

# Model Predictive Control

Silvan Stadelmann - 5. November 2025 - v1.0.1

[github.com/silvasta/summary-mpc](https://github.com/silvasta/summary-mpc)



## Contents

<b>1 Nominal MPC</b>	<b>2</b>
<b>2 Practical MPC</b>	<b>2</b>
2.1 Steady-state Target Problem . . . . .	2
2.2 Offset-free Reference Tracking . . . . .	3
2.3 Soft Constraints . . . . .	4
<b>3 Robust MPC</b>	<b>5</b>
3.1 Robust Constraint-Tightening MPC . . . . .	5
3.2 Robust Tube MPC . . . . .	6
<b>4 Invariance</b>	<b>7</b>
4.1 Control Invariance . . . . .	7
4.2 Robust Invariance . . . . .	7
4.3 Computing Invariant Sets and Pre-sets . . . . .	8
<b>5 Optimization</b>	<b>9</b>
5.1 Convex Sets . . . . .	9
5.2 Convex Functions . . . . .	10
5.3 Optimality Conditions . . . . .	10
5.4 Convex Optimization Problems . . . . .	10
<b>6 Introduction to Systems and Controls</b>	<b>11</b>
6.1 Linearization . . . . .	11
6.2 Discretization . . . . .	11
6.3 Analysis of Discrete-Time LTI Systems . . . . .	12
<b>7 Optimal Control</b>	<b>12</b>
7.1 Unconstrained Finite Horizon Control Problem . . . . .	12
7.2 Infinite Horizon Control Problem . . . . .	13
7.3 Constrained Finite Time Optimal Control Problem . . . . .	14
<b>8 Implementation, needed?</b>	<b>16</b>

## Requirements and Steps to MPC

- 1 Model of the System** dynamics to state space
- 2 State Estimator** track trajectory and disturbance
- 3 Optimal Control Problem** define strategy
- 4 Optimization problem** mathematical formulation
- 5 Get Optimal Control Sequence** solve optimization
- 6 Verify Closed-Loop Performance** iterative tests

# 1 Nominal MPC

## MPC Mathematical Formulation

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

**Constraints**

$$x_0 = x(k)$$

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

$$x_i \in \mathcal{X}$$

$$u_i \in \mathcal{U}$$

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

$l_f(\cdot)$  and  $\mathcal{X}_f$  are chosen to mimic an infinite horizon.

What can go wrong with *standard MPC*?

- No feasibility guarantee, the problem may not have a solution
- No stability guarantee, trajectories may not converge to origin

## Stability of MPC - Main Result

### Assumptions

- 1 Stage cost is strictly positive and only zero at the origin
- 2 Terminal set is **invariant** under 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  $\mathcal{X}_f$ :

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

- 3 Terminal cost is a continuous **Lyapunov function** s.t.

$$l_f(x_{i+1}) - l_f(x_i) \leq -l(x_i, \kappa_f(x_i)) \quad \forall x_i \in \mathcal{X}_f$$

**Theorem 1.** Under the previous assumptions, the closed-loop system under the MPC control law  $u_0^*(x)$  is asymptotically stable and the set  $\mathcal{X}_f$  is positive invariant for

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

**Finite-horizon MPC may not satisfy constraints for all time!**

**Finite-horizon MPC may not be stable!**

- An infinite-horizon provides stability and invariance.
- Infinite-horizon *faked* by forcing final state into an invariant set for which there exists invariance-inducing controller, whose infinite-horizon cost can be expressed in closed-form.
- Extends to non-linear systems, but compute sets is difficult!

# 2 Practical MPC

## 2.1 Steady-state Target Problem

- Reference is achieved by the target state  $x_s$  if  $z_s = Hx_s = r$
- Target state should be a steady-state, i.e.  $x_s = Ax_s + Bu_s$

$$\begin{aligned} x_s &= Ax_s + Bu_s \iff \begin{bmatrix} \mathbb{I} - A & -B \\ H & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix} \\ z_s &= Hx_s = r \end{aligned}$$

$\nexists$  solution  $\rightarrow \min(Hx_s - r)^\top Q_s(Hx_s - r)$  (closest  $x$  to  $r$ )

If  $\exists$  multiple feasible  $u_s \rightarrow$  compute  $\min u_s^\top R_s u_s$  (cheapest)

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

## 2.2 Offset-free Reference Tracking

### Reference Tracking

$$\begin{aligned} \Delta x_{k+1} &= x_{k+1} - x_s \\ &= A\Delta x_k + Bu_k - (Ax_s + Bu_s) \\ &= A\Delta x_k + B\Delta u_k \\ G_x x &\leq h_x \Rightarrow G_x \Delta x \leq h_x - G_x x_s \\ G_u u &\leq h_u \Rightarrow G_u \Delta u \leq h_u - G_u u_s \end{aligned}$$

