{X}_f \subseteq \mathcal{X} & &\mathcal{X}_f \text{ robust invariant}\\ \phi_{i+1} &= A\phi_i + w_i & &i = N, \dots\end{aligned}$$

and assume for now zero control input for simplicity.



Ensure that all possible states $\phi_i(x_0, U, W)$ satisfy system constraints $\mathcal{X}$
$\mathcal{X}$
$\mathcal{X}_f$
x_0
Ensure that all possible states $\phi_N(x_0, U, W)$ are contained in the terminal set $\mathcal{X}_f$



8.2.3.1 State Constraints

The state constraints can be written as

$$\begin{aligned}\phi_i(x_0, U, W) &= \left\{ x_i + \sum_{j=0}^{i-1} A^i w_{i-1-j} \middle| W \in \mathcal{W}^i \right\} \subseteq \mathcal{X}\\ &= x_i \in \mathcal{X} \ominus \mathcal{F}_i\\ &= x_i \in \left\{x \middle| x + \bar{w} \in \mathcal{X}, \bar{w} \in \mathcal{F}_i\right\}\end{aligned}$$

where the *disturbance reachable set* (DRS)

$$\begin{aligned}\mathcal{F}_i &= \mathcal{W} \oplus A\mathcal{W} \oplus \cdots A^{i-1}\mathcal{W}\\ &= \oplus_{j=0}^{i-1} A^j \mathcal{W}\end{aligned}$$

shares the dynamics of the system:

$$\mathcal{F}_{i+1} = A\mathcal{F}_i \oplus \mathcal{W}$$

Polytopic Constraints

For the polytopic case $\mathcal{X} = \{x \mid Fx \le f\}$ we get

$$Fx_i + F\sum_{j=0}^{i-1} A^j w_{i-1-j} \le f, \quad \forall W \in \mathcal{W}^i$$

which is identical to

$$\begin{aligned}Fx_i &\le f - \max_{W\in\mathcal{W}^i} F\sum_{j=0}^{i-1} A^j w_{i-1-j}\\ &= f - h_{\mathcal{W}^i}\left(F\sum_{j=0}^{i-1} A^j\right)\end{aligned}$$

where the support function can be pre-computed offline.

8.2.3.2 Terminal Constraint

The terminal constraint

$$\phi_N(x_0, U, W) \in \mathcal{X}_f$$

is handled identically, i.e.

$$x_N \in \mathcal{X}_f \ominus \mathcal{F}_N$$

8.3 Robust Open-Loop MPC

Putting all the pieces together, we can write the robust open-loop MPC problem as the nominal problem with **tighter constraints**:

$$\begin{aligned}\min_U \ & \sum_{i=0}^{N-1} I(x_i,u_i) + I_f(x_N)\\ \text{subject to } & x_{i+1} = Ax_i + Bu_i\\ & x_0 = x(k)\\ & x_i \in \mathcal{X} \ominus \mathcal{F}_i\\ & u_i \in \mathcal{U}\\ & x_N \in \mathcal{X}_f \ominus \mathcal{F}_N\end{aligned}$$

where $\mathcal{X}_f \subseteq \mathcal{X}$ is a robust invariant set for the dynamics $x(k+1) = Ax(k) + w(k), \forall w$.

Stability and ROA
Stability analysis needs more general stability theory (advanced MPC).

Robust open-loop MPC potentially has a very small region of attraction, in particular for unstable systems. Use **feedback** to tackle it: 8.4.

8.4 Robust Closed-Loop MPC

In the problem formulation 8.3 we assumed that the controller would apply a fixed feed-forward input sequence in the future no matter what disturbance we observe. However, the performance of the robust open-loop MPC can be improved by using a control policy $\mu_i(x_i) : \mathbb{R}^n \to \mathbb{R}^m$ i.e. optimize over the sequence

$$\left\{u_0, \mu_1(x_1), \dots, \mu_{N-1}(x_{N-1})\right\}$$

where x_i is a function of the first i disturbances.

As we cannot optimize over arbitrary functions, often structural assumptions on the feedback law are established, such as:

Pre-stabilization

$$\mu_i(x_i) = Kx_i + v_i$$

Fixed K that stabilizes the CL (i.e. $A + BK$ is stable)
+ simple
− conservative
− Kx_i usually large → remaining optimization in v_i shrinks

Linear Feedback

$$\mu_i(x_i) = K_i x_i + v_i$$

Optimize over time-varying K_i and v_i
− Non-convex and difficult to solve

Disturbance Feedback

$$\mu_i(x_i) = \sum_{j=0}^{j-1} M_{ij} + v_i$$

Optimize over time-varying M_{ij} and v_i, hence over *all* previous disturbances
+ Convex version of *Linear feedback*
+ Least restrictive viable method
+ Very effective
− Computationally expensive (matrix optimization variables)

Constraint Tightening MPC & Tube MPC

$$\mu_i(x_i) = K(x_i - \bar{x}_i) + v_i$$

Fixed K that stabilizes the CL (i.e. $A + BK$ is stable) and optimize over $\bar{x}_i$ and v_i
+ Simple, often effective
+ FB on *deviation from* x_i → remaining opt. in v_i grows

8.4.1 Separation of Nominal and Disturbance Control

In constraint tightening MPC & tube MPC, the idea is to track a nominal trajectory. This allows us to split the control into two parts:

1. **Nominal Control** of the disturbance free system
$$z(k+1) = Az(k) + Bv(k)$$
2. **Disturbance Control** that compensates for deviations from the nominal trajectory
$$u_i = K(x_i - z_i) + v_i$$
The controller K is computed offline. We then optimize over z_i and v_i.

8.4.2 Error Dynamics

The linearity of the system allows to write the error dynamics as:

$$\begin{aligned}e_{i+1} &= x_{i+1} - z_{i+1}\\ &= Ax_i + Bu_i + w_i - Az_i - Bv_i\\ &= (A+BK)(x_i - z_i) + w_i\\ &= \underbrace{(A+BK)}_{A_K} e_i + w_i\\ &= A_K e_i + w_i\\ &= \sum_{j=0}^{i-1} A_K^j w_{i-j-1}, \qquad e_0 = 0\end{aligned}$$

We can find the disturbance reachable set (DRS) $\mathcal{F}_i$ that contains all possible states of the error:

$$\mathcal{F}_i = \mathcal{W} \oplus A_K\mathcal{W} ... \oplus A_K^{i-1}\mathcal{W} = \bigoplus_{j=0}^{i-1} A_K^j \mathcal{W}, \quad \mathcal{F}_0 := \{0\}$$

Advantages of Feedback
Assuming that $A + BK$ is stable, and the set $\mathcal{W}$ is bounded, the error remains bounded. Hence, the method also works for open-loop **unstable** systems. This is a main difference compared to robust OL MPC.

8.4.2.1 Minimum Robust Invariant Set

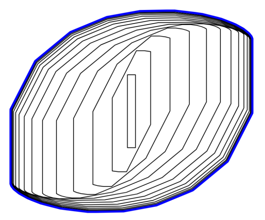
The sequence $\mathcal{F}_i$ converges to a set $\mathcal{F}_\infty$ that is the minimum robust invariant (mRPI) set for the error dynamics:

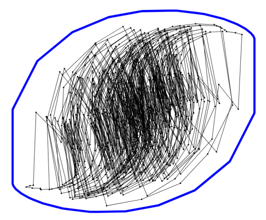
$$\mathcal{F}_\infty = \bigoplus_{j=0}^{\infty} A_K^j \mathcal{W}$$

which can be found by

```
$\Omega_0 \leftarrow \{0\}$
while do
    $\Omega_{i+1} \leftarrow \Omega_i \oplus A^i\mathcal{W}$
    if $\Omega_{i+1} = \Omega_i$ then
        return $\mathcal{F}_\infty = \Omega_i$
```

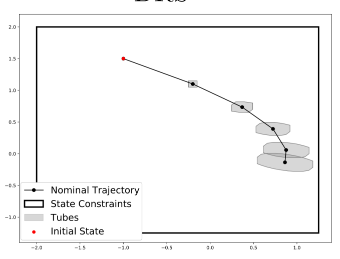
Note that the set $\mathcal{F}_i$ increases in size with i and converges to $\mathcal{F}_\infty$. If convergence does not happen in finite time, there are methods to slightly enlarge $\mathcal{F}_i$ so that $\mathcal{F}_i > F_\infty$.


Sets $A^j\mathcal{W}$ converging to minimal robust invariant set $\mathcal{F}_\infty$ in the limit

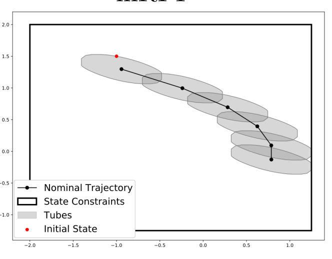

The state trajectory will stay in the set $\mathcal{F}_\infty$ for all time

Comparison of the two variants:

DRS



mRPI



Scalar Linear DTI
For a **scalar** linear discrete time system

$$x(k+1) = Ax(k) + Bu(k) + w(k)$$

with $u(k) = Kx(k)$ and $w \in \mathcal{W}$ the mRPI is given by

$$\mathcal{F}_\infty = \left[\frac{a}{1-A_K}, \frac{b}{1-A_K}\right]$$

with

$$A_K = A + BK \quad \text{and} \quad \mathcal{W} = [a,b]$$

8.4.3 Input-to-State Stability

Assume that the optimal cost J^* is Lipschitz continuous (true for linear systems, convex constraints and continuous stage costs)

$$|J^*(Ax + Bu^*(x) + w) - J^*(Ax + Bu^*(x))| \le \gamma\|w\|$$

for some $\gamma > 0$.

Then, the Lyapunov decrease can be bounded as

$$J^*(Ax + Bu^*(x) + w) - J^*(x) \le -I(x, u^*(x)) + \gamma\|w\|$$

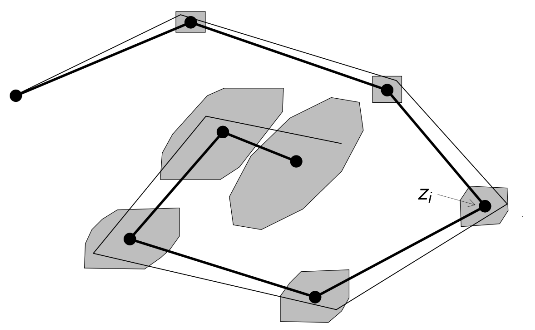
Hence,
· amount of decrease $\|x\|$
· amount of increase upper bounded by $\max_{w\in\mathcal{W}} \gamma\|w\|$
and the system moves toward the origin until there is a balance between the size of x and the size of w.

8.5 Constraint-Tightening MPC

In order to satisfy the state constraints for all possible disturbances W the constraints on the state x must be tightened:

$$z_i \oplus \mathcal{F}_i \subseteq \mathcal{X} \quad \Leftrightarrow \quad z_i \in \mathcal{X} \ominus \mathcal{F}_i$$

where the DRS $\mathcal{F}_i$ can be computed offline.



Similarly for the input constraints ($u_i = K(x_i - z_i) + Ke_i + v_i$):

$$u_i \in v_i \oplus K\mathcal{F}_i \subseteq \mathcal{U} \quad \Leftrightarrow \quad v_i \in \mathcal{U} \ominus K\mathcal{F}_i$$

Intuition: The method builds on the assumption that error pre-stabilization can be applied. Hence, we must ensure that v_i leaves enough control action to the prestabilization part → shrink set for v_i.

8.5.1 MPC Problem

The robust constraint-tightening MPC Problem over the nominal state z and nominal input v is given by:

$$\begin{aligned}\min_{Z,V} \ & \sum_{i=0}^{N-1} I(z_i, v_i) + I_f(z_N)\\ \text{subj. to } & z_{i+1} = Az_i + Bv_i && \forall i = 0,1,\dots,N-1\\ & z_i \in \mathcal{X} \ominus \mathcal{F}_i && \forall i = 0,1,\dots,N-1\\ & v_i \in \mathcal{U} \ominus K\mathcal{F}_i && \forall i = 0,1,\dots,N-1\\ & z_N \in \mathcal{X}_f^{ct} \ominus \mathcal{F}_N\\ & z_0 = x(k)\end{aligned}$$

Note that
· The (feedback) control law is given by
$$u(k) = v_0^* + K(x(k) - z_0^*) = v_0^*$$
(no actual FB applied → method simply tightens constr.)
· The terminal set $\mathcal{X}_f^{ct}$ is a robust invariant set, which for z_N must additionally be tightened by $\mathcal{F}_N$ to ensure we actually end up in $\mathcal{X}_f^{ct}$.

8.5.1.1 Constraint-Tightening MPC Assumptions

1. The stage cost $I(z,v)$ is positive definite and only zero at the origin.
2. The terminal set $\mathcal{X}_f^{ct}$ is a robust positively invariant set for the dynamics $x(k+1) = Ax(k) + Bu(k) + w(k)$ under the terminal controller $\kappa_f^{ct} z(k) = K_{ct}\, z(k)$:
$$(A + BK_{ct})z + w \in \mathcal{X}_f^{ct}, \quad \forall z \in \mathcal{X}_f^{ct}, \forall w \in \mathcal{W}$$
All state and input constraints are satisfied in $\mathcal{X}_f^{ct}$.
$$\mathcal{X}_f^{ct} \subseteq \mathcal{X}, \quad K_{ct}\mathcal{X}_f^{ct} \subseteq \mathcal{U}$$

8.5.1.2 Recursive Feasibility Guarantee

Let the terminal ingredients $(I_f, \mathcal{C}_f^{ct})$ be chosen such that
· $\mathcal{X}_f^{ct} \subseteq \mathcal{X}$
· for all $z \in \mathcal{X}_f^{ct}$:
  · $Kz \in \mathcal{U}$
  · $(A+BK)z + w \in \mathcal{X}_f^{ct}$, $\forall w \in \mathcal{W}$
  · $I_f\big((A+BK)z\big) - I_f(z) \le -I(z, \pi_f(z))$

Let
· $\mathcal{X}_N$ be the feasible set
· and $V^*(x(k))$ be the optimizer of the robust constraint-tightening MPC problem for $x(k) \in \mathcal{X}_N$.

Then **robust invariance** is achieved

$$Ax(k) + Bv_0^*(x(k)) + w \in \mathcal{X}_N, \ \forall w \in \mathcal{W}$$

i.e. the problem is **recursively feasible**.

Intuition The computed current trajectory is feasible for *any* disturbance, and hence also for the one that actually occurs.

8.5.1.3 Stability Guarantee

Uses input-to-state stability → details in advanced MPC

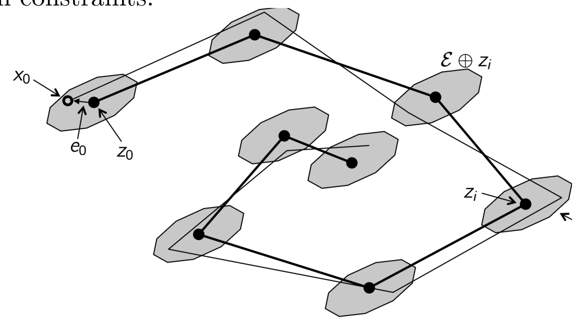
8.6 Tube MPC

In contrast to constraint-tightening MPC, the tube MPC bounds the maximum error with a fixed size set

$$\mathcal{E} : e_i \in \mathcal{E} \Rightarrow e_{i+1} \in \mathcal{E},$$

ideally the minimum RPI set $\mathcal{F}_\infty$. A larger set can be chosen but **must** be invariant.

Therefore, the goal is to plan a nominal trajectory z_i i.e. the tube center, such that all possible state trajectories $z_i \oplus \mathcal{E}$ are within constraints.



Constraint-Tightening

Similar to before, the constraints on the state x must be tightened as

$$z_i \oplus \mathcal{E} \subseteq \mathcal{X} \quad \Leftrightarrow \quad z_i \in \mathcal{X} \ominus \mathcal{E}$$

where the mRPI $\mathcal{E}$ again can be computed offline.