### Convergence

Assume feasible target with  $x_s \in \mathcal{X}, u_s \in \mathcal{U}$ , choose terminal weight  $V_f(x)$  and constraint  $\mathcal{X}_f$  as in regulation case satisfying

$$V_f(x(k+1)) - V_f(x(k)) \leq -l(x(k), Kx(k))$$

and  $(A + BK)x \in \mathcal{X} \quad \forall x \in \mathcal{X}_f$  for both

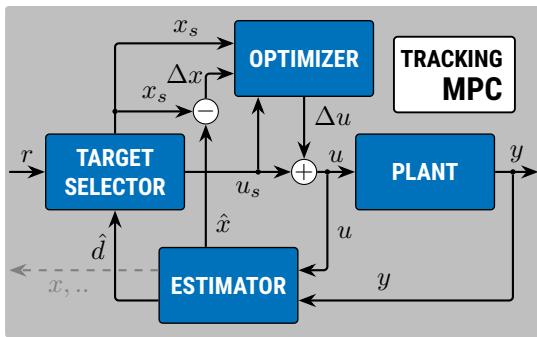
If in addition the target reference  $x_s, u_s$  is such that

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

then the closed loop system converges to the target reference.

*Proof.* Invariance under local control law is inherited from regulation case. Constraint satisfaction is provided by extra conditions and convergence comes from the asymptotic stability of the regulation problem:  $\Delta x(k) \rightarrow 0$  for  $k \rightarrow \infty$   $\square$

**Terminal set** use  $\mathcal{X}_f^{\text{scaled}} = \alpha \mathcal{X}_f$  (s.t. constraints satisfied)



### Disturbance Cancellation

**Approach** Model the disturbance, use the measurements and model to estimate the state and disturbance and find control inputs that use the disturbance estimate to remove offset.

### Augmented Model

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

**Constant disturbance**  $d_{k+1} = d_k$

Observable iff  $\begin{bmatrix} A-I & B_d \\ C & C_d \end{bmatrix}$  has full rank (assuming  $n_x = n_d$ )

### Observer For Augmented Model

$$\begin{bmatrix} \hat{x}_{k+1} \\ \hat{d}_{k+1} \end{bmatrix} = \begin{bmatrix} A & B_d \\ 0 & \mathbb{I} \end{bmatrix} \begin{bmatrix} \hat{x}_k \\ \hat{d}_k \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k + \begin{bmatrix} L_x \\ L_d \end{bmatrix} (C\hat{x}_k + C_d\hat{d}_k - y_k)$$

**Error Dynamics**  $\Rightarrow$  choose  $L$  s.t error dynamics converge to 0

$$\begin{bmatrix} x_{k+1} - \hat{x}_{k+1} \\ d_{k+1} - \hat{d}_{k+1} \end{bmatrix} = \left( \begin{bmatrix} A & B_d \\ 0 & \mathbb{I} \end{bmatrix} + \begin{bmatrix} L_x \\ L_d \end{bmatrix} \begin{bmatrix} C & C_d \end{bmatrix} \right) \begin{bmatrix} x_k - \hat{x}_k \\ d_k - \hat{d}_k \end{bmatrix}$$

**Lemma 1.** Steady-state of an asym. stable observer satisfies:

$$\begin{bmatrix} A - \mathbb{I} & B \\ C & 0 \end{bmatrix} \begin{bmatrix} \hat{x}_\infty \\ u_\infty \end{bmatrix} = \begin{bmatrix} -B_d \hat{d}_\infty \\ y_\infty - C_d \hat{d}_\infty \end{bmatrix} \text{ (for } n_y = n_d\text{)}$$

$\Rightarrow$  Observer output  $C\hat{x}_\infty + C_d\hat{d}_\infty$  tracks  $y_\infty$  without offset

### Reference Tracking with Disturbance Cancellation

**Goal** Track constant reference:  $Hy(k) = z(k) \rightarrow r, k \rightarrow \infty$

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

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

### Offset-free Tracking - Main Result

**Theorem 2.** Assuming RHC recursively feasible,  $n_d = n_y$ , unconstrained for  $k \geq j$ , and the closed loop system

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

converges, then  $z(k) = Hy(k) \rightarrow r$  as  $k \rightarrow \infty$

## 2.3 Soft Constraints

**Input constraints** are dictated by physical constraints on the actuators and are **usually hard**

**State/output constraints** arise from practical restrictions on the allowed operating range and are **rarely hard**

**Hard state/output constraints** always lead to **complications in the controller implementation**

### Soft Constrained MPC

$$\min_u \sum_{i=0}^{N-1} x_i^\top Q x_i + u_i^\top R u_i + l_\epsilon(\epsilon_i) + x_N^\top P x_N + l_\epsilon(\epsilon_N) \quad (2)$$

#### Constraints

##### Quadratic penalty

$$l_\epsilon(\epsilon_i) = \epsilon_i^\top S \epsilon_i \quad (\text{e.g } S = Q)$$

##### + linear norm penalty

$$l_\epsilon(\epsilon_i) = v|\epsilon_i|_{1/\infty}$$

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

$$H_x x_i \leq k_x + \epsilon_i$$

$$H_u u_i \leq k_u$$

$$\epsilon_i \geq 0 \text{ slack variable}$$

$$\begin{aligned} \text{Original} \quad & \min_z f(z) \\ \text{s.t. } & g(z) \leq 0 \end{aligned}$$

$$\begin{aligned} \text{Softened} \quad & \min_{z, \epsilon} f(z) + l_\epsilon(\epsilon) \\ \text{s.t. } & g(z) \leq \epsilon \\ & \epsilon \geq 0 \end{aligned}$$

**Requirement on  $l_\epsilon(\epsilon)$**  If the original problem has a feasible solution  $z^*$ , then the softened problem should have the same solution  $z^*$ , and  $\epsilon = 0$ .

**Theorem 3 (Exact Penalty Function).**  $l_\epsilon(\epsilon) = v \cdot \epsilon$  satisfies requirement for any  $v > \lambda^* \geq 0$ , where  $\lambda^*$  is optimal Lagrange multiplier for original problem

### 3 Robust MPC

**Uncertain System**  $x(k+1) = g(x(k), u(k), w(k); \theta)$

#### Robust Constraint Satisfaction

**Idea** Compute a set of tighter constraints such that if the nominal system meets these constraints, then the uncertain system will too. We then do MPC on the nominal system.

**Goal** Ensure constraints satisfied for the MPC sequence.

**Disturbance reachable set**  $\mathcal{F}_i = \bigoplus_{j=0}^{i-1} A^j \mathcal{W}$

#### Robust Open-Loop MPC

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

#### Closed Loop Robust MPC

**Idea** Separate the available control authority into two parts:

$$z(k+1) = Az(k) + Bv(k)$$

steers noise-free *nominal* system to origin

$$u_i = K(x_i - z_i) + v_i$$

compensates for deviations, i.e. a *tracking* controller, to keep the real trajectory close to the nominal system.

⇒ We fix the linear feedback controller  $K$  offline, and optimize over the nominal inputs  $\{v_0, \dots, v_{N-1}\}$  and nominal trajectory  $\{z_0, \dots, z_N\}$ , which results in a convex problem.

$$e_{i+1} = x_{i+1} - z_{i+1} = (A + BK)e_i + w_i$$

#### 3.1 Robust Constraint-Tightening MPC

#### Robust Constraint-Tightening MPC

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

$$\mathcal{F}_0 := 0 \quad F_i := \mathcal{W} \oplus A_K \mathcal{W} \oplus \dots A_K^{i-1} \mathcal{W},$$

$$\text{Control Law } u(k) = v_0^* + K(x(k) - z_0) = v_0^*$$

## 3.2 Robust Tube MPC

**Idea** Ignore noise and plan the nominal trajectory, bound maximum error at any time with RPI set  $\mathcal{E} : \epsilon_i \in \mathcal{E}, \epsilon_{i+1} \in \mathcal{E}$

Ideally  $\mathcal{E}$  is selected as the minimum RPI set  $F_\infty$

We know that the real trajectory stays 'nearby' the nominal one  $x_i \in z_i \oplus \mathcal{E}$  because we plan to apply the controller  $u_i = K(x_i - z_i) + v_i$  in the future.

(we won't actually do this, but it's a valid sub-optimal plan)

We must ensure that all possible state trajectories satisfy the constraints. This is now equivalent to ensuring that  $x_i \in z_i \oplus \mathcal{E}$  (address input constraints later)

What do we need to make this work?

### Compute the set $\mathcal{E}$ that the error will remain inside

Previously we wanted the **maximum robust invariant set**, or the largest set in which our terminal control law works.

We now want the **minimum robust invariant set**, or the smallest set that the state will remain inside despite the noise.

Modify constraints on nominal trajectory  $\{z_i\}$

$$x_i \in z_i \oplus \mathcal{E} = \{z_i + e | e \in \mathcal{E}\}$$

### Tube MPC

$$\text{Cost function } J(Z, V) := \sum_{i=0}^{N-1} l(z_i, v_i) + l_f(z_N)$$

$$\text{Feasible set } \mathcal{Z}(x_0) := \begin{cases} z_{i+1} &= Az_i + Bv_i \\ z_i &\in \mathcal{X} \ominus \mathcal{E} \\ v_i &\in \mathcal{U} \ominus K\mathcal{E} \\ z_N &\in \mathcal{X}_f \\ x_0 &\in z_0 \oplus \mathcal{E} \end{cases}$$

$$\text{Optimization } (V^*(x_0), Z^*(x_0)) = \underset{V, Z}{\operatorname{argmin}} \{J(Z, V) | (Z, V) \in \mathcal{Z}(x_0)\}$$

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

**Theorem 4** (Robust Invariance of Tube MPC). The set  $\mathcal{Z} := \{x | \mathcal{Z}(x) \neq \emptyset\}$  is a robust invariant set of the system  $x(k+1) = Ax(k) + B\mu_{\text{tube}}(x(k)) + w(k)$  subject to the constraints  $x, u \in \mathcal{X} \times \mathcal{U}$ .