Similarly for the input constraints:

$$u_i \in K\mathcal{E} \oplus v_i \subset \mathcal{U} \quad \Leftrightarrow \quad v_i \in \mathcal{U} \ominus K\mathcal{E}$$

8.6.1 Tube MPC Problem

The robust tube MPC Problem over the nominal state z and nominal input v with tightened constraints is given by:

$$\begin{aligned} \min_{Z,V} \; & \sum_{i=0}^{N-1} I(z_i, v_i) + I_f(z_N) \\ \text{subj. to } & z_{i+1} = Az_i + Bv_i && \forall i = 0, 1, \dots, N-1 \\ & z_i \in \mathcal{X} \ominus \mathcal{E} && \forall i = 0, 1, \dots, N-1 \\ & v_i \in \mathcal{U} \ominus K\mathcal{E} && \forall i = 0, 1, \dots, N-1 \\ & z_N \in \mathcal{X}_f \\ & x_0 \in z_0 \oplus \mathcal{E} \end{aligned}$$

with the applied control law

$$\mu_{\text{tube}}(x) = K(x - z_0^*(x)) + v_0^*(x)$$

Note that

- the cost is calculated with respect to the tube centers (nominal system)
- the terminal set is w.r.t. the tightened constraints
- The first tube center z_0 is also an optimization variable i.e. z_0 has to be within $\mathcal{E}$ of x_0
- The nominal dynamics can be different from the disturbance dynamics
- The state trajectory only converges to a neighborhood of the origin $\rightarrow \sum_{i=0}^{\infty} \ell(x_i, u_i)$ can be infinite

Comparison to Constraint-Tightening MPC

+ The terminal set $\mathcal{X}_f$ is simply the invariant set from the OL problem. **No** robust invariance needed!
+ Nominal and disturbed dynamics are completely decoupled
+ Region of convergence can be specified
- Feasible set smaller

Implementation:
Offline

1. Choose a stabilizing controller K so that $A+BK$ is stable.
2. Compute the minimal robust invariant set $\mathcal{E} = \mathcal{F}_\infty$ for the system $x(k+1) = (A+BK)x(k) + w(k), w \in \mathcal{W}$[1]
3. Compute the tightened constraints

$$\widetilde{\mathcal{X}} := \mathcal{X} \ominus \mathcal{E}$$
$$\widetilde{\mathcal{U}} := \mathcal{U} \ominus K\mathcal{E}$$

4. Choose terminal weight function I_f and constraint $\mathcal{X}_f$ satisfying assumptions stated in Section 8.6.1.2.

Online

1. Measure / estimate state x
2. Solve the Tube MPC optimization problem
3. Set the input to $u = K(x - Z_0^*(x)) + v_0^*(x)$

8.6.1.1 Example: Offline Constraint Tightening

Assume $\mathcal{X} = \{x \mid \|x\|_\infty \le 3\}$, i.e.

$$\mathcal{X} = \left\{ x \,\middle|\, \begin{bmatrix} I \\ -I \end{bmatrix} x \le \begin{bmatrix} 3 \\ 3 \end{bmatrix} \right\}$$

Let $\mathcal{E} = \{e \mid Fe \le f\}$. Then the tightened state constraints can be found by solving a set of LPs:

$$\mathcal{X} \ominus \mathcal{E} = \left\{ x \,\middle|\, \begin{bmatrix} I \\ -I \end{bmatrix} x \le \begin{bmatrix} 3 - \max_{e \in \mathcal{E}} [1 \;\, 0] e \\ 3 - \max_{e \in \mathcal{E}} [0 \;\, 1] e \\ 3 - \max_{e \in \mathcal{E}} [-1 \;\, 0] e \\ 3 - \max_{e \in \mathcal{E}} [0 \;\, -1] e \end{bmatrix} \right\}$$

Similarly, for the input constraints $\mathcal{U} = \{u \mid |u| \le 0.5\}$:

$$\mathcal{U} \ominus K\mathcal{E} = \left\{ u \,\middle|\, \begin{bmatrix} 1 \\ -1 \end{bmatrix} u \le \begin{bmatrix} 0.5 - \max\{Ke \mid Fe \le f\} \\ 0.5 - \min\{Ke \mid Fe \le f\} \end{bmatrix} \right\}$$

8.6.1.2 Tube MPC Assumptions

Similar to standard MPC, three main conditions are needed to ensure stability and recursive feasibility of the nominal trajectory:

1. The stage cost is a positive definite function: strictly positive and only zero at the origin.
2. The terminal set is invariant for the **nominal system** under the local controller $\kappa_f(z)$:
$$Az + B\kappa_f(z) \in \mathcal{X}_f, \quad \forall z \in \mathcal{X}_f$$
and satisfies the **tightened** constraints:
$$\mathcal{X}_f \subseteq \mathcal{X} \ominus \mathcal{E}, \quad \kappa_f(z) \in \mathcal{U} \ominus K\mathcal{E} \quad \forall z \in \mathcal{X}_f$$
3. Terminal cost is a Lyapunov function in $\mathcal{X}_f$:
$$I_f(Az + B\kappa_f(z)) - I_f(z) \le -I(z, \kappa_f(z)), \quad \forall z \in \mathcal{X}_f$$

Note that w can be ignored in 2. as nominal and disturbance dynamics are decoupled.

8.6.1.3 Nominal Stability and Recursive Feasibility

Theorem: Robust Invariance of Tube MPC
Let $\mathcal{Z}$ be the feasible set, i.e $\mathcal{Z} := \{x \mid \mathcal{Z}(x) \neq \emptyset\}$. Then $\mathcal{Z}$ is a robust invariant set for the closed-loop system

$$x(k+1) = Ax(k) + B\mu_{\text{tube}}(x(k)) + w(k)$$

under the constraints $x \in \mathcal{X}$, $u \in \mathcal{U}$.

Theorem: Robust Stability of Tube MPC
The state $x(k)$ of the system

$$x(k+1) = Ax(k) + B\mu_{\text{tube}}(x(k)) + w(k)$$

converges in the limit to the set $\mathcal{E}$ and hence,

$$\lim_{k \to \infty} \operatorname{dist}(x(k), \mathcal{E}) = 0$$

where $\operatorname{dist}(\cdot,\cdot)$ denotes any distance function.

The cost for the nominal trajectory vanishes:

$$\lim_{k \to \infty} J(z_0^*(x(k))) = 0 \quad \Rightarrow \quad \lim_{k \to \infty} z_0^*(x(k)) = 0$$

8.6.1.4 Proof Sketch

Recursive Feasibility
Given the optimal solution $(v_0^*, \dots, v_{N-1}^*, z_0^*, \dots, z_N^*)$ for $x(k)$, the next state is

$$x(k+1) = Ax(k) + BK(x(k) - z_0^*) + Bv_0^* + w, \quad w \in \mathcal{W}$$

By construction, $x(k+1) \in z_1^* \oplus \mathcal{E}$. The shifted trajectory

$$\{v_1^*, \dots, v_{N-1}^*, \kappa_f(z_N^*)\}, \quad \{z_1^*, \dots, z_N^*, Az_N^* + B\kappa_f(z_N^*)\}$$

is feasible for all $x(k+1)$.

Details:

- Feasibility of $\{v_1^*, \dots, v_{N-1}^*, \kappa_f(z_N^*)\}$
  - $v_i^* \in \mathcal{U} \ominus K\mathcal{E}$ for $i = 1, \dots, N-1$ given from feas. at k.
  - $\kappa_f(z_N^*) \in \mathcal{U} \ominus K\mathcal{E}$ for $z_N^* \in \mathcal{X}_f$ by Assumption 2. (Section 8.6.1.2)
- Feasibility of $\{z_1^*, \dots, z_N^*, Az_N^* + B\kappa_f(z_N^*)\}$
  - $z_i^* \in \mathcal{X} \ominus \mathcal{E}$ for $i = 1, \dots, N-1$ given from feas. at k.
  - $z_N^* \in \mathcal{X}_f \subseteq \mathcal{X} \ominus \mathcal{E}$ given from feasibility at k.
  - $Az_N^* + B\kappa_f(z_N^*) \in \mathcal{X}_f$ from invariance of $\mathcal{X}_f$ by Assumption 2. (Section 8.6.1.2)
  - $x(k+1) \in z_1^* \oplus \mathcal{E}$ follows from:

$$\begin{aligned} \underbrace{x(k+1) - z_1^*}_{e_{k+1}} &= Ax(k) + BK(x(k) - z_0^*) + Bv_0^* + w \\ &\quad - Az_0^* + Bv_0^* \\ &= (A + BK)\underbrace{(x(k) - z_0^*)}_{e_k} + w(k) \\ &\in \mathcal{E} \text{ if } (x(k) - z_0^*) \in \mathcal{E} \\ &\Rightarrow x(k+1) \in z_1^* \oplus \mathcal{E} \end{aligned}$$

Stability
As in standard MPC, we have the value function expression

$$J^*(x(k)) = \sum_{i=0}^{N-1} \ell(z_i^*, v_i^*) + \ell_f(z_N^*)$$

At the next time step $k+1$, it holds that

$$\begin{aligned} J^*(x(k+1)) &\le \sum_{i=1}^{N} \ell(z_i^*, v_i^*) + \ell_f(z_{N+1}) \\ &= J^*(x(k)) - \underbrace{\ell(z_0^*, v_0^*)}_{\ge 0} \\ &\quad - \underbrace{\ell_f(z_N^*) - \ell_f(z_{N+1}) - \ell(z_N^*, \kappa_f(z_N^*))}_{\le 0} \end{aligned}$$

where the last inequality uses the fact that ℓ_f is a Lyapunov function on $\mathcal{X}_f$.

9 Numerical Methods and Implementation

9.1 Explicit MPC

Online computation time of the optimal control law can be drastically reduced by pre-computing the control law as a function of the state x. In other words, for **small systems** (3-6 states),

1. The optimization problem is **solved offline and parametrically** for a set of states.
2. During runtime, the control input is **queried from the pre-computed solution** (piecewise affine for linear system/ constraints).

9.1.1 Active Set and Critical Region

Active Set
The active set $A(x)$ is the set of indices of active constraints at a given state $x \in \mathcal{X}_0$:

$$A(x) = \left\{ j \in \{1, \dots, m\} \,\middle|\, G_j U - E_j x(k) = w_j \right\}$$

Its complement is the set of non-active constraints:

$$NA(x) = \left\{ j \in \{1, \dots, m\} \,\middle|\, G_j U - E_j x(k) < w_j \right\}$$

Critical Region
The critical region CR is the set of states x for which the same active set $A(x)$ is active at the optimum:

$$CR = \{x \in \mathcal{X} : A(x) = A(x^*)\}$$

9.1.2 Quadratic Cost

The CFTOC problem

$$\begin{aligned} J^*(x(k)) = \min_U & \begin{bmatrix} U^\top & x(k)^\top \end{bmatrix} \begin{bmatrix} H & F^\top \\ F & Y \end{bmatrix} \begin{bmatrix} U \\ x(k) \end{bmatrix} \\ & \text{subj. to } GU \le w + E\,x(k) \end{aligned}$$

is a **multiparametric quadratic program** (MPQP) with the following properties:

- The optimal solution is continuous and piecewise affine on polyhedra CR^j
$$u_0^* = \kappa(x(k)) = F^j x + g^j \quad \text{if } x \in CR^j \; j = 1, \dots, N^r$$
- The polyhedra $CR^j = \left\{ x \in \mathbb{R}^n \,\middle|\, H^j x \le K^j \right\}$ are a partition of the feasible polyhedron $\mathcal{X}_0$.
- The value function $J^*(x(k))$ is convex and piecewise quadratic on polyhedra.

9.1.3 1/∞-Norm Cost

The CFTOC problem

$$\begin{aligned} \min_U \; & c^\top U \\ \text{subj. to } \; & \bar{G}U \le \bar{w} + \bar{E}x(k) \end{aligned}$$

Is a multiparametric linear program (MPLP) with the following properties:

- The optimal solution is continuous and piecewise affine on polyhedra CR^j
$$u_0^* = \kappa(x(k)) = F^j x + g^j \quad \text{if } x \in CR^j \; j = 1, \dots, N^r$$
- The polyhedra $CR^j = \left\{ x \in \mathbb{R}^n \,\middle|\, H^j x \le K^j \right\}$ are a partition of the feasible polyhedron $\mathcal{X}_0$.
- In case of multiple optimizers a piecewise affine control law exists.
- The value function $J^*(x(k))$ is convex and piecewise linear on polyhedra.

9.1.4 Point Location

Given m regions, there are multiple methods to find the critical region CR^j for a given state $x(k)$. Then, the corresponding piecewise affine function must be evaluated at $x(k)$ to obtain the control input.

Sequential Search $\mathcal{O}(m)$
Simply check each polyhedron until the one containing $x(k)$ is found:

```
given $x = x(k)$
for each $j$ do
    if $A_j x + b_j \le 0$ then
        $x$ is in region $j$
```

Logarithmic/Tree Search $(\mathcal{O}(\log(m)))$
By constructing a search tree offline, the critical region can potentially be found in logarithmic time. The search tree is constructed by recursively splitting the polyhedra CR^j by hyperplanes. Reasonable for < 1000 regions.

9.2 Iterative MPC

Given an initial guess x_0, cost function $f(x)$, and a feasible set $\mathbb{Q}$, **iterative optimization methods** compute a sequence of iterates

$$x_{k+1} = \Psi(x_k, f, \mathbb{Q}), \quad K = 0, 1, \dots, m-1$$

that are ϵ-stationary

$$\|f(x^{(m)}) - f(x^*)\| \le \epsilon$$

and sufficiently feasible

$$\operatorname{dist}(x^{(m)}, \mathbb{Q}) \le \delta$$

9.2.1 Unconstrained Minimization

Problem
Solve

$$\min_x f(x)$$

where we assume

- f convex and twice continuously differentiable
- f differentiable at x^*
- optimal value $p^* = \min_x f(x)$ finite

The goal is to iteratively solve for the necessary and sufficient condition

$$\nabla f(x^*) = 0$$

Descent Methods
Descent methods update the current iterate x_k in the search direction Δx_k with a step size h_k:

$$x_{k+1} = x_k + h_k \Delta x_k \quad \rightarrow \quad f(x_{k+1}) < f(x_k)$$

where

- Δx is a descent direction
- There exists a $h_k > 0$ such that we obtain a cost decrease, if we step to a certain extent into the opposite gradient

direction:

$$\exists h_k > 0 \rightarrow f(x_{k+1}) < f(x_k) \text{ if } \nabla f(x_k)^\top \Delta x_k < 0$$

The exact direction is to be chosen (e.g. opposite gradient or Newton).

Algorithm

```
given $x_0 \in \mathrm{dom}(f)$
repeat
    1. Compute a descent direction $\Delta x_k$
    2. Line search: Choose step size $h_k > 0$ such that
       $f(x_k + h_k \Delta x_k) < f(x_k)$
    3. Update $x_{k+1} := x_k + h_k \Delta x_k$
until termination condition (e.g. $f(x_k) - f(x^*) \leq \varepsilon_1$ or $\|x_k - x_{k-1}\| \leq \varepsilon_2$)
```

9.2.1.1 Gradient Descent / First-Order Method

Assume

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \quad \forall x, y \in \mathbb{R}^n$$

i.e.

- ∇f is Lipschitz-continuous with constant L
- f can be upper bounded by a quadratic function

$$f(x) \leq f(y) + \nabla f(y)^\top (x - y) + \frac{L}{2}\|x - y\|^2 \quad \forall x, y \in \mathbb{R}^n$$

This quadratic upper bound can be minimized by choosing the negative gradient $-\nabla f$ as the direction of descent and step size $\frac{1}{L}$ (Lipschitz constant of the gradient). The update rule becomes

$$x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$$

which is guaranteed to converge to a local minimum of f.

Finding L

Assuming f is L-smooth, the Lipschitz constant L can be determined globally. It is the maximum spectral norm of the Hessian $\nabla^2 f(x)$, i.e.

$$\begin{aligned} L &= \max_{x \in \mathcal{X}} \|\nabla^2 f(x)\|_2 \\ &= |\max_{x \in \mathcal{X}} \sigma_{\max}(\nabla^2 f(x))| \\ &= |\lambda_{\max}(\nabla^2 f(x))| \end{aligned}$$

As the Hessian is always symmetric, L is simply its **largest absolute eigenvalue**.

9.2.1.2 Newton's Method / Second-Order Method

Newton's method minimizes the second-order Taylor expansion of f around the current iterate x_k:

$$\begin{aligned} x_{k+1} &= \arg\min_x \; f(x_k) + \nabla f(x_k)^\top (x - x_k) + \cdots \\ &\quad + \frac{1}{2}(x - x_k)^\top \nabla^2 f(x_k)(x - x_k) \\ x_{k+1} &= x_k \underbrace{-\left(\nabla^2 f(x_k)\right)^{-1} \nabla f(x_k)}_{\text{Newton's direction}} \end{aligned}$$

As the second-order Taylor expansion is **not** necessarily an upper bound, we can obtain a cost increase. Hence, the remaining problem is to choose $h_k > 0$ such that the update decreases f:

$$x_{k+1} = x_k - h_k \left(\nabla^2 f(x_k)\right)^{-1} \nabla f(x_k)$$

for example by using line search.

9.2.1.3 Line Search

Exact Line Search

Compute the best step size h_k along the direction of descent Δx_k:

$$h_k = \arg\min_{h > 0} f(x_k + h\Delta x_k)$$

This is an optimization in one variable which can be solved by bisection (slow, many evaluations of f).

Backtracking Line Search (Inexact)

Find a step size h_k that decreases the cost function f sufficiently by repeatedly downscaling the step size:

```
Initialize $h_k = 1$
while $f(x_k + h_k \Delta x_{nt}) > f(x_k) + \alpha h_k \nabla f(x_k)^\top \Delta x_{nt}$ do
    $h_k \leftarrow \beta h_k$
```

with $\alpha \in (0, 0.5)$ and $\beta \in (0, 1)$.
The condition

$$f(x_k + h_k \Delta x_{nt}) > f(x_k) + \alpha h_k \nabla f(x_k)^\top \Delta x_{nt}$$

is called **Armijo condition** and quantifies the cost decrease compared to the predicted cost decrease of the linear approximation.

9.2.2 Constrained Minimization

- **PGD**: better for small horizon lengths, faster iterations, but more iterations needed.
- **Interior Point Method**: better for large horizon lengths, fewer iterations, but slower iterations.

9.2.2.1 Equality Constrained Newton's Method

Incorporating equality constraints

$$\begin{aligned} \min_x \quad & f(x) \\ \text{subj. to} \quad & Ax = b \end{aligned}$$

into Newton's method (and rewriting $x - x_k = \Delta x$) yields

$$\begin{aligned} \Delta x_{nt}(x_k) \in \arg\min_{\Delta x_k} \quad & \frac{1}{2}\Delta x \nabla^2 f(x_k) \Delta x + \nabla f(x_k)\Delta x \\ \text{subj. to} \quad & A\Delta x_k = \underbrace{-Ax_k + b}_{=0 \text{ optimally}} \end{aligned}$$

Descent Direction

By optimality on the above problem, we have to solve

$$\begin{aligned} \nabla^2 f(x_k)\Delta x + \nabla f(x_k) + A^T \nu &= 0 \\ A\Delta x &= 0 \end{aligned}$$

This, however, is a linear system of equations, yielding the wanted search direction:

$$\begin{bmatrix} \nabla^2 f(x_k) & A^\top \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \lambda_k \end{bmatrix} = \begin{bmatrix} -\nabla f(x_k) \\ 0 \end{bmatrix} \quad \begin{matrix} \text{Stationarity} \\ A\Delta x = 0 \end{matrix}$$

If the equality constraint is fulfilled at initilization, the nullspace constraint $A\Delta x = 0$ enforces that the next x_{k+1} still satisfies the equality constraint:

$$Ax_{k+1} = Ax_k + h_k \underbrace{A\Delta x_k}_{=0} = b$$

Conclusion

Equality constraints yield easy optimization problems as the search directions can be found by solving a simple linear system.

9.2.2.2 Projected Gradient Method

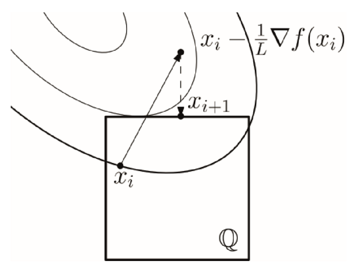
Problem

Consider the constrained convex optimization problem

$$\min f(x) \quad \text{subject to} \quad x \in \mathbb{Q}$$

where f is convex and L-smooth, and the feasible set $\mathbb{Q}$ is convex.

Step Projection



Constraints on x can be incorporated by projecting the gradient onto $\mathbb{Q}$ using the euclidian projection:

$$\begin{aligned} x_{k+1} &= \Pi_{\mathbb{Q}}(x_k - h_k \nabla f(x_k)) \\ \Pi_{\mathbb{Q}}(x) &= \arg\min_{y \in \mathbb{Q}} \|y - x\|^2 \end{aligned}$$

Under our assumptions, if we choose step size $h_k = \frac{1}{L}$, Projected Gradient Descent (PGD) converges to a local minimum of f with convergence rates similar to the unconstrained case.

MPC

In the MPC problem setup, the optimization variable is the control input U. Therefore, **input constraints** can be incorporated by using PGD in the CFTOC formulation with substitution.

MPC with **state constraints** are more complicated as the projection of the intersection $(\mathcal{U} \times \mathcal{X}) \cap \{z | Az = b\}$ is not trivial. One approach could be to solve the dual problem (simple inequalities $\lambda >= 0$ with cheap projection), which would require special attention to preserving the feasibility of the primal problem.

9.2.2.3 Interior Point Method

Problem

$$\begin{aligned} \min_x \quad & f(x) \\ \text{subj. to} \quad & g_i(x) \leq 0 \quad i = 1, \ldots, m \\ & Cx = d \end{aligned}$$

Assumptions

- f, g_i are convex, and twice continuously differentiable.
- $f(x^*)$ is finite and attained
- strict feasibility: $\exists x$ such that $g_i(x) < 0$ and $Cx = d$.
- feasible set is closed and compact
- strong duality holds $\rightarrow$ KKT conditions can be used

Primal-Dual relaxed KKT Conditions

Primal-Dual interior-point methods solve the following relaxed KKT system of equations for both, the primal and dual variables:

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

where

- $s \in \mathbb{R}^m$ are slack variables
- $\{(x, \nu, \lambda, s) | \text{above eqns. hold}\}$ is called *primal-dual central path*
- one attempts to successively reduce κ to zero (central path)

At every iteration, the KKT conditions are linearized at the current iterate $(x_k, \nu_k, \lambda_k, s_k)$:

$$\begin{bmatrix} H & C^\top & G^\top & 0 \\ C & 0 & 0 & 0 \\ G & 0 & 0 & \mathbb{I} \\ 0 & 0 & S & \Lambda \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \nu \\ \Delta \lambda \\ \Delta s \end{bmatrix} = - \begin{bmatrix} \nabla f(x) + C^\top \nu + G^\top \lambda \\ Cx - d \\ g(x) + s \\ S\lambda - v \end{bmatrix}$$

where

$$\begin{aligned} S &= \mathrm{diag}(s_1, \ldots, s_m) \\ \Lambda &= \mathrm{diag}(\lambda_1, \ldots, \lambda_m) \\ H(x, \lambda) &= \nabla^2 f(x) + \sum_{i=1}^{m} \lambda_i \nabla^2 g_i(x) \\ G(x)^\top &= \begin{bmatrix} \nabla g_1(x) & \nabla g_2(x) & \cdots & \nabla g_m(x) \end{bmatrix} \end{aligned}$$

and v replaces κ in the KKT conditions (further relaxation).

The resulting direction $\Delta[x, \nu, \lambda, s](v)$ is found by solving the linear system and depends on the choice of the relaxation parameter v. For

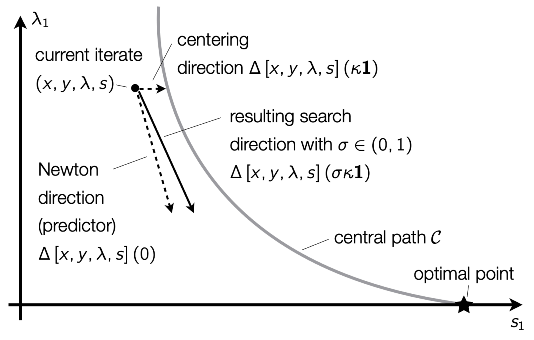
- $v = 0$, the direction is a **Newton step**.
- $v = \kappa\mathbf{1}$, the direction is called a centering direction that approaches the central path.

Predictor-Corrector Method

Predictor-Corrector methods use a linear combination of the two directions, i.e.

$$\Delta[x, \nu, \lambda, s](v) = \Delta[x, \nu, \lambda, s](\sigma\kappa\mathbf{1})$$

where $\sigma \in (0, 1)$, which ensures fast convergence.



λ_1
current iterate
(x, y, λ, s)
centering
direction $\Delta[x, y, \lambda, s](\kappa\mathbf{1})$
resulting search
direction with $\sigma \in (0, 1)$
$\Delta[x, y, \lambda, s](\sigma\kappa\mathbf{1})$
Newton
direction
(predictor)
$\Delta[x, y, \lambda, s](0)$
central path $\mathcal{C}$
optimal point
s_1

10 Appendix

10.1 Receding Horizon Control Notation

Consider the DT LTI model

$$x(k+1) = Ax(k) + Bu(k), \quad y(k) = Cx(k),$$

where $x(k) \in \mathcal{X}$, $u(k) \in \mathcal{U}$, $\forall k \geq 0$.

10.1.1 Absolute Time Notation

The CFTOC problem in absolute time notation

$$J^*_{k\to k+N|k}(x(k)) = \min_{U_{k\to k+N|k}} l_f(x_{k+N|k}) + \sum_{i=0}^{N-1} l(x_{k+i|k}, u_{k+i|k})$$

subj. to

$$\begin{aligned} x_{k+i+1|k} &= Ax_{k+i|k} + Bu_{k+i|k}, \quad i = 0, \dots, N-1, \\ x_{k+i|k} &\in \mathcal{X}, \quad u_{k+i|k} \in U, \quad i = 0, \dots, N-1, \\ x_{k+N|k} &\in \mathcal{X}_f, \\ x_{k|k} &= x(k), \end{aligned}$$

is solved at time k for the optimal input sequence

$$U_{k\to k+N|k} = \{u_{k|k}, \dots, u_{k+N-1|k}\}$$

State Indices

The state $x_{k+i|k}$ is the predicted state at time $k+i$, computed by starting from the current state $x_{k|k} = x(k)$ and applying the input sequence $u_{k|k}, \dots, u_{k+i-1|k}$ to the system model

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

For example, the states $x_{3|1}$ and $x_{3|2}$ live both at time 3, but are computed at different points in time.

Receding Horizon Control Law

The receding horizon control law is given by the first element of $U^*_{k\to k+N|k}$:

$$\kappa_k(x(k)) = u(k) = u^*_{k|k}(x(k))$$

10.1.2 Relative Time Notation

System, constraints, and cost function are time-invariant. Hence, the solution $\kappa_k(x(k))$ is a time-invariant function of the IC $x(k)$, allowing for the simplified notation

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

subj. to

$$\begin{aligned} x_{i+1} &= Ax_i + Bu_i, \quad i = 0, \dots, N-1, \\ x_i &\in \mathcal{X}, \quad u_i \in U, \quad i = 0, \dots, N-1, \\ x_N &\in \mathcal{X}_f, \\ x_0 &= x(k), \end{aligned}$$

where $U = \{u_0, \dots, u_{N-1}\}$.

Receding Horizon Control Law

The control law and closed-loop system are time-invariant as well, and we write $\kappa(x(k))$ for $\kappa_k(x(k))$.

1 System Theory

1.1 Discretization

Euler Discretization (Forward)

$$\dot{x}^c \approx \frac{x^c(t+T_s) - x^c(t)}{T_s}$$

with T_s describing the sampling time, hence

$$x(k) := x^c(t_0 + kT_s)$$
$$u(k) := u^c(t_0 + kT_s)$$

Therefore a LTI system becomes

$$\begin{aligned} x(k+1) &= (\mathbb{I} + T_s A^c)x(k) + T_s B^c u(k) &&= Ax(k) + Bu(k) \\ y(k) &= C^c x(k) + D^c u(k) &&= Cx(k) + Du(k) \end{aligned}$$

Exact Discretization of LTI (ZOH)

$$x(t_{k+1}) = \underbrace{e^{A^c T_s}}_{=A} x(t_k) + \underbrace{\int_0^{T_s} e^{A^c(T_s - \tau)} B^c d\tau}_{B}\, u(t_k)$$

$$B = (A^c)^{-1}(A - \mathbb{I})B^c$$

if A^c is invertible.

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

Solution of DT LTI System

$$x(k+N) = A^N x(k) + \sum_{i=0}^{N-1} A^i B u(k+N-1-i)$$

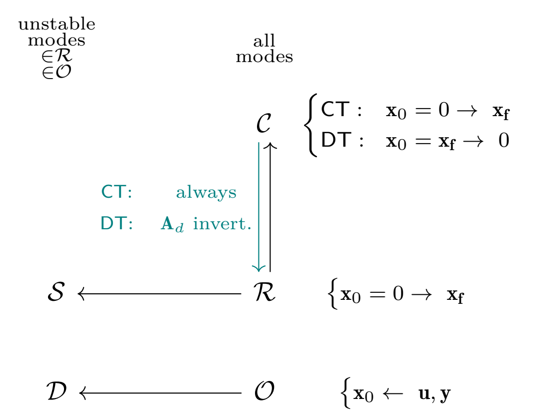
1.2 LTI DT System Analysis

1.2.1 Similarity Transform

$$\begin{aligned} \tilde{x}(k) &= Tx(k); && \det(T) \neq 0 \\ \tilde{x}(k+1) &= TAT^{-1}\tilde{x}(k) + TBu(k) &&= \tilde{A}\tilde{x}(k) + \tilde{B}u(k) \\ y(k) &= CT^{-1}\tilde{x}(k) + Du(k) &&= \tilde{C}\tilde{x}(k) + \tilde{D}u(k) \end{aligned}$$

1.2.2 Controllability and Observability

$$\mathcal{R} \subseteq \mathcal{C}, \quad \mathcal{R} \subset \mathcal{S}, \quad \mathcal{O} \subset \mathcal{D}$$



1.2.2.1 Controllability

Controllable iff

$$\mathcal{C} := \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}$$

has **full row rank**.

Stabilizability
iff all uncontrollable modes are stable, i.e.
- A system is stabilizable if all unstable modes are controllable.
- Controllability always implies stabilizability

PBH test:
$\operatorname{rank}([\lambda_j I - A \mid B]) = n \quad \forall \lambda_j \in \Lambda_A^+ \Rightarrow (A, B)$ is stabilizable

where Λ_A^+ is the set of unstable eigenvalues.

1.2.2.2 Observability

Observable iff

$$\mathcal{O} := \begin{bmatrix} C^\top & (CA)^\top & (CA^2)^\top & \cdots & (CA^{n-1})^\top \end{bmatrix}^\top$$

has **full column rank**.

Detectability
iff all unobservable modes are stable
- A system is detectable if all unstable modes are observable.
- Observability always implies detectability.

PBH test:
$\operatorname{rank}\left(\left[A^\top - \lambda_j I | C^\top\right]\right) = n \; \forall \lambda_j \in \Lambda_A^+ \Rightarrow (A, C)$ is detectable

1.3 Nonlinear System Analysis

1.3.1 Lyapunov Stability

The equilibrium point $\bar{x}$ of a DT system is **Lyapunov stable** iff for every $\epsilon > 0$ there exists a $\delta(\epsilon)$ such that

$$\|x(0) - \bar{x}\| < \delta(\epsilon) \rightarrow \|x(k) - \bar{x}\| < \epsilon, \; \forall k \geq 0$$

Global Asymptotic Stability
An equilibrium point $\bar{x}$ of a system is globally asymptotically stable if it is Lyapunov stable and attractive

$$\lim_{k \to \infty} \|x(k) - \bar{x}\| = 0, \; \forall x(0) \in \mathbb{R}$$

1.3.2 Global Lyapunov Function

A function $V : \mathbb{R}^n \to \mathbb{R}$, continuous at the origin, finite $\forall x \in \mathbb{R}^n$ that satisfies

$$\begin{aligned} \|x\| \to \infty \Rightarrow V(x) &\to \infty && \text{(radially unbounded)} \\ V(0) = 0 \text{ and } V(x) &> 0 && \forall x \in \mathbb{R}^n \setminus \{0\} \\ V(g(x)) - V(x) &\leq -\alpha(x) && \forall x \in \mathbb{R}^n, \alpha \succ 0 : \mathbb{R}^n \to \mathbb{R} \end{aligned}$$

is called a **Lyapunov function**.

1.3.3 Local Lyapunov Function

If the conditions on asymptotical stability only hold on a closed and bounded, positively invariant set $\Omega \subset \mathbb{R}^n$ and V is finite $\forall x \in \Omega$, then the system is locally asymptotically stable in Ω, where Ω is also called the **region of attraction (ROA)**.

1.3.4 Global Lyapunov Stability

If a system admits a Lyapunov function $V(x)$, then $x = 0$ is **globally asymptotically stable**.

If $V(x)$ satisfies the conditions only with $\alpha(x)$ being positive **semi**definite, then $x = 0$ is **globally Lyapunov stable**.

1.4 Global Lyapunov Stability of LTI DT System

For a DT LTI system one can construct a Lyapunov function by choosing

$$V(x) = x^\top P x, \quad \alpha(x) = x^\top Q x, \; Q \succ 0$$

where $P \succ 0$ is the solution of the DT Lyapunov equation

$$A^\top P A - P = -Q, \quad Q \succ 0$$

which as a unique solution if A is stable, i.e. all eigenvalues inside the unit circle.

Closed Loop Control

$$(A + BK)^\top P(A + BK) - P = -Q, \quad Q \succ 0$$

Note that Q is not the stage cost matrix!

1.4.1 Direct and Indirect Lyapunov Methods

1.4.1.1 Direct Lyapunov Method

The direct method involves finding a Lyapunov function $V(x)$ and proving that it satisfies the conditions for stability directly.

1.4.1.2 Indirect Lyapunov Method

The indirect method uses the linearization and Hartman-Grobman theorem to conclude stability of the nonlinear system from the stability of the linearized system in a local neighborhood of the equilibrium point.

2 Unconstrained Linear Quadratic Optimal Control

2.1 Finite Horizon LQR

$$J^*(x(0)) = \min_U x_N^\top P x_N + \sum_{i=0}^{N-1} (x_i^\top Q x_i + u_i^\top R u_i)$$
$$\text{subj. to } x_{i+1} = Ax_i + Bu_i, \quad i = 0, \ldots, N-1$$
$$x_0 = x(0)$$

2.1.1 Comparison: Batch vs. Recursive Approach

Batch Approach
- · open-loop control sequence
- + Allows for constraints
- - large matrix inversions

Recursive Approach
- + feedback policy → disturbance robustness

2.1.2 Batch Approach

Problem Setup

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

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

$$\overline{Q} := \operatorname{blockdiag}(Q, \ldots, Q, P), \; \overline{R} := \operatorname{blockdiag}(R, \ldots, R)$$

$$J(x(0), U) = U^\top H U + 2x(0)^\top F U + x(0)^\top {S^x}^\top \overline{Q} S^x x(0)$$
$$H := (S^u)^\top \overline{Q} S^u + \overline{R} \succ 0, \qquad F := (S^x)^\top \overline{Q} S^u$$

Optimal Control Solution
Setting the gradient to zero yields the optimal control action

$$U^*(x(0)) = -H^{-1} F^\top x(0)$$

Note that H^{-1} always exists as $H \succ 0$ i.e. full rank.

$$J^*(x(0)) = x(0)^\top \left((S^x)^\top \overline{Q} S^x - F H^{-1} F^\top\right) x(0)$$

2.1.3 Recursive Approach

2.1.3.1 Bellman's Principle of Optimality

For any $j = 0 \ldots N$

$$J_j^\star(x_j) = \min_{u_j} J(x_i, u_i) + J_{j+1}^\star(x_{j+1})$$
$$\text{subj. to } x_{j+1} = Ax_j + Bu_j$$

2.1.3.2 Optimal Control Method

Problem Setup

$$J_j^\star(x(j)) = \min_{U_{j \to N}} x_N^\top P x_N + \sum_{i=j}^{N-1} (x_i^\top Q x_i + u_i^\top R u_i)$$
$$\text{subj. to } x_{i+1} = Ax_i + Bu_i, i = j, \ldots, N-1$$
$$x_j = x(j)$$

Optimal Control Solution

$$u_i^\star(x_i) = F_i x_i$$
$$F_i = -(B^\top P_{i+1} B + R)^{-1} B^\top P_{i+1} A$$

The optimal cost-to-go is

$$J_i^\star(x_i) = x(i)^\top P_i x(i)$$

with cost matrices found from the Discrete Time Riccati equation / **Riccati Difference equation (RDE)**:

$$P_i = A^\top P_{i+1} A + Q - A^\top P_{i+1} B (B^\top P_{i+1} B + R)^{-1} B^\top P_{i+1} A$$
$$P_N = P \text{ (init. with terminal weight)}$$

and the full-trajectory **optimal cost** $J^*(x(0))$ is

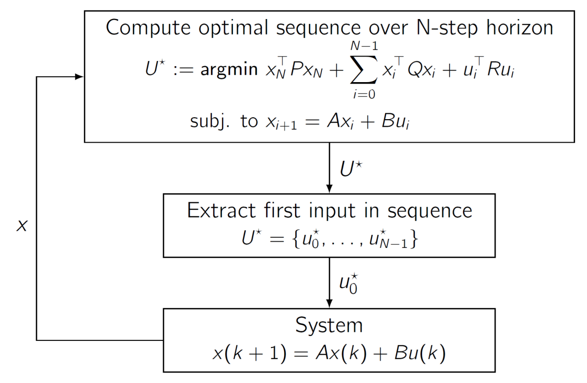
$$x(0)^\top P_0 x(0)$$

2.1.4 Stability of Finite Horizon LQR

There are **three common choices** for the **terminal weight** P in finite horizon LQR to ensure CL stability.