**Theorem 5** (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 to the limit of the set  $\mathcal{E}$ .

### Tube MPC - Quick Summary

To implement tube MPC:

#### - Offline -

1. Stabilizing controller  $K$  so that  $A + BK$  is (Schur) stable
2. Compute the minimal robust invariant set  $E = F_\infty$  for the system  $x(k+1) = (A + BK)x(k) + w(k), w \in \mathcal{W}^1$
3. Compute tightened constraints  $\bar{\mathcal{X}} := \mathcal{X} \ominus \mathcal{E}, \bar{\mathcal{U}} := \mathcal{U} \ominus K\mathcal{E}$
4. Choose terminal weight function  $l_f$  and constraint  $\mathcal{X}_f$  satisfying assumptions\*

#### - Online -

1. Measure / estimate state  $x$
2. Solve optimization problem for  $(V^*(x_0), Z^*(x_0))$
3. Set the input to  $u = K(x - z_0^*(x)) + v_0^*(x)$

## 4 Invariance

**Definition 1** (Positively Invariant Set  $\mathcal{O}$ ). For an autonomous or closed-loop system, the set  $\mathcal{O}$  is positively invariant if:

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

**Definition 2** (Maximal Positively Invariant Set  $\mathcal{O}_\infty$ ). A set that contains all  $\mathcal{O}$  is the maximal positively invariant set  $\mathcal{O}_\infty \subset \mathcal{X}$

**Definition 3** (Pre-Sets). The set of states that in the dynamic system  $x(k+1) = g(x(k))$  in one time step evolves into the target set  $\mathcal{S}$  is the **pre-set** of  $\mathcal{S} \Rightarrow \text{pre}(\mathcal{S}) := \{x \mid g(x) \in \mathcal{S}\}$

**Lemma 2.** **Invariant Sets from Lyapunov Functions**

If  $V : \mathbb{R}^n \rightarrow \mathbb{R}$  is a Lyapunov function for  $x(k+1) = g(x(k))$ , then  $Y := \{x \mid V(x) \leq \alpha\}$  is an invariant set for all  $\alpha \geq 0$

*Proof.* Lyapunov property  $V(g(x)) - V(x) < 0$  implies that once  $V(x(k)) \leq \alpha$ ,  $V(x(j)) < \alpha, \forall j \geq k \rightarrow$  Invariance  $\square$

**Example System**  $x(k+1) = Ax(k)$ ,  $A^\top PA - P \prec 0 \prec P$  and resulting Lyapunov function  $V(x(k)) = x(k)^\top Px(k)$

**Goal** Find the largest  $\alpha$  s.t the invariant set  $Y_\alpha \in \mathcal{X}$

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

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

**Theorem 6** (Geometric condition for invariance). Set  $\mathcal{O}$  is positively invariant set iff  $\mathcal{O} \subseteq \text{pre}(\mathcal{O}) \Leftrightarrow \text{pre}(\mathcal{O}) \cap \mathcal{O} = \mathcal{O}$

*Proof.* **Necessary** if  $\mathcal{O} \not\subseteq \text{pre}(\mathcal{O})$ , then  $\exists \bar{x} \in \mathcal{O} \text{ s.t. } \bar{x} \notin \text{pre}(\mathcal{O}) \rightsquigarrow \bar{x} \in \mathcal{O}, \bar{x} \notin \text{pre}(\mathcal{O})$ , thus  $\mathcal{O}$  not positively invariant

**Sufficient** if  $\mathcal{O}$  not positive invariant set, then  $\exists \bar{x} \in \mathcal{O} \text{ s.t. } g(\bar{x}) \notin \mathcal{O} \rightsquigarrow \bar{x} \in \mathcal{O}, \bar{x} \notin \text{pre}(\mathcal{O})$  thus  $\mathcal{O} \not\subseteq \text{pre}(\mathcal{O})$   $\square$

### 4.1 Control Invariance

**Definition 4** (Control Invariant Set).  $\mathcal{C} \subseteq \mathcal{X}$  control invariant if

$$x(k) \in \mathcal{C} \Rightarrow \exists u(k) \in \mathcal{U} \text{ s.t. } g(x(k), u(k)) \in \mathcal{C} \quad \forall k$$

**Definition 5** (Maximal Control Invariant Set  $\mathcal{C}_\infty$ ). A set that contains all  $\mathcal{C}$  is the maximal positively invariant set  $\mathcal{C}_\infty \subset \mathcal{X}$

**Intuition** For all states in  $\mathcal{C}_\infty$  exists control law s.t constraints are never violated  $\rightsquigarrow$  **The best any controller could ever do**

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

Set  $\mathcal{C}$  is control invariant iff:  $\mathcal{C} \subseteq \text{pre}(\mathcal{C}) \Leftrightarrow \text{pre}(\mathcal{C}) \cap \mathcal{C} = \mathcal{C}$

#### Control Law from Control Invariant Set

Control law  $\kappa(x(k))$  will **guarantee** that the system with control invariant set  $\mathcal{C}$  satisfies constraints **for all time** if

$$x(k+1) = g(x(k), \kappa(x)) \rightarrow g(x, \kappa(x)) \in \mathcal{C} \quad \forall x \in \mathcal{C}$$

We can use this fact to **synthesize** control law  $\kappa$

$$\kappa(x) := \underset{u}{\operatorname{argmin}} \{f(x, u) \mid g(x, u) \in \mathcal{C}\}$$

with  $f$  as any function (including  $f(x, u) = 0$ )

Does not ensure that system will converge

Difficult because calculating control invariant sets is hard

**MPC** implicitly describes  $\mathcal{C}$  s.t easy to represent/compute

### 4.2 Robust Invariance

**Definition 6** (Robust Positive Invariant Set  $\mathcal{O}^W$ ). For the autonomous system  $x(k+1) = g(x(k), w(k))$ , the set  $\mathcal{O}^W$  is robust positive invariant if:

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

Given set  $\Omega$  and dynamic system  $x(k+1) = g(x(k), w(k))$ ,

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

**Definition 7** (Robust Pre-Sets). The set of states that in the dynamic system  $x(k+1) = g(x(k), w(k))$  for all disturbance  $w \in \mathcal{W}$  in one time step evolves into the target set  $\Omega$  is the **pre-set** of  $\Omega \Rightarrow \text{pre}^{\mathcal{W}}(\Omega) := \{x | g(x, w) \in \Omega \forall w \in \mathcal{W}\}$

**Theorem 7** (Geometric condition for robust invariance). Set  $\mathcal{O}^{\mathcal{W}}$  is robust positive invariant iff  $\mathcal{O}^{\mathcal{W}} \subseteq \text{pre}^{\mathcal{W}}(\mathcal{O}^{\mathcal{W}})$

### Minimum Robust Invariant Set

$$\mathcal{F}_{\infty} = \bigoplus_{j=0}^{\infty} A_K^j \mathcal{W}, \mathcal{F}_0 := \{0\} \Rightarrow \mathcal{F}_n = \mathcal{F}_{n+1} = \mathcal{F}_{\infty}$$

## 4.3 Computing Invariant Sets and Pre-sets

### Conceptual Algorithm

#### Pseudo algorithm

Inputs, outputs, especially pre-set calculation will be different, the overall concept not, used for:

#### Invariant Sets

#### Control Invariant Sets

#### Robust Invariant Sets

Replace Intersection with Minkowski Sum and Pre-Set with  $A^i \mathcal{W}$  to get an Idea for **Minimum Robust Invariant set**

### System for Pre-Set Computation

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ u(k) \in \mathcal{U} &:= \{u | Gu \leq g\} \\ \mathcal{S} &:= \{x | Fx \leq f\} \end{aligned}$$

### Invariant Pre-Set

$$\begin{aligned} \text{pre}(S) &:= \{x | Ax \in S\} \\ &= \{x | FAx \leq f\} \end{aligned}$$

### Control Invariant Pre-Set

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

This is a **projection** operation

### System for Robust Pre-Set Computation

$$\begin{aligned} x(k+1) &= Ax(k) + w(k) \\ \Omega &:= \{x | Fx \leq f\} \end{aligned}$$

### Robust Invariant Pre-Set

$$\begin{aligned} \text{pre}^{\mathcal{W}}(\Omega) &= \{x | FAx + Fw \leq f\} \\ &= \{x | FAx \leq f - \max_{w \in \mathcal{W}} Fw\} \\ &= \{x | FAx \leq f - h_{\mathcal{W}^i}(F)\} \end{aligned}$$

where  $h_{\mathcal{W}^i}(F)$  is the **support function**  $\sigma_{\mathcal{C}}(a) = \sup_{x \in \mathcal{C}} a^T x$