1. Choose P s.t. it matches the infinite horizon LQR solution
$$P = A^\top P A + Q - A^\top P B (B^\top P B + R)^{-1} B^\top P A$$
2. Choose P assuming no control action after the end of the finite horizon
$$x(k+1) = Ax(k), k = N, \ldots, \infty$$
$$A^\top P A + Q = P \text{ (solve LE for } P)$$
This approach only makes sense if the system is asymptotically stable (or no positive definite solution P will exist).
3. Force both state and input both to be zero after the end of the finite horizon. No P is needed but the constraint
$$x_{i+N} = 0$$

2.2 Receding Horizon LQR


Compute optimal sequence over N-step horizon
U* := argmin x_N^⊤ P x_N + Σ_{i=0}^{N−1} x_i^⊤ Q x_i + u_i^⊤ R u_i
subj. to x_{i+1} = Ax_i + Bu_i
U*
Extract first input in sequence
U* = {u_0*, ..., u_{N−1}*}
u_0*
System
x(k+1) = Ax(k) + Bu(k)
x

2.2.1 Stability of Receding Horizon LQR

Depending on the specific situation
- receding horizon *can* result in a stable CL trajectory with a horizon length that would yield an unstable trajectory in the OL case. However, receding horizon LQR does **not** guarantee a stable CL in general.
- OL control can be sufficient to achieve stability, i.e. receding horizon is not always needed in theory.

2.3 Infinite Horizon LQR

Problem Setup

$$J_\infty(x(0)) = \min_{u(\cdot)} \sum_{i=0}^{\infty} \left(x_i^\top Q x_i + u_i^\top R u_i\right)$$
$$\text{subj. to } x_{i+1} = Ax_i + Bu_i, \quad i = 0, 1, 2, \ldots, \infty,$$
$$x_0 = x(0)$$

Optimal Control Solution

$$u^\star(k) = F_\infty x(k)$$
$$F_\infty := -(B^\top P_\infty B + R)^{-1} B^\top P_\infty A$$
$$J_\infty(x(k)) = x(k)^\top P_\infty x(k)$$

where P_∞ is the solution of the **Discrete Time Algebraic Riccati equation (DARE)** ($P_i = P_{i+1} = P_\infty$)

$$P_\infty = A^\top P_\infty A + Q - A^\top P_\infty B (B^\top P_\infty B + R)^{-1} B^\top P_\infty A$$

Stability
Assuming
- (A, B) stabilizable
- $(Q^{\frac{1}{2}}, A)$ detectable,

then the RDE (initialized with Q at $i = \infty$ and solved for $i \searrow 0$) converges to the **unique positive definite** solution P_∞.

Then $J^\star(x) = x^T P_\infty x = V(x)$ is a Lyapunov function for the CL system $x^+ = (A + BF_\infty)x$, i.e. the system is asymptotically stable.

3 Convex Optimization

3.1 Problem Formulation and Terminology

Primal Problem

$$\begin{aligned} \min_{x \in \operatorname{dom}(f)} \; & f(x) \\ \text{subj. to } \; & g_i(x) \leq 0 \quad i = 1, \ldots, m \\ & h_i(x) = 0 \quad i = 1, \ldots, p \end{aligned}$$

3.1.1 Constraints

$g_i(x)$ is **active** at $\bar{x}$ if $g_i(\bar{x}) = 0$. Thus, equality constraints are always active.

A **redundant** constraint does not change the feasible set.

3.1.2 Optimality

$x \in \mathcal{X}$ is **locally optimal** if, for some $R > 0$

$$y \in \mathcal{X}, \; \|x - y\| < R \quad \Rightarrow \quad f(x) \leq f(y)$$

$x \in \mathcal{X}$ is **globally optimal** if

$$f(x) \leq f(y) \quad \forall y \in \mathcal{X}$$

3.2 Convex Sets

A set $\mathcal{X}$ is convex **iff** for any pair of points x and y in $\mathcal{X}$:

$$\lambda x + (1 - \lambda) y \in \mathcal{X}, \quad \forall \lambda \in [0, 1], \; \forall x, y \in \mathcal{X}$$

3.2.1 Hyperplanes and Halfspaces

A **hyperplane** is defined by $\{x \in \mathbb{R}^n \mid a^\top x = b\}$ for $a \neq 0$, where $a \in \mathbb{R}^n$ is the normal vector to the hyperplane.

A **halfspace** is everything on one side of a hyperplane $\{x \in \mathbb{R}^n \mid a^\top x \leq b\}$ for $a \neq 0$. It can either be open (strict inequality) or closed (non-strict inequality).

3.2.2 Polyhedra and Polytopes

Polyhedra
A polyhedron is the intersection of a finite number of closed halfspaces:

$$P := \{x \mid a_i^\top x \leq b_i, i = 1, \ldots, m\} = \{x \mid Ax \leq b\}$$

$$A := \begin{bmatrix} a_1 & a_2 & \ldots & a_m \end{bmatrix}^\top \quad \text{and} \quad b := \begin{bmatrix} b_1 & b_2 & \ldots & b_m \end{bmatrix}^\top$$

Polytopes
A polytope is a bounded polyhedron.

Minkowski-Weyl Theorem
For a set $P \subseteq \mathbb{R}^d$, the following are equivalent:
- P is a polytope $\{x \mid Ax \leq b\}$ (bounded)
- P is finitely generated, i.e., these exist a finite set of vectors $\{v_i\}$ such that $P = \operatorname{co}(\{v_1, \ldots, v_s\})$

3.2.3 Ellipsoids

$$\{x \mid (x - x_c)^\top A^{-1} (x - x_c) \leq 1\}, \quad A \succ 0$$

3.2.4 Norm Balls

The norm ball, defined by $\{x| \; ||x - x_c|| \leq r\}$ where x_c is the centre of the ball and $r \geq 0$ is the radius, is always convex for any norm.

Euclidean Ball
The Euclidean ball $B(x_c, r)$ is a special case of the ellipsoid, for which $A = r^2 \mathbb{I}$, so that $B(x_c, r) := \{x | \|x - x_c\|_2 \leq r\}$.

3.2.5 Set Operations

The intersection of convex sets is convex, whereas the union of convex sets is not necessarily convex.

3.2.6 Convex Hull

For any subset S of $\mathbb{R}^d$, the *convex hull* $\operatorname{co}(S)$ is the intersection of all convex sets containing S, i.e. the smallest convex set containing S.

3.3 Convex Functions

A function $f : \operatorname{dom}(f) \to \mathbb{R}$ is convex **iff** $\operatorname{dom}(f)$ is convex and $\forall x, y \in \operatorname{dom}(f)$

$$f(\lambda x + (1 - \lambda) y) \leq \lambda f(x) + (1 - \lambda) f(y), \quad \forall \lambda \in (0, 1)$$

The function $f : \operatorname{dom}(f) \to \mathbb{R}$ is strictly convex if this inequality is strict.

The function f is **concave iff** $\operatorname{dom}(f)$ is convex and $-f$ is convex.

First Order Condition

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x), \quad \forall x, y \in \operatorname{dom}(f)$$

Second Order Condition

$$\nabla^2 f(x) \geq 0, \quad \forall x \in \operatorname{dom}(f), \quad \nabla^2 f(x)_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$$

3.3.1 Level and Sublevel Sets

Level Sets

$$L_\alpha := \{x | x \in \operatorname{dom}(f), \; f(x) = \alpha\}$$

Sublevel Sets

$$C_\alpha := \{x | x \in \operatorname{dom}(f), \; f(x) \leq \alpha\}$$

f is convex $\Rightarrow$ sublevel sets of f are convex $\forall \alpha$.

3.3.2 Examples of Convex Functions

Convex
- Affine functions: $ax + b, \; \forall a, b \in \mathbb{R}$
- Exponential functions: $e^{ax} \; \forall a \in \mathbb{R}$
- Powers: x^α on domain $\mathbb{R}_{++}$ for $\alpha \leq 0, \alpha \geq 1$
- Vector norms on $\mathbb{R}^n$:

$$||x||_p = \left(\sum_{i=1}^n |x|^p\right)^{\frac{1}{p}}, \text{ for } p \geq 1$$
$$||x||_\infty = \max_i |x_i|$$

Convexity Preserving Operations
- Nonnegative weighted sums: $\sum_{i=1}^n \theta_i f_i(x)$ for $\theta_i \geq 0$
- Composition with an affine function: $f(ax + b)$
- Pointwise maximum/supremum: $\max(f_1(x), f_2(x))$
- Partial minimization

Concave
- Affine functions: $ax + b$
- Powers: x^α on domain $\mathbb{R}_{++}$ for $0 \leq \alpha \leq 1$
- Logarithm: $\log(x)$ on domain $\mathbb{R}_{++}$
- Entropy: $-x \log(x)$ on domain $\mathbb{R}_{++}$

3.4 Convex Optimization Problems

$$\begin{aligned} \min_{x \in \operatorname{dom}(f)} \; & f(x) \\ \text{subj. to } \; & g_i(x) \leq 0 \quad i = 1, \ldots, m \\ & a_i^\top x = b_i \quad i = 1, \ldots, p \\ & (Ax = b \quad A \in \mathbb{R}^{p \times n}) \end{aligned}$$

$$\begin{cases} f, g_i & \text{convex functions} \\ \operatorname{dom}(f) & \text{convex set} \\ h_i(x) = a_i^\top x - b & \text{affine functions} \end{cases}$$

As a result, the feasible set $\mathcal{X}$ is convex.

3.4.1 Equivalent Optimization Problems

Slack Variables

$$\begin{aligned} & \min f(x) \\ & \text{s.t. } A_i x \leq b_i, && i = 1, \ldots, m \end{aligned}$$

is equivalent to

$$\begin{aligned} & \min f(x) \\ & \text{s.t. } A_i x + s_i = b_i, && i = 1, \ldots, m \\ & \quad s_i \geq 0, && i = 1, \ldots, m \end{aligned}$$

3.4.2 Linear Program (LP)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \; & c^\top x \\ \text{s.t. } \; & Gx \leq h \\ & Ax = b \end{aligned}$$

where the feasible set $\mathcal{P}$ is a polyhedron (convex).

For the set of optimizers $\mathcal{X}_{opt}$, generally, three cases can occur:
1. **Unbounded** $\mathcal{P}$ is unbounded below
2. **Bounded with unique optimizer**: $\mathcal{X}_{opt}$ is a **singleton**. *At least* n constraints are active.
3. **Bounded with multiple optimizers**: $\mathcal{X}_{opt}$ is a (bounded or unbounded) subset of $\mathbb{R}^n$. *At least* one constraint is active. In this case, the solution is sensitive to noise.

3.4.3 Quadratic Program (QP)

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \; & \frac{1}{2} x^\top H x + q^\top x + r \\ \text{s.t. } \; & Gx \leq h \\ & Ax = b \end{aligned}$$

is convex if $H > 0$.

Problems with concave objective $H \prec 0$ are quadratic programs, but hard.

If feasible, in general two cases can occur:
1. **Case 1**: optimizer lies strictly inside the feasible polyhedron. No constraints active.
2. **Case 2**: optimizer lies on the boundary of the feasible polyhedron. At least one constraint active.

3.5 Optimality Conditions

3.5.1 Lagrange Dual Problem

The possibly non-convex primal problem can be transformed into a convex dual problem using the **Lagrangian Function**:

$$L : \operatorname{dom}(f) \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$$

$$L(x, \lambda, \nu) = f(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{i=1}^p \nu_i h_i(x), \qquad \lambda_i \geq 0$$

and the **dual function** d which is a lower bound for the primal optimal value p^*:

$$d(\lambda, \nu) = \inf_{x \in \operatorname{dom}(f)} L(x, \lambda, \nu) \leq p^*, \quad \forall (\lambda \geq 0, \nu \in \mathbb{R}^p)$$

Dual Problem

$$\max_{\lambda, \nu} d(\lambda, \nu)$$
$$\text{subject to} \quad \lambda \geq 0$$

- is convex ($\min_{\lambda, \nu} -d(\lambda, \nu)$), even if the primal is not.
- has an optimal value $d^* \leq p^*$.
- the point (λ, ν) is **dual feasible** if $\lambda \geq 0$ and $(\lambda, \nu) \in \operatorname{dom}(d)$.
- Can often impose the constraint $(\lambda, \nu) \in \operatorname{dom}(d)$ explicitly.

3.5.1.1 Dual of a Linear Program (LP)

$$d(\lambda, \nu) = \begin{cases} -b^\top \nu - e^\top \lambda & \text{if } A^\top \nu + C^\top \lambda + c = 0 \\ -\infty & \text{otherwise} \end{cases}$$

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & c^\top x & \qquad \max_{\lambda, \nu} \quad & -b^\top \nu - e^\top \lambda \\ \text{s.t.} \quad & Ax - b = 0 & \text{s.t.} \quad & A^\top \nu + C^\top \lambda + c = 0 \\ & Cx - e \leq 0 & & \lambda \geq 0 \end{aligned}$$

Both are LPs

3.5.1.2 Dual of a Quadratic Program (QP)

$$d(\lambda) = -\frac{1}{2}\left(c + C^\top \lambda\right)^\top Q^{-1} \left(c + C^\top \lambda\right) - e^\top \lambda$$

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \; & \frac{1}{2} x^\top Q x + c^\top x \\ \text{s.t.} \quad & Cx - e \leq 0 \\ & Q \succ 0 \end{aligned}$$

$$\min_\lambda \frac{1}{2} \lambda^\top C Q^{-1} C^\top \lambda + \left(C Q^{-1} c + e\right)^\top + \frac{1}{2} c^\top Q^{-1} c$$
$$\text{s.t.} \quad \lambda \geq 0$$

Both are QPs

3.5.2 Weak and Strong Duality

Duality Gap: $p^* - d^*$
Weak Duality
It is always true that

$$d^* \leq p^*.$$

Strong Duality
It is sometimes true that

$$p^* = d^*.$$

Slater Condition
If the problem is **convex**, then the Slater condition states:

If there exists at least one **strictly feasible point**, i.e.

$$\{x | Ax = b, \; g_i(x) < 0, \; \forall i \in \{1, \ldots, m\}\} \neq \emptyset$$

this **always implies strong duality**.

3.5.3 Karush-Kuhn-Tucker (KKT) Conditions

Assume that f, all g_i and h_i are differentiable.
1. Primal Feasibility:
$$g_i(x^*) \leq 0 \quad i = 1, \ldots, m$$
$$h_i(x^*) = 0 \quad i = 1, \ldots, p$$
2. Dual Feasibility:
$$\lambda^* \geq 0$$
3. Complementary Slackness:
$$\lambda_i^* g_i(x^*) = 0 \quad i = 1, \ldots, m$$
implying
$$\lambda_i^* = 0 \text{ for every } g_i(x^*) < 0.$$
$$g_i(x^*) = 0 \text{ for every } \lambda_i^* > 0$$
4. Stationarity:
$$\nabla_x L(x^*, \lambda^*, \nu^*) = \nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*)$$
$$\ldots + \sum_{i=1}^p \nu_i^* \nabla h_i(x^*) = 0$$

Convex Problems
Iff (x^*, λ^*, ν^*) satisfy the KKT conditions, then they are optimal and Slater's condition holds, hence strong duality holds

$$p^* = d^*.$$

General Problems
For a general optimization problem there is only a necessary condition that states:
If x^* and (λ^*, ν^*) are primal and dual optimal solutions and strong duality holds, then x^* and (λ^*, ν^*) satisfy the KKT conditions.

3.5.4 Sensitivity Analysis

$$\begin{aligned} \min_x \quad & f(x) & \max_{\lambda,\nu} \quad & d(\lambda,\nu) - u^\top\lambda - v^\top\nu \\ \text{s.t.} \quad & g_i(x) \le u_i & \text{s.t.} \quad & \lambda \ge 0 \\ & h_i(x) = v_i & & \end{aligned}$$

where u and v represent the perturbation.