## 5 Optimization

Mathematical Optimization Problem	
<b>Decision variable</b> $x \in \mathbb{R}^n$ <b>Objective function</b> $f : \text{dom}(f) \rightarrow \mathbb{R}$ <b>Inequality constraints</b> $g_i \ (i \in \# \text{constraints})$ <b>Equality constraints</b> $h_i \ (i \in \# \text{constraints})$ <b>Feasible set</b> $\mathcal{X} := \{x   g_i(x) \leq 0, h_i(x) = 0\}$	<b>minimize</b> $f(x)$ <b>subject to:</b> $g_i(x) \leq 0$ $h_i(x) = 0$
<b>Feasible point</b> $x \in \text{dom}(f)$ with $g_i(x) \leq 0, h_i(x) = 0$ <b>Strictly feasible point</b> $x$ with strict inequality $g_i(x) < 0$ <b>Optimal value</b> $f^*$ (or $p^*$ ) = $\inf\{f(x)   g_i(x) \leq 0, h_i(x) = 0\}$ $f^* = +\infty$ : OP infeasible, $f^* = -\infty$ : OP unbound below <b>Optimizer</b> set: $\arg\min_{x \in \mathcal{X}} f(x) := \{x \in \mathcal{X}   f(x) = f^*\}$	
$x^*$ is a <b>Global Minimum</b> if $f(x^*) \leq f(x)$ $x^*$ is a <b>Local Minimum</b> if $\exists \epsilon > 0$ s.t. $f(x^*) \leq f(x) \forall x \in \mathcal{X} \cap B_\epsilon(x^*)$ , open ball with center $x^*$ and radius $\epsilon$	

### 5.1 Convex Sets

**Definition 8** (Convex Set). Set  $\mathcal{C}$  is convex if and only if

$$\theta x + (1 - \theta)y \in \mathcal{C}, \forall x, y \in \mathcal{C}, \forall \theta \in [0, 1]$$

**Intersection**  $\mathcal{C}_1, \mathcal{C}_2 \text{ cv} \Rightarrow \mathcal{C}_1 \cap \mathcal{C}_2 \text{ convex (cv)}$

**Image under affine map**  $\mathcal{C} \subseteq \mathbb{R}^n \text{ cv} \Rightarrow \{Ax + b \mid x \in \mathcal{C}\} \text{ cv}$

**Inverse loaM**  $\mathcal{C} \subseteq \mathbb{R}^m \text{ cv} \Rightarrow \{x \in \mathbb{R}^n \mid Ax + b \in \mathcal{C}\} \text{ cv}$

**Definition 9** (Hyperplanes).  $\{x \in \mathbb{R}^n \mid a^T x = b\}$

**Definition 10** (Halfspaces).  $\{x \in \mathbb{R}^n \mid a^T x \leq b\}$

can be **open** (strict inequality) or **closed** (non-strict inequality)

**Definition 11** (Polyhedra). intersection of **finite** number of closed halfspaces: polyhedra  $\{x \in \mathbb{R}^n \mid A^{q \times n}x \leq b^{q \times 1}\}$

**Definition 12** (Polytope). is a **bounded** polyhedron.

**Definition 13** (Convex hull). for  $\{v_1, \dots, v_k\} \in \mathbb{R}^d$  is:

$$\text{co}(\{v_1, \dots, v_k\}) := \{x | x = \sum_i \lambda_i v_i, \lambda \geq 0, \sum_i \lambda_i = 1\}$$

**Definition 14** (Ellipsoid). set:  $\{x | (x - x_c)^\top A^{-1}(x - x_c) \leq 1\}$  where  $x_c$  is center of ellipsoid,  $A \succ 0$  (i.e. positive definite)  
(Semi-axis lengths are square roots of eigenvalues of  $A$ )

**Definition 15** (Norm Ball).  $B_r(x) := \{\xi \in \mathbb{R}^n : |\xi - x|_p < r\}$  where  $p$  defines the  $l_p$  norm,  $p = \{1|2|\dots|\infty\}$

**Theorem 8.** Minkowski-Weyl

The following statements are equivalent for  $\mathcal{P} \subseteq \mathbb{R}^d$

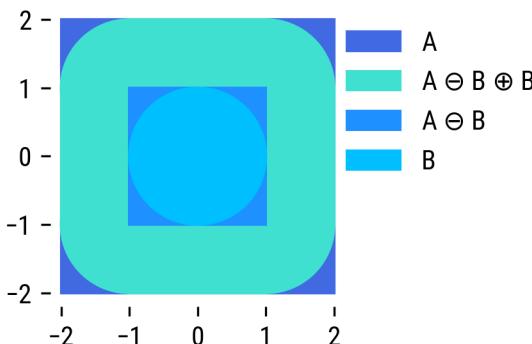
$\mathcal{P}$  is a polytope and there exists  $A, b$  s.t  $\mathcal{P} = \{x \mid Ax \leq b\}$

$\mathcal{P}$  finitely generated,  $\exists$  finite set  $\{v_i\}$  s.t  $\mathcal{P} = \text{co}(\{v_1, \dots, v_s\})$

**Definition 16.** **Minkowski Sum** for  $A, B \subset \mathbb{R}^n$

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

**Example**  $[a, b] \oplus [c, d] = [a + c, b + d]$



**Definition 17.** **Pontryagin Difference** for  $A, B \subset \mathbb{R}^n$

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

**Example**  $[a, b] \ominus [c, d] = [a - c, b - d]$

## 5.2 Convex Functions

**Definition 18** (Convex Function).  $f : \mathcal{C}_{\text{convex}} \rightarrow \mathbb{R}$  is convex iff

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y), \forall x, y \forall \theta \in [0, 1]$$

$f$  is strictly convex if this inequality is strict.