Global Sensitivity Analysis
In the case where strong duality holds for the unperturbed problem, the weak duality of the perturbed problem implies

$$\begin{aligned} p^*(u,v) &\ge d^*(\lambda^*,\nu^*) - u^\top\lambda^* - v^\top\nu^* \\ &= p^*(0,0) - u^\top\lambda^* - v^\top\nu^* \end{aligned}$$

Local Sensitivity Analysis
If in addition $p^*(u,v)$ is differentiable at $(0,0)$, then

$$\lambda_i^* = -\frac{\partial p^*(0,0)}{\partial u_i}, \qquad \nu_i^* = -\frac{\partial p^*(0,0)}{\partial v_i},$$

4 Constrained Finite Time Optimal Control (CFTOC)

4.1 Constrained Linear Optimal Control

$$\begin{aligned} J^*(x(k)) = \min_U \; & I_f(x_N) + \sum_{i=0}^{N-1} I(x_i,u_i) \\ \text{subject to} \quad & x_{i+1} = Ax_i + Bu_i, \quad i = 0,\dots,N-1 \\ & x_i \in \mathcal{X}, u_i \in \mathcal{U}, \quad i = 0,\dots,N-1 \\ & x_N \in \mathcal{X}_f \\ & x_0 = x(k) \end{aligned}$$

The feasible set describes a set of initial states for which the optimal control problem is feasible. It is defined only by the constraints and not by the cost:

$$\begin{aligned} \mathcal{X}_N = \{ & x_0 \in \mathbb{R}^n | \exists (u_0,\dots,u_{N-1}) \text{ such that } x_i \in \mathcal{X}, u_i \in \mathcal{U} \\ & i = 0,\dots,N-1, x_N \in \mathcal{X}_f, \text{where } x_{i+1} = Ax_i + Bu_i\} \end{aligned}$$

4.1.1 Quadratic Cost CFTOC

$$\begin{aligned} J^*(x(k)) = \min_U \; & x_N^T P x_N + \sum_{i=0}^{N-1} x_i^\top Q x_i + u_i^\top R u_i \\ \text{subject to} \quad & x_{i+1} = Ax_i + Bu_i, \quad i = 0,\dots,N-1 \\ & x_i \in \mathcal{X}, u_i \in \mathcal{U}, \quad i = 0,\dots,N-1 \\ & x_N \in \mathcal{X}_f \\ & x_0 = x(k) \end{aligned}$$

4.1.1.1 Construction of QP Without Substitution

$$J^*(x(k)) = \min_z \begin{bmatrix} z^\top & x(k) \end{bmatrix} \underbrace{\begin{bmatrix} \bar{H} & 0 \\ 0 & Q \end{bmatrix}}_{\widetilde{H}} \begin{bmatrix} z^\top & x(k)^\top \end{bmatrix}^\top$$

$$\begin{aligned} \text{subject to} \quad & G_{in} z \le w_{in} + E_{in} x(k) \\ & G_{eq} z = E_{eq} x(k) \end{aligned}$$

$$z = \begin{bmatrix} x_1^\top & \cdots & x_N^\top & u_0^\top & \cdots & u_{N-1}^\top \end{bmatrix}^\top \in \mathbb{R}^{n_z := N(n_x+n_u)}$$

Cost

$$\bar{H} = \left[\begin{array}{c|c} \text{blkdiag}(Q,\dots,Q,P) & 0 \\ \hline 0 & \text{blkdiag}(R,\dots,R) \end{array}\right]$$

Equality Constraints

$$G_{eq} = \left[\begin{array}{cccc|cccc} I & & & & -B & & & \\ -A & I & & & & -B & & \\ & & \ddots & & & & \ddots & \\ & & -A & I & & & & -B \end{array}\right] \in \mathbb{R}^{N\cdot n_x \times n_z}$$

$$E_{eq} = \begin{bmatrix} A^\top & 0 & \cdots & 0 \end{bmatrix}^\top \in \mathbb{R}^{N\cdot n_x \times n_x}$$

Inequality Constraints

$$G_{in} \in \mathbb{R}^{(n_{in,x}+n_{in,u})\times n_z}$$

$$= \left[\begin{array}{c|c} 0 & 0 \\ \hline \text{blkdiag}(A_x,\dots,A_x,A_f) & 0 \\ \hline 0 & \text{blkdiag}(A_u,\dots,A_u) \end{array}\right]$$

$$w_{in} \in \mathbb{R}^{(n_{in,x}+n_{in,u})\times 1}, \quad E_{in} \in \mathbb{R}^{(n_{in,x}+n_{in,u})\times n_x}$$

$$\begin{aligned} w_{in} &= \begin{bmatrix} b_x{}^\top & |b_x{}^\top & \cdots & b_x{}^\top & b_f{}^\top & |b_u{}^\top & \cdots & b_u{}^\top \end{bmatrix}^\top \\ E_{in} &= \begin{bmatrix} -A_x{}^\top & |0 & \cdots & 0 & 0 & |0 & \cdots & 0 \end{bmatrix} \end{aligned}$$

4.1.1.2 Construction of QP With Substitution

See Subsection 2.1.2

$$X = S^x x(k) + S^u U$$

Cost

$$J^*(x(k),U) = \min_U \begin{bmatrix} U^\top & x(k)^\top \end{bmatrix} \underbrace{\begin{bmatrix} H & F^\top \\ F & Y \end{bmatrix}}_{\widetilde{H}} \begin{bmatrix} U^\top & x(k)^\top \end{bmatrix}^\top$$

$$\text{subject to} \quad GU \le w + Ex(k)$$

Inequality Constraints

$$G \in \mathbb{R}^{(n_{in,u}+n_{in,x})\times(N\cdot n_u)}$$

$$= \left[\begin{array}{cccc} A_u & 0 & \cdots & 0 \\ 0 & A_u & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_u \\ \hline 0 & 0 & \cdots & 0 \\ A_x B & 0 & \cdots & 0 \\ A_x AB & A_x B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_f A^{N-1}B & A_f A^{N-2}B & \cdots & A_f B \end{array}\right]$$

$$w \in \mathbb{R}^{(n_{in,u}+n_{in,x})\times 1}, \quad E \in \mathbb{R}^{(n_{in,u}+n_{in,x})\times n_x}$$

$$w = \begin{bmatrix} b_u{}^\top & \cdots & b_u{}^\top & |b_x{}^\top & b_x{}^\top & \cdots & b_f{}^\top \end{bmatrix},$$

$$E = \begin{bmatrix} 0^\top & \cdots & 0^\top & |-A_x{}^\top & -A_x A^\top & \cdots & -A_f A^{N\top} \end{bmatrix}$$

4.1.1.3 Comparison: Substitution vs. No Substitution

	Substitution	No Substitution
# opt. vars.	Nn_u	$N(n_x+n_u)$
benefits	less opt. vars.	
	less constraints	sparse constr. ($\propto N$)
drawbacks	complicated constr.	more opt. vars.
	more constr. ($\propto N^2$)	

Note that substitution transforms input constraints into state constraints, making them more involved in general. Hence, no substitution is often preferable. For small N and large n_x however, substitution can be more efficient.

4.1.1.4 State Feedback Solution

As $n_x >= 1$, the CFTOC problem is a **multiparametric quadratic program (mp-QP)** in general with the following solution properties:

- u_0^* is of the form (nonlinear feedback policy)

  $$u_0^* = \kappa(x(k)), \quad \forall x(k) \in \mathcal{X}_0$$

  with $\kappa : \mathbb{R}^n \to \mathbb{R}^m$ cont., **piecewise affine** on polyhedra

  $$\kappa = \kappa(x) = F^j x + g^j, \quad \text{if} \quad x \in CR^j, \quad j = 1,\cdots,N^r$$

- The polyhedral sets for the individual control laws

  $$CR^j = \{x \in \mathbb{R}^n | H^j x \le K^j\}, j = 1,\cdots,N^r$$

  are a partition of the feasible polyhedron $\mathcal{X}_0$.
- $J^*(x(k))$ is **convex**, **piecewise quadratic** on polyhedra.

Explicit MPC addresses how to compute this solution.

4.1.2 1-Norm and ∞-Norm Cost CFTOC

4.1.2.1 l_∞ Minimization

$$\min_{x\in\mathbb{R}^n} \|x\|_\infty = \min_{x\in\mathbb{R}^n} [\max\{x_1,\dots,x_n,-x_1,\dots,-x_n\}]$$
$$\text{subj. to } Fx \le g$$

Auxiliary Variable Formulation

$$\begin{aligned} & \min_{x,t} t \\ \text{subj. to } & -\mathbf{1}t \le x \le \mathbf{1}t \\ & Fx \le g \end{aligned}$$

Application to CFTOC
Introduces a *scalar* auxiliary variable for each state of the trajectory

$$\begin{aligned} & z := \{\varepsilon_0^x,\dots,\varepsilon_N^x,\varepsilon_0^u,\dots,\varepsilon_{N-1}^u,u_0^\top,\dots,u_{N-1}^\top\} \in \mathbb{R}^s, \\ & s := (m+1)N + N + 1 \end{aligned}$$

$$\begin{aligned} \min_z \quad & \varepsilon_0^x + \cdots + \varepsilon_N^x + \varepsilon_0^u + \cdots + \varepsilon_{N-1}^u \\ \text{subj. to} \quad & -\mathbf{1}_n\varepsilon_{x_i} \le Qx_i \le \mathbf{1}_n\varepsilon_{x_i} \\ & -\mathbf{1}_r\varepsilon_{x_N} \le Px_N \le \mathbf{1}_r\varepsilon_{x_N} \\ & -\mathbf{1}_m\varepsilon_{u_i} \le Ru_i \le \mathbf{1}_m\varepsilon_{u_i} \\ & x_i = A^i x_0 + \sum_{j=0}^{i-1} A^j B u_{i-1-j} \in \mathcal{X} \\ & u_i \in \mathcal{U}, x_N \in \mathcal{X}_f, x_0 = x(k) \end{aligned}$$

4.1.2.2 l_1 Minimization

$$\min_{x\in\mathbb{R}^n} \|x\|_1 = \min_{x\in\mathbb{R}^n} \left[\sum_{i=1}^m \max\{x_i,-x_i\}\right]$$
$$\text{subj. to } Fx \le g$$

Auxiliary Variable Formulation

$$\begin{aligned} \min_{x\in\mathbb{R}^n,\mathbf{t}\in\mathbb{R}^n} \quad & \mathbf{1}^\top\mathbf{t}, \\ \text{subj. to} \quad & -\mathbf{t} \le x \le \mathbf{t}, \\ & Fx \le g \end{aligned}$$

Application to CFTOC

$$\begin{aligned} & z := \{(\boldsymbol{\varepsilon}_0^x)^\top,\dots,(\boldsymbol{\varepsilon}_N^x)^\top,(\boldsymbol{\varepsilon}_0^u)^\top,\dots,(\boldsymbol{\varepsilon}_{N-1}^u)^\top,u_0^\top,\dots,u_{N-1}^\top\} \\ & z \in \mathbb{R}^s \text{ with } s := n(N+1) + 2mN \end{aligned}$$

$$\begin{aligned} \min_z \quad & \mathbf{1}^\top\boldsymbol{\varepsilon}_0^x + \cdots + \mathbf{1}^\top\boldsymbol{\varepsilon}_N^x + \mathbf{1}^\top\boldsymbol{\varepsilon}_0^u + \cdots + \mathbf{1}^\top\boldsymbol{\varepsilon}_{N-1}^u \\ \text{subj. to} \quad & -\boldsymbol{\varepsilon}_{x_i} \le Qx_i \le \boldsymbol{\varepsilon}_{x_i} \\ & -\boldsymbol{\varepsilon}_{x_N} \le Px_N \le \boldsymbol{\varepsilon}_{x_N} \\ & -\boldsymbol{\varepsilon}_{u_i} \le Ru_i \le \boldsymbol{\varepsilon}_{u_i} \end{aligned}$$

General Formulation

$$y = \begin{bmatrix} x \\ \mathbf{z} \end{bmatrix} \in \mathbb{R}^{(n+N)}$$

$$\min_x \|Ax\|_1 \quad \Leftrightarrow \quad \begin{aligned} \min_y \; & \begin{bmatrix} 0 & \mathbf{1}^\top \end{bmatrix} y \\ \text{s.t.} \; & \begin{bmatrix} A & -\mathbb{I} \\ -A & -\mathbb{I} \end{bmatrix} y \le 0 \end{aligned}$$

4.1.2.3 l_1, l_∞ State Feedback Solution

$$\min_{z\in\mathbb{R}^n} c^T z$$
$$\text{subj. to } \bar{G}z \le \bar{w} + \bar{S}x(k)$$

is again a mp-LP with the following solution properties:

- u_0^* has the form:

  $$u_0^* = \kappa(x(0)), \quad \forall x(0) \in \mathcal{X}_0,$$

  where $\kappa : \mathbb{R}^n \to \mathbb{R}^m$ is cont. piecewise affine on polyhedra:

  $$\kappa(x) = F^j x + g^j, \quad \text{if } x \in CR^j, \quad j = 1,\dots,N^r$$

- The polyhedral sets

  $$CR^j = \{x \in \mathbb{R}^n \mid H^j x \le K^j\}, \quad j = 1,\dots,N^r$$

  are a partition of the feasible polyhedron $\mathcal{X}_0$.
- In case of multiple optimizers, a **piecewise affine** control law exists.
- The value function $J^*(x(0))$ is **convex** and **piecewise affine** on polyhedra.

4.2 Common Constraints

4.2.1 Polytopic Constraints

Input Constraints

$$u_{\min} \le u \le u_{\max} \quad \Leftrightarrow \quad \begin{bmatrix} -\mathbb{I} \\ \mathbb{I} \end{bmatrix} u \le \begin{bmatrix} -u_{\min} \\ u_{\max} \end{bmatrix}$$

Rate Constraints

$$\|x_K - x_{k+1}\|_\infty \le \alpha \quad \Leftrightarrow \quad \begin{bmatrix} \mathbb{I} & -\mathbb{I} \\ -\mathbb{I} & \mathbb{I} \end{bmatrix} \begin{bmatrix} x_k \\ x_{k+1} \end{bmatrix} \le \mathbf{1}\alpha$$

Magnitude Constraints

$$\|Cx_k\|_\infty \le \alpha \quad \Leftrightarrow \quad \begin{bmatrix} C \\ -C \end{bmatrix} x_k \le \mathbf{1}\alpha$$

5 Invariance

5.1 Invariance

5.1.1 Invariant Sets

Positively Invariant Set
A set $\mathcal{O}$ is said to be a positively invariant set for an autonomous system if

$$x(k) \in \mathcal{O} \Rightarrow x(k+1) \in \mathcal{O}, \quad \forall k \in \{0,1,\dots\}$$

Maximal Positively Invariant Set $\mathcal{O}_\infty$
The set $\mathcal{O}_\infty \subset \mathcal{X}$ is the maximal positively invariant set with respect to $\mathcal{X}$ if $\mathcal{O}_\infty$ is positively invariant and $\mathcal{O}_\infty$ contains all positively invariant sets.

Geometric Condition for Invariance
A set $\mathcal{O}$ is a positively invariant set if and only if

$$\mathcal{O} \subseteq \text{pre}(\mathcal{O})$$

$$\mathcal{O} \subseteq \text{pre}(\mathcal{O}) \iff \text{pre}(\mathcal{O}) \cap \mathcal{O} = \mathcal{O}$$

5.1.2 Pre-Sets

Given a set S and the dynamic system $x(k+1) = g(x(k))$, the pre-set of S is the set of states that evolve into the target set S in **one** time step:

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

Linear Autonomous Systems

$$\text{pre}(S) = \{x \mid FAx \le f\}$$

5.1.3 Computing Invariant Sets

```
 Input: $g$, $X$
 Output: $\mathcal{O}_\infty$
 $\Omega_0 \leftarrow X$
while true do
    $\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$
    if $\Omega_{i+1} = \Omega_i$ then
        return $\mathcal{O}_\infty = \Omega_i$
```

5.2 Control Invariance

5.2.1 Control Invariant Sets

A set $\mathcal{C} \subseteq \mathcal{X}$ is said to be a control invariant set if

$$x(k) \in \mathcal{C} \Rightarrow \exists u(k) \in \mathcal{U}$$
$$\text{such that } g(x(k),u(k)) \in \mathcal{C} \text{ for all } k \in \mathbb{N}^+$$

Maximal Control Invariant Set $\mathcal{C}_\infty$
The set $\mathcal{C}_\infty$ is said to be the maximal control invariant set for the system $x(k+1) = g(x(k),u(k))$ subject to the constraints $(x,u) \in \mathcal{X}\times\mathcal{U}$ if it is control invariant and **contains all control invariant sets** contained in $\mathcal{X}$.

5.2.2 Conceptual Calculation of Control Invariant Sets

The concept of a pre-set extends to systems with exogenous inputs:

$$\text{pre}(S) := \{x \mid \exists u \in \mathcal{U} \text{ s.t. } g(x,u) \in S\}$$

A set $\mathcal{C}$ is a control invariant set if and only if $\mathcal{C} \subseteq \text{pre}(\mathcal{C})$.

Pre-set Computation for Constrained LTI System

$$\begin{aligned} \text{pre}(S) &= \{x \mid \exists u \in \mathcal{U}, \; Ax + Bu \in S\} \\ &= \{x \mid \exists u \in \mathcal{U}, \; FAx + FBu \le f\} \\ &= \left\{x \mid \exists u, \begin{bmatrix} FA & FB \\ 0 & G \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \le \begin{bmatrix} f \\ g \end{bmatrix}\right\} \end{aligned}$$

5.2.2.1 Usefulness

MPC implicitly describes a (suboptimal) control invariant set such that it's easy to represent and compute.

5.2.2.2 Control Laws from Control Invariant Sets

We can synthesize a control law from a control invariant set by solving the following **optimization problem**:

$$\kappa(x) := \arg\min\{f(x,u) \mid g(x,u) \in \mathcal{C}, \; u \in \mathcal{U}\},$$

that will satisfy constraints (but not necessarily converges).

5.3 Computation of Simple Invariant Sets

5.3.1 Polytopes

5.3.1.1 Intersection

$$S \cap T = \left\{x \middle| \begin{bmatrix} C \\ D \end{bmatrix} x \le \begin{bmatrix} c \\ d \end{bmatrix}\right\}$$

5.3.1.2 Equality Test (Subset Test)

$P = \{x \mid Cx \le c\}$ is a subset of $Q = \{x \mid Dx \le d\}$ if for each row, the **support function** is a subset of D:

$$h_P(D_i) \le d_i$$

where the support (extremum of P in direction D_i) is

$$\begin{aligned} h_p(D_i) := & \max_x D_i x \\ & \text{s.t. } Cx \le c \end{aligned}$$

5.3.2 Ellipsoids

If $V : \mathbb{R}^d \to \mathbb{R}$ is a Lyapunov function for the system $x(k+1) = g(x(k))$, then the sublevel set

$$Y = \{x \mid V(x) \le \alpha\}$$

is an invariant set for all $\alpha \ge 0$.