**Definition 19** (Epigraph).  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  cv  $\Leftrightarrow \text{epi}(f)$  is cv set

$$\text{epi}(f) := \{(x, t) \in \mathbb{R}^{n+1} | f(x) \leq t\}$$

**Check Convexity**  $f$  is convex if it is composition of simple convex function with convexity preserving operations or if

$f : \mathbb{R}^n \rightarrow \mathbb{R}$  twice differentiable,  $\partial^2 f / \partial x^2 \succeq 0 \forall x \in \mathbb{R}^n$

$g : \mathbb{R} \rightarrow \mathbb{R}$  with  $g(t) = f(x + tv)$  convex in  $t \forall x, v \in \mathbb{R}^n \rightarrow f$  convex (restriction to a line)

- the point wise maximum of convex functions is convex

- the sum of convex functions is convex

-  $f(Ax + b)$  is convex if  $f$  is convex

## 5.3 Optimality Conditions

### Lagrange Duality

$$\text{Consider } f^* = \inf_{x \in \mathbb{R}^n} f(x) \text{ s.t. } g(x) \leq 0, h(x) = 0 \quad (3)$$

$$\text{Lagrangian} \quad \mathcal{L}(x, \lambda, \nu) = f(x) + \lambda^T g(x) + \nu^T h(x)$$

$$\text{Dual Function} \quad d(\lambda, \nu) = \inf_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda, \nu)$$

**Proposition 1** (Weak Duality).  $d(\lambda, \nu) \leq f^*, \forall \lambda \geq 0, \nu \in \mathbb{R}^h$

**Definition 20** (Constraint qualification). **Slater's Condition** holds if  $\exists$  at least one **strictly feasible point**  $\hat{x}$  ( $h(\hat{x}) = 0, g(\hat{x}) < 0$ )

**Proposition 2** (Strong Duality). If Slater's condition holds and OP is convex  $\Rightarrow \exists \lambda \geq 0, \nu \in \mathbb{R}^{n_h}$  s.t.  $d(\lambda, \nu) = f^*$

### KKT Conditions (Karush-Kuhn-Tucker)

**Theorem 9** (KKT Conditions). If Slater's condition holds and (3) is convex  $\rightarrow x^* \in \mathbb{R}^n$  is a minimizer of the primal (3) and  $(\lambda^* \geq 0, \nu^*) \in \mathbb{R}^{n_g} \times \mathbb{R}^{n_h}$  is a maximizer of the dual  $\Leftrightarrow$  is equivalent to the following statements:

**KKT-1** (Stationary Lagrangian)  $\nabla_x \mathcal{L}(x^*, \lambda^*, \nu^*) = 0$

**KKT-2** (primal feasibility)  $g(x^*) \leq 0, h(x^*) = 0$

**KKT-3** (dual feasibility)  $\lambda^*, \nu^* \in \mathbb{R}^{n_h} \geq 0$

**KKT-4** (complementary slackness)  $\lambda^{*T} g(x^*) = 0$

slackness)  $\nu^{*T} h(x^*) = 0$

In addition we have:  $\sup_{\lambda \geq 0, \nu \in \mathbb{R}^{n_h}} q(\lambda, \nu) = \inf_{x \in \mathcal{C}} f(x)$

**Remark** Without Slater, KKT1-4 still implies  $x^*$  minimizes (3) and  $\lambda, \nu$  maximizes dual, but the converse is no longer true. There can be primal-minimizer/dual-maximizer not satisfy KKT.

## 5.4 Convex Optimization Problems

**Theorem 10.** For a convex optimization problem, **any** locally optimal solution is globally optimal (local optima are global optima).

**Linear Programming** minimize  $c^T x$  s.t.  $Ax - b \geq 0, x \geq 0$

Step 1:  $\mathcal{L}(x, \lambda_1, \lambda_2) = c^T x - \lambda_1^T (Ax - b) - \lambda_2^T x, \lambda_i \geq 0$

Step 2:  $\inf_{x \in \mathbb{R}^n} \mathcal{L} = \lambda_1^T b$ , if  $c - A^T \lambda_1 - \lambda_2 = 0$ , else  $-\infty$

Step 3: Dual, maximize  $b^T \lambda$  s.t.  $c - A^T \lambda \geq 0, \lambda \geq 0$  (again LP)

**Proposition 3.** The optimal solution of a linear program (if it exists) lies always on the boundary of the feasible set and there exists an optimal solution that is a vertex of the feasible set.