5.3.2.1 Linear Systems

For linear systems $x(k+1) = Ax(k)$, the Lyapunov function $V(x) = x^\top Px$ (with $P \succ 0$ and $A^\top PA - P \prec 0$) yields the ellipsoidal invariant set

$$Y_\alpha = \{x \mid x^\top Px \le \alpha\} \subset \mathcal{X} = \{x \mid Fx \le f\}$$

Maxmium Invariant Set
If we want to find the largest such Y_α, we must solve

$$\begin{aligned} & \max_\alpha \alpha \\ \text{s.t. } & h_{Y_\alpha}(F_i) \le f_i \quad \forall i \in \{1,...,n\} \\ & h_{Y_\alpha}(F_i) = \|P^{-1/2}F_i^T\|\sqrt{\alpha} \\ & \alpha^* = \min_{i\in\{1,\dots,n\}} \frac{f_i^2}{F_i P^{-1} F_i^\top} \end{aligned}$$

6 Feasibility and Stability

6.1 Zero Terminal Constraint

6.1.1 Feasibility

Assume feasibility for $x(k)$ with optimal solution

$$\{u_0^*,u_1^*,\dots,u_{N-1}^*\}, \quad \{x(k),x_1^*,x_2^*,\dots,x_N^*\}$$

Applying the first control input u_0^* to the system, the state will evolve as

$$x(k+1) = Ax(k) + Bu(k) = Ax(k) + Bu_0^* = x_1^*$$

it follows directly that the sequence

$$\widetilde{U} = \{u_1^*,\dots,u_{N-1}^*,0\} \;\to\; \widetilde{X} = \{x_1^*,\dots,x_N^*,\underbrace{Ax_N^* + Bu_N}_{0}\}$$

is feasible too. And therefore, recursive feasibility is guaranteed and the feasible set is **control invariant**.

6.1.2 Stability

$$\begin{aligned} J^*(x(k+1)) &\le \widetilde{J}(x(k+1)) \\ &= \sum_{i=1}^{N-1} I(x_i^*,u_i^*) + I(x_N^*,0) \\ &= \underbrace{\sum_{i=0}^{N-1} I(x_i^*,u_i^*)}_{J^*(x(k))} - \underbrace{I(x_0^*,u_0^*)}_{\ge 0} + \underbrace{I(x_N^*,0)}_{0} \\ &\le J^*(x(k)) \end{aligned}$$

6.2 Terminal Subset Constraint

6.2.1 Stability and Recursive Feasibility: Main Result

S1.1 Stage cost is positive definite, i.e. it is strictly positive and only zero at the origin.

S1.2 The terminal set $\mathcal{X}_f$ is **invariant** under the local control law $\kappa_f(x_i)$:

$$x_{i+1} = Ax_i + B\kappa_f(x_i) \in \mathcal{X}_f, \quad \forall\, x_i \in \mathcal{X}_f$$

All state and input **constraints are satisfied** in $\mathcal{X}_f$:

$$\mathcal{X}_f \subseteq \mathcal{X}, \; \kappa_f(x_i) \in \mathcal{U}, \quad \forall\, x_i \in \mathcal{X}_f$$

S1.3 The terminal cost is a continuous **Lyapunov function** in the terminal set $\mathcal{X}_f$ and satisfies:

$$I_f(x_{i+1}) - I_f(x_i) \le -I(x_i,\kappa_f(x_i)), \quad \forall\, x_i \in \mathcal{X}_f$$

where 1. & 2. ensure recursive feasibility, 3. ensures stability.

Then the closed-loop system under the MPC control law $u_0^*(x)$ is asymptotically stable and the set $\mathcal{X}_N$ (region of attraction) is positive invariant (and equal to the feasible set) for the system.

6.2.2 Choice of Terminal Sets and Cost

1. Design a local control law κ_f around $x = 0$.
2. Determine the terminal cost to stay in that set under the local control law.
3. Compute the maximum invariant set for the closed loop system under the local control law hence $\mathcal{X}_f$.

6.2.2.1 Linear System, Quadratic Cost

- Choose $P = P_\infty$ as the solution of the ARE.
- Choose the terminal set to be the maximum invariant set for the closed loop dynamics $(A + BF_\infty)$ including state constraints:

  $$\mathcal{X}_{cl} := \left\{x \middle| \begin{bmatrix} A_x \\ A_u F_\infty \end{bmatrix} x \le \begin{bmatrix} b_x \\ b_u \end{bmatrix}\right\}$$

6.2.2.2 Horizon Length Versus Feasible Set

For an MPC **without** terminal constraint, increasing N shrinks the feasible set, as more constraints are added. One has

$$\mathcal{X}_{N+j} \subseteq \mathcal{X}_N$$

For an MPC **with** terminal constraint, with larger N, the feasible set and region of attraction **grow**. The region of attraction ultimately approaches the maximum control invariant set. One has

$$\mathcal{X}_N \subseteq \mathcal{X}_{N+j}$$

Practical Note
In practice, one can enlarge N and check stability by sampling.

7 Practical MPC

7.1 Reference Tracking

The following concepts apply to piecewise constant references and not to general time-varying ones.

7.1.1 Steady-State Target Problem

$$\begin{aligned} x_s &= Ax_s + Bu_s \\ Hx_s &= r \end{aligned} \quad \Leftrightarrow \quad \underbrace{\begin{bmatrix} I-A & -B \\ H & 0 \end{bmatrix}}_{(n_x+n_r)\times(n_x+n_u)} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix}$$

Multiple Solutions
Choose the cheapest (projection):

$$\begin{aligned} \min \quad & u_s^\top R_s u_s \\ \text{s.t.} \quad & \begin{bmatrix} I-A & -B \\ H & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix} \\ & x_s \in \mathcal{X}, \quad u_s \in \mathcal{U}. \end{aligned}$$

No Solution
Compute the unique reachable set point that is closest to r:

$$\begin{aligned} \min \quad & (Hx_s - r)^\top Q_s (Hx_s - r) \\ \text{s.t.} \quad & x_s = Ax_s + Bu_s \\ & x_s \in \mathcal{X}, \quad u_s \in \mathcal{U}. \end{aligned}$$

7.1.2 Delta-Formulation for Tracking

$$\begin{aligned} \Delta U^* = \arg\min_{\Delta U} \quad & \sum_{i=0}^{N-1} \Delta x_i^\top Q \Delta x_i + \Delta u_i^\top R \Delta u_i + V_f(\Delta x_N) \\ \text{s.t.} \quad & \Delta x_0 = \Delta x(k) = x(k) - x_s \\ & \Delta x_{i+1} = A\Delta x_i + B\Delta u_i \\ & G_x \Delta x_i \le h_x - G_x x_s \\ & G_u \Delta u_i \le h_u - G_u u_s \\ & \Delta x_N \in \mathcal{X}_f \end{aligned}$$

Control Law

$$u_0^* = \Delta u_0^* + u_s.$$

Note that if the target steady-state is uniquely defined by the reference, we can also include the target condition as a constraint in the MPC problem.

7.1.3 Convergence

Lyapunov Function
Identical dynamics $\to$ use the same Lyapunov function.

Terminal Set
The closed-loop system converges to the target if in addition to $\Delta x_N \in \mathcal{X}_f$

$$x_s \oplus \mathcal{X}_f \subseteq \mathcal{X}, \quad K\Delta x + u_s \in \mathcal{U}, \quad \forall \Delta x \in \mathcal{X}_f$$

7.1.4 Terminal Set

Terminal Set Scaling
To enlarge the set of feasible targets one can scale (preserving invariance) the terminal set to increase the set of feasible targets:

$$\mathcal{X}_f^{\text{scaled}} = \alpha\mathcal{X}_f$$

7.1.5 Offset-Free Reference Tracking

Algorithm
At each sampling time:

1. Estimate state and disturbance $\widehat{x}$, $\widehat{d}$
2. Solve steady-state target problem using $\widehat{d}$
3. Solve MPC problem:

$$\begin{aligned} \min_U \; & \sum_{i=0}^{N-1} (x_i - x_s)^T Q (x_i - x_s) \\ & + (u_i - u_s)^T R (u_i - u_s) + V_f(x_N - x_s) \\ \text{s.t. } & x_0 = \widehat{x}(k), \quad d_0 = \widehat{d}(k) \\ & x_{i+1} = Ax_i + Bu_i + B_d d_i, \quad i = 0,\dots,N \\ & d_{i+1} = d_i, \quad i = 0,\dots,N \\ & x_i \in \mathcal{X}, \quad u_i \in \mathcal{U}, \quad x_N - x_s \in \mathcal{X}_f \end{aligned}$$

Observability of Augmented System
The augmented system is observable if and only if (A,C) is observable and

$$\begin{bmatrix} A-I & B_d \\ C & C_d \end{bmatrix}$$

has full column rank, i.e. rank $= n_x + n_d$.

Note that

- The number of measured outputs must be large enough: $n_d \le n_y$.
- This is known as Hautus observability condition (or PBH test).

7.1.5.1 Linear State Estimation

$$\begin{aligned} \begin{bmatrix} \widehat{x}(k+1) \\ \widehat{d}(k+1) \end{bmatrix} &= \begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} \begin{bmatrix} \widehat{x}(k) \\ \widehat{d}(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) \\ &+ \begin{bmatrix} L_x \\ L_d \end{bmatrix} \left(-y(k) + C\widehat{x}(k) + C_d\widehat{d}(k)\right) \end{aligned}$$

Error Dynamics

$$\begin{aligned} \begin{bmatrix} \eta(k+1) \\ \eta_d(k+1) \end{bmatrix} &= \begin{bmatrix} x(k+1) - \widehat{x}(k+1) \\ d(k+1) - \widehat{d}(k+1) \end{bmatrix} \\ &= \left(\begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix} + \begin{bmatrix} L_x \\ L_d \end{bmatrix} \begin{bmatrix} C & C_d \end{bmatrix}\right) \begin{bmatrix} x(k) - \widehat{x}(k) \\ d(k) - \widehat{d}(k) \end{bmatrix} \end{aligned}$$

Choose L such that the error dynamics $A + LC$ are stable (i.e. LQG).

7.1.5.2 Steady-State Selection

$$\begin{bmatrix} A-I & B \\ HC & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} -B_d\widehat{d} \\ r - HC_d\widehat{d} \end{bmatrix}$$

7.1.5.3 Offset-free Tracking: Main Result

Let $\kappa(\widehat{x}(k),\widehat{d}(k),r) = u_0^\star$ be the estimation-based control law. Assume

- $n_d = n_y$
- the RHC is recursively feasible and unconstrained for $k \ge j$
- and the closed-loop system

$$\begin{aligned} x(k+1) &= Ax(k) + B\kappa\big(\widehat{x}(k),\widehat{d}(k),r\big) + B_d d \\ \widehat{x}(k+1) &= \big(A + L_x C\big)\widehat{x}(k) + \big(B_d + L_x C_d\big)\widehat{d}(k) \\ &\quad + B\kappa\big(\widehat{x}(k),\widehat{d}(k),r\big) - L_x y(k) \\ \widehat{d}(k+1) &= L_d C\widehat{x}(k) + \big(I + L_d C_d\big)\widehat{d}(k) - L_d y(k) \end{aligned}$$

converges $\big(\widehat{x}(k) \to \widehat{x}_\infty, \widehat{d}(k) \to \widehat{d}_\infty, y(k) \to y_\infty \text{ for } k \to \infty\big)$.
Then $z(k) = Hy(k) \to r$ as $k \to \infty$, i.e. we track the reference in presence of the disturbance.

7.2 Enlarging the Feasible Set

7.2.1 MPC Without Terminal Set

Conditions
We can remove the terminal constraint while maintaining stability if

- initial state lies in sufficiently small subset of feasible set
- N is sufficiently large

such that the terminal state satisfies terminal constraint without enforcing it in the optimization.
In that case, the solution of the finite horizon MPC problem

corresponds to the infinite horizon solution.

Properties

+ Controller defined in a larger feasible set
- Characterization of ROA or specification of required N extremely difficult

Method

Enlarge horizon and check stability by **sampling**.

With larger horizon length N, region of attraction approaches maximum control invariant set.

7.2.2 Soft Constrained MPC

$$
\begin{aligned}
\min_u \quad & \sum_{i=0}^{N-1} x_i^\top Q x_i + u_i^\top R u_i + \ell_\epsilon(\epsilon_i) + x_N^\top P x_N + \ell_\epsilon(\epsilon_N) \\
\text{s.t.} \quad & x_{i+1} = A x_i + B u_i \\
& H_x x_i \le k_x + \epsilon_i \\
& H_u u_i \le k_u \\
& \epsilon_i \ge 0
\end{aligned}
$$

7.2.2.1 Penalty Function Choices

Exact Penalty

The penalty function $\ell_\epsilon(\epsilon_i)$ should be as follows: If the original problem has a feasible solution $z^\star$, then the softened problem should have the same solution and $\epsilon = 0$.

Main Result

$$\ell_\epsilon(\epsilon) = v \cdot \epsilon$$

satisfies the requirement for any $v > \lambda^\star \ge 0$, where $\lambda^\star$ is the optimal Lagrange multiplier for the original problem.

Practical Penalty

Combine linear and quadratic terms for tuning:

$$\ell_\epsilon(\epsilon) = v \cdot \epsilon + s \cdot \epsilon^2$$

with $v > \lambda^\star$ and $s > 0$.

Extension to Multiple Constraints

Assume multiple constraints $g_j(z) \le 0,\ j = 1, \dots, r$. The penalty then reads

$$\ell_\epsilon(\epsilon) = v \cdot \|\epsilon\|_{1/\infty} + \epsilon^\top S \epsilon$$

where

- $\epsilon = [\epsilon_1, \dots, \epsilon_r]^\top$,
- $v > \|\lambda^\star\|_D$,
- and $S \succ 0$ can be used to weight violations differently.

Note that $\|\cdot\|_D$ denotes the dual norm.

Comparison of Penalty Functions

The following properties hold for quadratic and linear penalties respectively

- Quadratic:
  + Well-posed QP (positive definite Hessian)
  - Increase in S hardens soft constraints
  - **Not** exact for any choice of $s > 0$
- Linear:
  + Allows for exact penalties if v large enough
  + If v is large enough, constraints satisfied if possible
  - Large v makes tuning hard and causes numerical issues

7.2.2.2 Simplification: Separation of Objectives

Step 1: Minimize Violation

$$
\begin{aligned}
\min_{u,\epsilon} \quad & \sum_{i=0}^{N-1} \epsilon_i^\top S \epsilon_i + v^\top \epsilon_i \\
\text{s.t.} \quad & x_{i+1} = A x_i + B u_i \\
& H_x x_i \le k_x + \epsilon_i \\
& H_u u_i \le k_u \\
& \epsilon_i \ge 0
\end{aligned}
$$

Step 2: Optimize Performance

$$
\begin{aligned}
\min_u \quad & \sum_{i=0}^{N-1} x_i^\top Q x_i + u_i^\top R u_i + x_N^\top P x_N \\
\text{s.t.} \quad & x_{i+1} = A x_i + B u_i \\
& H_x x_i \le k_x + \epsilon_i^{\min} \\
& H_u u_i \le k_u
\end{aligned}
$$

Properties

+ Simplifies tuning, constraints satisfied if possible
- Requires solving two optimization problems

8 Robust MPC

We assume an uncertain constrained system of the form:

$$
\begin{aligned}
x(k+1) &= g\big(x(k), u(k), w(k); \theta\big), \\
x, u &\in \mathcal{X}, \mathcal{U}, \quad w \in \mathcal{W}, \quad \theta \in \Theta
\end{aligned}
$$

where

- w is random noise, changing with time, influencing system evolution
- θ are unknown, constant or slowly-varying parameters that impact the dynamics

Goals of Robust Constrained Control

Design control law $u(k) = \kappa(x(k))$ such that the system:

1. Satisfies constraints: $\{x(k)\} \subset \mathcal{X}, \{u(k)\} \subset \mathcal{U}$ for all disturbance realizations
2. Is stable: converges to a neighborhood of the origin
3. Optimizes (expected/worst-case) "performance"
4. Maximizes the set $\{x(0) \mid$ Conditions 1-3 are met $\}$

8.1 Modeling Uncertainty

8.1.1 Common Uncertainty Models

Measurement / Input Bias

$$g(x(k), u(k), w(k); \theta) = \widetilde{g}(x(k), u(k)) + \theta$$

θ unknown, but constant.

Linear Parameter Varying System

$$g\big(x(k), u(k), \theta(k)\big) = \left(\sum_{j=0}^{n_\theta} \theta_j(k) A_j\right) x(k) + \left(\sum_{j=0}^{n_\theta} \theta_j(k) B_j\right) u(k)$$

time-varying parameters $\theta(k)$ describe a convex combination of A_j, B_j with $\mathbf{1}^\top \theta(k) = 1$, $\theta(k) \ge 0$.

Additive Stochastic Noise

$$g\big(x(k), u(k), w(k); \theta\big) = Ax(k) + Bu(k) + w(k)$$

where w comes from a known distribution. Used in stochastic MPC.

Additive Bounded Noise

$$g\big(x(k), u(k), w(k); \theta\big) = Ax(k) + Bu(k) + w(k), \quad w \in \mathcal{W}$$

where A, B are known, w is unknown but **bounded** (from a discrete set) and changing at each sampling instance.

Note that

- we may model many nonlinearities in this fashion, but often a conservative model
- the noise is *persistent*, i.e., it does not converge to zero in the limit

8.1.2 Robust Invariant Sets

Robust Positive Invariant Set

A set $\mathcal{O}_\mathcal{W}$ is said to be a robust positive invariant set for the autonomous system $x(k+1) = g\big(x(k), w(k)\big)$ if

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

The same concept is used for CL systems $x(k+1) = g\big(x(k), \kappa(x(k), w(k))\big)$.

A robust positive invariant set exists only for asymptotically stable systems.

Geometric Condition for Robust Invariance

A set $\mathcal{O}_\mathcal{W}$ is a robust positive invariant set if and only if

$$\mathcal{O}_\mathcal{W} \subseteq \operatorname{pre}^\mathcal{W}(\mathcal{O}_\mathcal{W})$$

Computing Robust Invariant Sets

```
  **Input:** $g, \mathcal{X}, \mathcal{W}$
  **Output:** $\mathcal{O}_\infty^\mathcal{W}$
  $\Omega_0 \leftarrow X$
 while true do
     $\Omega_{i+1} \leftarrow \operatorname{pre}^\mathcal{W}(\Omega_i) \cap \Omega_i$
     if $\Omega_{i+1} = \Omega_i$ then
         return $\mathcal{O}_\infty^\mathcal{W} = \Omega_i$
```

where Ω_0 is chosen so that it is as large as possible, choosing any $w \in \mathcal{W}$.

8.1.2.1 Robust Pre-Sets

$$\operatorname{pre}^\mathcal{W}(\Omega) := \{x \mid g(x, w) \in \Omega,\ \forall w \in \mathcal{W}\}$$

Computing Robust Pre-Sets for Linear Systems

Let $g(x(k), w(k)) = Ax(k) + w(k)$ and $\Omega := \{x \mid Fx \le f\}$. Then, fulfilling the state constraints can be seen as parallel displacement of the constraint boundaries

$$
\begin{aligned}
\operatorname{pre}_\mathcal{W}(\Omega) &= \{x \mid Ax + w \in \Omega,\ \forall w \in \mathcal{W}\} \\
&= \{x \mid F_j A x \le f_j - F_j w,\ \forall w \in \mathcal{W}\} \\
&= \{x \mid F_j A x \le f_j - \max_{w \in \mathcal{W}} F_j\} \\
&= \{x \mid F A x \le f - h_\mathcal{W}(F)\}
\end{aligned}
$$

where $h_\mathcal{W}$ is the support function.

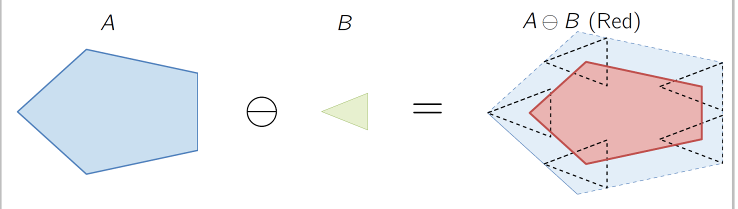
8.1.2.2 Minkowski Sum and Pontryagin Difference

Let A, B be subsets of $\mathbb{R}^n$. The Minkowski Sum is:

$$A \oplus B := \{x + y \mid x \in A, y \in B\}$$

The Pontryagin Difference is:

$$A \ominus B := \{x \mid x + e \in A, \forall e \in B\}$$



For scalar sets

$$
\begin{aligned}
[a, b] \oplus [c, d] &= [a + c, b + d] \\
[a, b] \ominus [c, d] &= [a - c, b - d]
\end{aligned}
$$

8.2 Impact of Bounded Additive Noise

Assume $0 \in \mathcal{W}$ (nominal trajectory).

8.2.1 Uncertain State Evolution

$$
\begin{aligned}
\phi_i &= A^i x_0 + \sum_{j=0}^{i-1} A^j B u_{i-1-j} + \sum_{j=0}^{i-1} A^j w_{i-1-j} \\
&= x_i + \sum_{j=0}^{i-1} A^j w_{i-1-j}
\end{aligned}
$$

8.2.2 Cost

$$J(x_0, U, W) := \sum_{i=0}^{N-1} I(\phi_i(x_0, U, W), u_i) + I_f(\phi_N(x_0, U, W))$$

There are multiple methods to achieve **independence** of W:

- Minimize the expected value
  $$J_N(x_0, U) = \mathbb{E}_w\Big[J(x_0, U, W)\Big]$$
- Consider the worst-case
  $$J_N(x_0, U) = \max_{W \in \mathcal{W}^{N-1}} J(x_0, U, W)$$
- Consider the nominal case
  $$J_N(x_0, U) = J(x_0, U, 0)$$
- Use an estimate of the disturbance (certainty equivalence)
  $$J_N(x_0, U) = J(x_0, U, \widehat{W})$$

8.2.3 Robust Constraint Satisfaction

8.2.3.1 State Constraints

$$
\begin{aligned}
\phi_i(x_0, U, W) &= x_i \in \big\{x \big| x + \bar{w} \in \mathcal{X}, \bar{w} \in \mathcal{F}_i\big\} \\
&= x_i \in \mathcal{X} \ominus \mathcal{F}_i
\end{aligned}
$$

where the *disturbance reachable set* (DRS)

$$
\begin{aligned}
\mathcal{F}_i &= \mathcal{W} \oplus A\mathcal{W} \oplus \cdots A^{i-1}\mathcal{W} \\
&= \oplus_{j=0}^{i-1} A^j \mathcal{W}
\end{aligned}
$$

shares the dynamics of the system:

$$\mathcal{F}_{i+1} = A\mathcal{F}_i \oplus \mathcal{W}$$

Polytopic Constraints

For the polytopic case $\mathcal{X} = \{x \mid Fx \le f\}$ we get

$$
\begin{aligned}
F x_i + F \sum_{j=0}^{i-1} A^j w_{i-1-j} &\le f, \qquad \forall W \in \mathcal{W}^i \\
F x_i &\le f - \max_{W \in \mathcal{W}^i} F \sum_{j=0}^{i-1} A^j w_{i-1-j} \\
&= f - h_{\mathcal{W}^i}\left(F \sum_{j=0}^{i-1} A^j\right)
\end{aligned}
$$

8.2.3.2 Terminal Constraint

$$
\begin{aligned}
\phi_N(x_0, U, W) &\in \mathcal{X}_f \\
x_N &\in \mathcal{X}_f \ominus \mathcal{F}_N
\end{aligned}
$$

8.3 Robust Open-Loop MPC

$$
\begin{aligned}
\min_U \ & \sum_{i=0}^{N-1} I(x_i, u_i) + I_f(x_N) \\
\text{subject to } & x_{i+1} = Ax_i + Bu_i \\
& x_0 = x(k) \\
& x_i \in \mathcal{X} \ominus \mathcal{F}_i \\
& u_i \in \mathcal{U} \\
& x_N \in \mathcal{X}_f \ominus \mathcal{F}_N
\end{aligned}
$$

here $\mathcal{X}_f \subseteq \mathcal{X}$ is a robust invariant set for the dynamics $x(k+1) = Ax(k) + w(k), \forall w$.

Robust open-loop MPC potentially has a very small region of attraction, in particular for unstable systems.

8.4 Robust Closed-Loop MPC

Pre-stabilization

$$\mu_i(x_i) = K x_i + v_i$$

Fixed K that stabilizes the CL (i.e. $A + BK$ is stable)

+ simple
− conservative
− Kx_i usually large → remaining optimization in v_i shrinks

Linear Feedback

$$\mu_i(x_i) = K_i x_i + v_i$$

Optimize over time-varying K_i and v_i

− Non-convex and difficult to solve

Disturbance Feedback

$$\mu_i(x_i) = \sum_{j=0}^{j-1} M_{ij} + v_i$$

Optimize over time-varying M_{ij} and v_i, hence over *all* previous disturbances

+ Convex version of *Linear feedback*
+ Least restrictive viable method
+ Very effective
− Computationally expensive (matrix optimization variables)

Constraint Tightening MPC & Tube MPC

$$\mu_i(x_i) = K(x_i - \bar{x}_i) + v_i$$

Fixed K that stabilizes the CL (i.e. $A + BK$ is stable) and optimize over $\bar{x}_i$ and v_i

+ Simple, often effective
+ FB on *deviation from* x_i → remaining opt. in v_i grows

8.4.1 Separation of Nominal and Disturbance Control

1. **Nominal Control** of the disturbance free system
   $$z(k+1) = Az(k) + Bv(k)$$
2. **Disturbance Control** that compensates for deviations from the nominal trajectory
   $$u_i = K(x_i - z_i) + v_i$$
   The controller K is computed offline. We then optimize over z_i and v_i.

8.4.2 Error Dynamics / DRS

$$
\begin{aligned}
e_{i+1} &= x_{i+1} - z_{i+1} \\
&= \underbrace{(A + BK)}_{A_K} e_i + w_i \\
&= \sum_{j=0}^{i-1} A_K^j w_{i-j-1}, \qquad e_0 = 0
\end{aligned}
$$

The DRS for the closed-loop system containing all possible errors is defined as

$$\mathcal{F}_i = \mathcal{W} \oplus A_K\mathcal{W} \ldots \oplus A_K^{i-1}\mathcal{W} = \bigoplus_{j=0}^{i-1} A_K^j \mathcal{W}, \quad \mathcal{F}_0 := \{0\}$$

Advantages of Feedback

Also works for open-loop **unstable** systems.

8.4.2.1 Minimum Robust Invariant Set (mRPI)

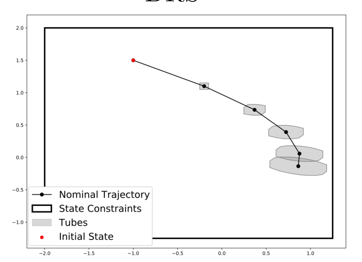
$$\mathcal{F}_\infty = \bigoplus_{j=0}^{\infty} A_K^j \mathcal{W}$$

```
 $\Omega_0 \leftarrow \{0\}$
 while  do
     $\Omega_{i+1} \leftarrow \Omega_i \oplus A^i\mathcal{W}$
     if $\Omega_{i+1} = \Omega_i$ then
         return $\mathcal{F}_\infty = \Omega_i$
```

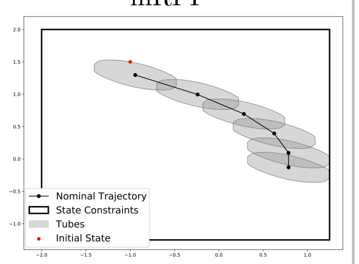
Note that the set $\mathcal{F}_i$ increases in size with i and converges to $\mathcal{F}_\infty$. If convergence does not happen in finite time, there are methods to slightly enlarge $\mathcal{F}_i$ so that $\mathcal{F}_i > F_\infty$.

Comparison of the two variants:

DRS



mRPI



Scalar Linear DTI

$$
\begin{aligned}
x(k+1) &= Ax(k) + Bu(k) + w(k) \\
u(k) &= Kx(k)
\end{aligned}
$$

$$\mathcal{F}_\infty = \left[\frac{a}{1 - A_K}, \frac{b}{1 - A_K}\right]$$

with

$$A_K = A + BK \quad \text{and} \quad \mathcal{W} = [a, b]$$

8.4.3 Input-to-State Stability

Assume that the optimal cost J^* is Lipschitz continuous (true for linear systems, convex constraints and continuous stage costs)

$$|J^*(Ax + Bu^*(x) + w) - J^*(Ax + Bu^*(x))| \le \gamma \|w\|$$

for some $\gamma > 0$.

Then, the Lyapunov decrease can be bounded as

$$J^*(Ax + Bu^*(x) + w) - J^*(x) \le -I(x, u^*(x)) + \gamma\|w\|$$

Hence,

- amount of decrease $\|x\|$
- amount of increase upper bounded by $\max_{w \in \mathcal{W}} \gamma\|w\|$

and the system moves toward the origin until there is a balance between the size of x and the size of w.

8.5 Constraint-Tightening MPC

$$z_i \oplus \mathcal{F}_i \subseteq \mathcal{X} \quad \Leftrightarrow \quad z_i \in \mathcal{X} \ominus \mathcal{F}_i$$

where the DRS $\mathcal{F}_i$ can be computed offline.

8.5.1 MPC Problem

$$
\begin{aligned}
\min_{Z,V} \ & \sum_{i=0}^{N-1} I(z_i, v_i) + I_f(z_N) \\
\text{subj. to } & z_{i+1} = Az_i + Bv_i && \forall i = 0, 1, \dots, N-1 \\
& z_i \in \mathcal{X} \ominus \mathcal{F}_i && \forall i = 0, 1, \dots, N-1 \\
& v_i \in \mathcal{U} \ominus K\mathcal{F}_i && \forall i = 0, 1, \dots, N-1 \\
& z_N \in \mathcal{X}_f^{ct} \ominus \mathcal{F}_N \\
& z_0 = x(k)
\end{aligned}
$$

The terminal set $\mathcal{X}_f^{\text{ct}}$ is a robust invariant set.

$$u(k) = v_0^* + K(x(k) - z_0^*) = v_0^*$$

8.5.1.1 Constraint-Tightening MPC Assumptions

S3.1 The stage cost $I(z, v)$ is positive definite and only zero at the origin.

S3.2 The terminal set $\mathcal{X}_f^{\text{ct}}$ is a robust positively invariant set for the dynamics $x(k+1) = Ax(k) + Bu(k) + w(k)$ under the terminal controller $\kappa_f^{\text{ct}} z(k) = K_{\text{ct}}\, z(k)$:

$$(A + BK_{ct})z + w \in \mathcal{X}_f^{\text{ct}}, \quad \forall z \in \mathcal{X}_f^{\text{ct}}, \forall w \in \mathcal{W}$$

All state and input constraints are satisfied in $\mathcal{X}_f^{\text{ct}}$.

$$\mathcal{X}_f^{\text{ct}} \subseteq \mathcal{X}, \quad K_{\text{ct}}\mathcal{X}_f^{\text{ct}} \subseteq \mathcal{U}$$

8.5.1.2 Recursive Feasibility Guarantee

Let the terminal ingredients $(I_f, \mathcal{C}_f^{ct})$ be chosen such that

- $\mathcal{X}_f^{\text{ct}} \subseteq \mathcal{X}$
- for all $z \in \mathcal{X}_f^{ct}$:
  - $Kz \in \mathcal{U}$
  - $(A + BK)z + w \in \mathcal{X}_f^{ct}$, $\forall w \in \mathcal{W}$
  - $I_f\big((A + BK)z\big) - I_f(z) \le -I(z, \pi_f(z))$

Let

- $\mathcal{X}_N$ be the feasible set
- and $V^*(x(k))$ be the optimizer of the robust constraint-tightening MPC problem for $x(k) \in \mathcal{X}_N$.

Then **robust invariance** is achieved

$$Ax(k) + Bv_0^*(x(k)) + w \in \mathcal{X}_N, \ \forall w \in \mathcal{W}$$

i.e. the problem is **recursively feasible**.

8.5.1.3 Stability Guarantee

Uses input-to-state stability → details in advanced MPC.

8.6 Tube MPC

The goal is to plan a nominal trajectory z_i i.e. the tube center, such that all possible state trajectories $z_i \oplus \mathcal{E}$ are within constraints.

$$
\begin{aligned}
z_i \oplus \mathcal{E} \subseteq \mathcal{X} \quad &\Leftrightarrow \quad z_i \in \mathcal{X} \ominus \mathcal{E} \\
u_i \in K\mathcal{E} \oplus v_i \subset \mathcal{U} \quad &\Leftrightarrow \quad v_i \in \mathcal{U} \ominus K\mathcal{E}
\end{aligned}
$$

where the mRPI $\mathcal{E}$ again can be computed offline.

Ideally $\mathcal{E}$ is the the minimum RPI set $\mathcal{F}_\infty$. A larger set can be chosen but **must** be invariant.

8.6.1 Tube MPC Problem

$$
\begin{aligned}
\min_{Z,V} \ & \sum_{i=0}^{N-1} I(z_i, v_i) + I_f(z_N) \\
\text{subj. to } & z_{i+1} = Az_i + Bv_i && \forall i = 0, 1, \dots, N-1 \\
& z_i \in \mathcal{X} \ominus \mathcal{E} && \forall i = 0, 1, \dots, N-1 \\
& v_i \in \mathcal{U} \ominus K\mathcal{E} && \forall i = 0, 1, \dots, N-1 \\
& z_N \in \mathcal{X}_f \\
& x_0 \in z_0 \oplus \mathcal{E}
\end{aligned}
$$

with the applied control law

$$\mu_{\mathsf{tube}}(x) = K(x - z_0^*(x)) + v_0^*(x)$$

Note that

- the cost is calculated with respect to the tube centers (nominal system)
- the terminal set is w.r.t. the tightened constraints
- The first tube center z_0 is also an optimization variable i.e. z_0 has to be within $\mathcal{E}$ of x_0
- The nominal dynamics can be different from the disturbance dynamics
- The state trajectory only converges to a neighborhood of the origin → $\sum_{i=0}^{\infty} \ell(x_i, u_i)$ can be infinite

Comparison to Constraint-Tightening MPC

+ The terminal set $\mathcal{X}_f$ is simply the invariant set from the OL problem. **No** robust invariance needed!
+ Nominal and disturbed dynamics are completely decoupled
+ Region of convergence can be specified
- Feasible set smaller

Implementation:

Offline

1. Choose a stabilizing controller K so that $A + BK$ is stable.
2. Compute the minimal robust invariant set $\mathcal{E} = \mathcal{F}_\infty$ for the system $x(k+1) = (A + BK)x(k) + w(k), w \in \mathcal{W}^1$
3. Compute the tightened constraints
   $$
   \begin{aligned}
   \widetilde{\mathcal{X}} &:= \mathcal{X} \ominus \mathcal{E} \\
   \widetilde{\mathcal{U}} &:= \mathcal{U} \ominus K\mathcal{E}
   \end{aligned}
   $$
4. Choose terminal weight function I_f and constraint $\mathcal{X}_f$ satisfying the tube assumptions.

Online

1. Measure / estimate state x
2. Solve the Tube MPC optimization problem
3. Set the input to $u = K(x - Z_0^*(x)) + v_0^*(x)$

8.6.1.1 Tube MPC Assumptions

S2.1 The stage cost is a positive definite function: strictly positive and only zero at the origin.

S2.2 The terminal set is invariant for the **nominal system** under the local controller $\kappa_f(z)$:

$$Az + B\kappa_f(z) \in \mathcal{X}_f, \quad \forall z \in \mathcal{X}_f$$

and satisfies the **tightened** constraints:

$$\mathcal{X}_f \subseteq \mathcal{X} \ominus \mathcal{E}, \quad \kappa_f(z) \in \mathcal{U} \ominus K\mathcal{E} \quad \forall z \in \mathcal{X}_f$$

S2.3 Terminal cost is a Lyapunov function in $\mathcal{X}_f$:

$$I_f(Az + B\kappa_f(z)) - I_f(z) \le -I(z, \kappa_f(z)), \quad \forall z \in \mathcal{X}_f$$

1. The set $\mathcal{E}$ is a robust invariant set for the system under $u(k) = K_\mathcal{E}\, x(k)$

8.6.1.2 Nominal Stability and Recursive Feasibility

Theorem: Robust Invariance of Tube MPC

Let $\mathcal{Z}$ be the feasible set, i.e $\mathcal{Z} := \{x \mid \mathcal{Z}(x) \neq \emptyset\}$. Then $\mathcal{Z}$ is a robust invariant set for the closed-loop system

$$x(k+1) = Ax(k) + B\mu_{\text{tube}}(x(k)) + w(k)$$

under the constraints $x \in \mathcal{X}$, $u \in \mathcal{U}$.

Theorem: Robust Stability of Tube MPC

The state $x(k)$ of the system

$$x(k+1) = Ax(k) + B\mu_{\text{tube}}(x(k)) + w(k)$$

converges in the limit to the set $\mathcal{E}$ and hence,

$$\lim_{k \to \infty} \operatorname{dist}(x(k), \mathcal{E}) = 0$$

where $\operatorname{dist}(\cdot, \cdot)$ denotes any distance function.

The cost for the nominal trajectory vanishes:

$$\lim_{k \to \infty} J(z_0^*(x(k))) = 0 \quad \Rightarrow \quad \lim_{k \to \infty} z_0^*(x(k)) = 0$$

8.6.1.3 Proof Sketch

Recursive Feasibility

Given the optimal solution $\big(v_0^*, \dots, v_{N-1}^*, z_0^*, \dots, z_N^*\big)$ for $x(k)$, the next state is

$$x(k+1) = Ax(k) + BK(x(k) - z_0^*) + Bv_0^* + w, \quad w \in \mathcal{W}$$

By construction, $x(k+1) \in z_1^* \oplus \mathcal{E}$. The shifted trajectory

$$\big\{v_1^*, \dots, v_{N-1}^*, \kappa_f(z_N^*)\big\}, \quad \{z_1^*, \dots, z_N^*, Az_N^* + B\kappa_f(z_N^*)\}$$

is feasible for all $x(k+1)$.

Details:

- Feasibility of $\big\{v_1^*, \dots, v_{N-1}^*, \kappa_f(z_N^*)\big\}$
  - $v_i^* \in \mathcal{U} \ominus K\mathcal{E}$ for $i = 1, \dots, N-1$ given from feas. at k.
  - $\kappa_f(z_N^*) \in \mathcal{U} \ominus K\mathcal{E}$ for $z_N^* \in \mathcal{X}_f$ by Assumption 2. (Section **??**)
- Feasibility of $\{z_1^*, \dots, z_N^*, Az_N^* + B\kappa_f(z_N^*)\}$
  - $z_i^* \in \mathcal{X} \ominus \mathcal{E}$ for $i = 1, \dots, N-1$ given from feas. at k.
  - $z_N^* \in \mathcal{X}_f \subseteq \mathcal{X} \ominus \mathcal{E}$ given from feasibility at k.
  - $Az_N^* + B\kappa_f(z_N^*) \in \mathcal{X}_f$ from invariance of $\mathcal{X}_f$ by Assumption 2. (Section **??**)
  - $x(k+1) \in z_1^* \oplus \mathcal{E}$ follows from:

$$
\begin{aligned}
\underbrace{x(k+1) - z_1^*}_{e_{k+1}} &= Ax(k) + BK(x(k) - z_0^*) + Bv_0^* + w \\
&\qquad - Az_0^* + Bv_0^* \\
&= (A + BK)\underbrace{(x(k) - z_0^*)}_{e_k} + w(k) \\
&\in \mathcal{E} \text{ if } (x(k) - z_0^*) \in \mathcal{E} \\
&\Rightarrow x(k+1) \in z_1^* \oplus \mathcal{E}
\end{aligned}
$$

Stability

As in standard MPC, we have the value function expression

$$J^*(x(k)) = \sum_{i=0}^{N-1} \ell(z_i^*, v_i^*) + \ell_f(z_N^*)$$

At the next time step $k+1$, it holds that

$$
\begin{aligned}
J^*(x(k+1)) &\le \sum_{i=1}^{N} \ell(z_i^*, v_i^*) + \ell_f(z_{N+1}) \\
&= J^*(x(k)) - \underbrace{\ell(z_0^*, v_0^*)}_{\ge 0} \\
&\qquad - \underbrace{\ell_f(z_N^*) - \ell_f(z_{N+1}) - \ell(z_N^*, \kappa_f(z_N^*))}_{\le 0}
\end{aligned}
$$

where the last inequality uses the fact that ℓ_f is a Lyapunov function on $\mathcal{X}_f$.

9 Numerical Methods and Implementation

9.1 Explicit MPC

For **small systems** (3-6 states),

1. The optimization problem is **solved offline and parametrically** for a set of states.
2. During runtime, the control input is **queried from the pre-computed solution** (piecewise affine for linear system/ constraints).

9.1.1 Active Set and Critical Region

Active Set
The active set $A(x)$ is the set of indices of active constraints at a given state $x \in \mathcal{X}_0$:

$$A(x) = \left\{ j \in \{1, \dots, m\} \middle| G_j U - E_j x(k) = w_j \right\}$$

Its complement is the set of non-active constraints:

$$NA(x) = \left\{ j \in \{1, \dots, m\} \middle| G_j U - E_j x(k) < w_j \right\}$$

Critical Region
The critical region CR is the set of states x for which the same active set $A(x)$ is active at the optimum:

$$CR = \{x \in \mathcal{X} : A(x) = A(x^*)\}$$

9.1.2 Quadratic Cost

See quadratic cost CFTOC.

9.1.3 $1/\infty$-Norm Cost

See $1/\infty$-norm CFTOC.

9.1.4 Point Location

Given m regions, two possible methods to find the critical region CR^j for a given state $x(k)$. Then, the corresponding piecewise affine function must be evaluated at $x(k)$ to obtain the control input.

Sequential Search ($\mathcal{O}(m)$)
Simply check each polyhedron until the one containing $x(k)$ is found:

```
given $x = x(k)$
for each $j$ do
    if $A_j x + b_j \leq 0$ then
        $x$ is in region $j$
```

Logarithmic/Tree Search ($\mathcal{O}(\log(m))$)
By constructing a search tree offline, the critical region can potentially be found in logarithmic time. The search tree is constructed by recursively splitting the polyhedra CR^j by hyperplanes. Reasonable for < 1000 regions.

9.2 Iterative MPC

Given an initial guess x_0, cost function $f(x)$, and a feasible set $\mathbb{Q}$, **iterative optimization methods** compute a sequence of iterates

$$x_{k+1} = \Psi(x_k, f, \mathbb{Q}), \quad K = 0, 1, \dots, m-1$$

that are ϵ-stationary

$$\|f(x^{(m)}) - f(x^*)\| \leq \epsilon$$

and sufficiently feasible

$$\text{dist}(x^{(m)}, \mathbb{Q}) \leq \delta$$

9.2.1 Unconstrained Minimization

Problem
Solve

$$\min_x f(x)$$

where we assume

- f convex and twice continuously differentiable
- f differentiable at x^*
- optimal value $p^* = \min_x f(x)$ finite

The goal is to iteratively solve for the necessary and sufficient condition

$$\nabla f(x^*) = 0$$

Descent Methods
Descent methods update the current iterate x_k in the search direction Δx_k with a step size h_k:

$$x_{k+1} = x_k + h_k \Delta x_k \quad \rightarrow \quad f(x_{k+1}) < f(x_k)$$

where

- Δx is a descent direction
- There exists a $h_k > 0$ such that we obtain a cost decrease, if we step to a certain extent into the opposite gradient direction:

  $$\exists h_k > 0 \rightarrow f(x_{k+1}) < f(x_k) \text{ if } \nabla f(x_k)^\top \Delta x_k < 0$$

  The exact direction is to be chosen (e.g. opposite gradient or Newton).

Algorithm

```
given $x_0 \in \mathrm{dom}(f)$
repeat
    1. Compute a descent direction $\Delta x_k$
    2. Line search: Choose step size $h_k > 0$ such that
       $f(x_k + h_k \Delta x_k) < f(x_k)$
    3. Update $x_{k+1} := x_k + h_k \Delta x_k$
until termination condition (e.g. $f(x_k) - f(x^*) \leq \varepsilon_1$ or $\|x_k - x_{k-1}\| \leq \varepsilon_2$)
```

9.2.1.1 Gradient Descent / First-Order Method

Assume

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \quad \forall x, y \in \mathbb{R}^n$$

The update rule becomes

$$x_{k+1} = x_k - \frac{1}{L}\nabla f(x_k)$$

which is guaranteed to converge to a local minimum of f.

Finding L
Assuming f is L-smooth, the Lipschitz constant L can be determined globally. It is the maximum spectral norm of the Hessian $\nabla^2 f(x)$, i.e.

$$\begin{aligned} L &= \max_{x \in \mathcal{X}} \|\nabla^2 f(x)\|_2 \\ &= |\max_{x \in \mathcal{X}} \sigma_{\max}(\nabla^2 f(x))| \\ &= |\lambda_{\max}(\nabla^2 f(x))| \end{aligned}$$

As the Hessian is always symmetric, L is simply its **largest absolute eigenvalue**.

9.2.1.2 Newton's Method / Second-Order Method

Newton's method minimizes the second-order Taylor expansion of f around the current iterate x_k:

$$\begin{aligned} x_{k+1} &= \arg\min_x \; f(x_k) + \nabla f(x_k)^\top (x - x_k) + \cdots \\ &\quad + \frac{1}{2}(x - x_k)^\top \nabla^2 f(x_k)(x - x_k) \\ x_{k+1} &= x_k \underbrace{- \left(\nabla^2 f(x_k)\right)^{-1} \nabla f(x_k)}_{\text{Newton's direction}} \end{aligned}$$

As the second-order Taylor expansion is **not** necessarily an upper bound, we can obtain a cost increase. Hence, the remaining problem is to choose $h_k > 0$ such that the update decreases f:

$$x_{k+1} = x_k - h_k \left(\nabla^2 f(x_k)\right)^{-1} \nabla f(x_k)$$

for example by using line search.

9.2.1.3 Line Search

Exact Line Search
Compute the best step size h_k along the direction of descent Δx_k:

$$h_k = \arg\min_{h > 0} f(x_k + h \Delta x_k)$$

This is an optimization in one variable which can be solved by bisection (slow, many evaluations of f).

Backtracking Line Search (Inexact)
Find a step size h_k that decreases the cost function f sufficiently:

```
Initialize $h_k = 1$
while $f(x_k + h_k \Delta x_{nt}) > f(x_k) + \alpha h_k \nabla f(x_k)^\top \Delta x_{nt}$ do
    $h_k \leftarrow \beta h_k$
```

with $\alpha \in (0, 0.5)$ and $\beta \in (0, 1)$.

9.2.2 Constrained Minimization

- **PGD**: better for small horizon lengths, faster iterations, but more iterations needed.
- **Interior Point Method**: better for large horizon lengths, fewer iterations, but slower iterations.

9.2.2.1 Equality Constrained Newton's Method

Incorporating equality constraints

$$\begin{aligned} \min_x \quad & f(x) \\ \text{subj. to} \quad & Ax = b \end{aligned}$$

into Newton's method (and rewriting $x - x_k = \Delta x$) yields

$$\begin{aligned} \Delta x_{nt}(x_k) \in \arg\min_{\Delta x_k} \; & \frac{1}{2}\Delta x \nabla^2 f(x_k) \Delta x + \nabla f(x_k) \Delta x \\ \text{subj. to} \quad & A \Delta x_k = -A x_k + b \end{aligned}$$

Descent Direction
The corresponding descent direction can be found by solving a linear system of equations

$$\begin{bmatrix} \nabla^2 f(x_k) & A^\top \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x_k \\ \lambda_k \end{bmatrix} = \begin{bmatrix} -\nabla f(x_k) \\ 0 \end{bmatrix} \quad \begin{matrix} \text{Stationarity} \\ A\Delta x = 0 \end{matrix}$$

If the equality constraint is fulfilled at initilization, the nullspace constraint $A\Delta x = 0$ enforces that the next x_{k+1} still satisfies the equality constraint:

$$A x_{k+1} = A x_k + h_k A \Delta x_k = b$$

Conclusion
Equality constraints yield easy optimization problems as the search directions can be found by solving a simple linear system.

9.2.2.2 Projected Gradient Method

Problem

Consider the constrained convex optimization problem

$$\min f(x) \quad \text{subject to} \quad x \in \mathbb{Q}$$

where f is convex and L-smooth, and the feasible set $\mathbb{Q}$ is convex.

Step Projection
Constraints on x can be incorporated by projecting the gradient onto $\mathbb{Q}$ using the euclidian projection:

$$\begin{aligned} x_{k+1} &= \Pi_{\mathbb{Q}}(x_k - h_k \nabla f(x_k)) \\ \Pi_{\mathbb{Q}}(x) &= \arg\min_{y \in \mathbb{Q}} \|y - x\|^2 \end{aligned}$$

Under our assumptions, if we choose step size $h_k = \frac{1}{L}$, Projected Gradient Descent (PGD) converges to a local minimum of f with convergence rates similar to the unconstrained case.

MPC
In the MPC problem setup, the optimization variable is the control input U. Therefore, **input constraints** can be incorporated by using PGD in the CFTOC formulation with substitution.

MPC with **state constraints** are more complicated as the projection of the intersection $(\mathcal{U} \times \mathcal{X}) \cap \{z | Az = b\}$ is not trivial. One approach could be to solve the dual problem (simple inequalities $\lambda >= 0$ with cheap projection), which would require special attention to preserving the feasibility of the primal problem.

9.2.2.3 Interior Point Method

Problem

$$\begin{aligned} \min_x \quad & f(x) \\ \text{subj. to} \quad & g_i(x) \leq 0 \quad i = 1, \dots, m \\ & Cx = d \end{aligned}$$

Assumptions

- f, g_i are convex, and twice continuously differentiable.
- $f(x^*)$ is finite and attained
- strict feasibility: $\exists x$ such that $g_i(x) < 0$ and $Cx = d$.
- feasible set is closed and compact
- strong duality holds $\rightarrow$ KKT conditions can be used

Primal-Dual relaxed KKT Conditions
Primal-Dual interior-point methods solve the following relaxed KKT system of equations simultaneously for both, the primal and dual variables:

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

where

- $s \in \mathbb{R}^m$ are slack variables
- $\{(x, \nu, \lambda, s) | \text{above eqns. hold}\}$ is called *primal-dual central path*
- one attempts to successively reduce κ to zero (central path)

At every iteration, the KKT conditions are linearized at the current iterate $(x_k, \nu_k, \lambda_k, s_k)$:

$$\begin{bmatrix} H & C^\top & G^\top & 0 \\ C & 0 & 0 & 0 \\ G & 0 & 0 & \mathbb{I} \\ 0 & 0 & S & \Lambda \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \nu \\ \Delta \lambda \\ \Delta s \end{bmatrix} = - \begin{bmatrix} \nabla f(x) + C^\top \nu + G^\top \lambda \\ Cx - d \\ g(x) + s \\ S\lambda - v \end{bmatrix}$$

where

$$\begin{aligned} S &= \mathrm{diag}(s_1, \dots, s_m) \\ \Lambda &= \mathrm{diag}(\lambda_1, \dots, \lambda_m) \\ H(x, \lambda) &= \nabla^2 f(x) + \sum_{i=1}^{m} \lambda_i \nabla^2 g_i(x) \\ G(x)^\top &= \begin{bmatrix} \nabla g_1(x) & \nabla g_2(x) & \cdots & \nabla g_m(x) \end{bmatrix} \end{aligned}$$

and v replaces κ in the KKT conditions (further relaxation).

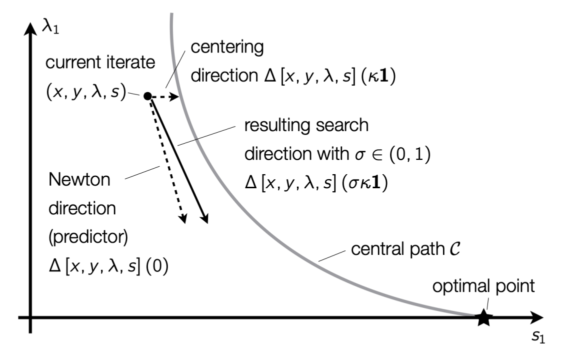
The resulting direction $\Delta[x, \nu, \lambda, s](v)$ is found by solving the linear system and depends on the choice of the relaxation parameter v. For

- $v = 0$, the direction is a **Newton step**.
- $v = \kappa\mathbf{1}$, the direction is called a centering direction that approaches the central path.

Predictor-Corrector Method
Predictor-Corrector methods use a linear combination of the two directions, i.e.

$$\Delta[x, \nu, \lambda, s](v) = \Delta[x, \nu, \lambda, s](\sigma\kappa\mathbf{1})$$

where $\sigma \in (0, 1)$, which ensures fast convergence.


λ_1
current iterate (x, y, λ, s)
centering direction $\Delta[x, y, \lambda, s](\kappa\mathbf{1})$
resulting search direction with $\sigma \in (0, 1)$ $\Delta[x, y, \lambda, s](\sigma\kappa\mathbf{1})$
Newton direction (predictor) $\Delta[x, y, \lambda, s](0)$
central path $\mathcal{C}$
optimal point
s_1

10 Appendix

$$\sum_{k=0}^{\infty} q^k = \frac{1}{1-q} \quad \text{for } |q| < 1$$