**Quadratic Programming** convex if  $P = P^T$  positive semi-definite minimize  $\frac{1}{2}x^T Px + q^T x$  s.t.  $Gx \leq h, Ax = b$

### Second-Order Cone Program

minimize  $f^T x$  s.t.  $|A_i x + b| \leq c_i^T x + d_i, Fx = g$

Second-order cone  $C_{n+1} = \{(x, t) \mid x \in \mathbb{R}^n, t \in \mathbb{R}, |x| \leq t\}$

$|A_i x + b| \leq c_i^T x + d_i \Leftrightarrow (A_i x + b, c_i^T x + d_i) \in C_{n+1}$

**Semi-Definite Programming** with symmetric  $F_i, X, A_i$

minimize  $c^T x$  s.t.  $\sum_{i=1}^n x_i F_i + G \preceq 0, Ax = b$

## 6 Introduction to Systems and Controls

**Idea** Create a model by solving the systems physical equations

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

(Exact Solution to ODE of a Linear System)

**Problem** Most physical systems are nonlinear

**Trick** use First Order Taylor expansion  $f(\bar{x}) + \left. \frac{\partial f}{\partial x^T} \right|_{\bar{x}} (x - \bar{x})$

### 6.1 Linearization

**Idea** Nonlinear system stable enough around an equilibrium

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

Find stationary operating point  $x_s, u_s$  and plug in derivative:

$$\begin{aligned} \Delta \dot{x} &= \dot{x} - \dot{x}_s \\ &= A^c \Delta x + B^c \Delta u \quad A^c = \left. \frac{\partial g}{\partial x^T} \right|_{\substack{x_s \\ u_s}} \quad B^c = \left. \frac{\partial g}{\partial u^T} \right|_{\substack{x_s \\ u_s}} \\ \Delta y &= y - y_s \\ &= C \Delta x + D \Delta u \quad C = \left. \frac{\partial h}{\partial x^T} \right|_{\substack{x_s \\ u_s}} \quad D = \left. \frac{\partial h}{\partial u^T} \right|_{\substack{x_s \\ u_s}} \end{aligned}$$

### 6.2 Discretization

For general nonlinear systems only approximate discretization methods exist, such as Euler, quality depends on sampling time

**Approximation**

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

**Notation**

$$\begin{aligned} x(k) &:= x^c(t_0 + kT_s) \\ u(k) &:= u^c(t_0 + kT_s) \end{aligned}$$

**Exact Discretization of Linear Time-Invariant Models**

$$\begin{aligned} x(t_{k+1}) &= \underbrace{e^{A^c T_s}}_{=A} x(t_k) + \underbrace{\int_0^{T_s} e^{A^c(T_s-\tau)} B^c d\tau}_{B=(A^c)^{-1}(A-I)B^c} u(t_k) \\ x(k+N) &= A^N x(k) + \sum_{i=0}^{N-1} A^i B u(k+N-1-i) \end{aligned}$$

### 6.3 Analysis of Discrete-Time LTI Systems

**Controllable** if  $\text{rank}(\mathcal{C}) = n$ ,  $\mathcal{C} = [B \quad \dots \quad A^{n-1}B]$

$\forall(x(0), x^*) \exists$  finite time  $N$  with inputs  $U$ , s.t.  $x(N) = x^*$

**Stabilizable** iff all uncontrollable modes stable

**Observable** if  $\text{rank}(\mathcal{O}) = n$ ,  $[C^\top \quad \dots \quad (CA^{n-1})^\top]^\top$

$\forall x(0) \exists$  finite time  $N$ , s.t. the measurements

$y(0), \dots, y(N-1)$  uniquely distinguish initial state  $x(0)$

**Detectable** iff all unobservable modes stable

## 7 Optimal Control

### Discrete-Time Optimal Control Problem

#### Cost Function

$$J(x_0, U) = \sum_{i=0}^{N-1} l(x_i, u_i) + l_f(x_N) \quad (4)$$

**Stage Cost**  $l_f(x_i, u_i)$  **Terminal Cost**  $l_f(x_N)$

#### Constraints

$$x_{i+1} = g(x_i, u_i)$$

$$x_0 = x(k)$$

$$h(x_i, u_i) \leq 0 \quad (\text{optional})$$

### Riccati Equations

**Riccati Difference Equation - RDE** solved recursively  $P_i = A^\top P_{i+1} A + Q - A^\top P_{i+1} B (B^\top P_{i+1} B + R)^{-1} B^\top P_{i+1} A$

**Algebraic Riccati Equation - ARE** solved analytically

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

### 7.1 Unconstrained Finite Horizon Control Problem

### Linear Quadratic Optimal Control

#### Cost Function

$$J^*(x(0)) := \min_U \sum_{i=0}^{N-1} x_i^\top Q x_i + u_i^\top R u_i + x_N^\top P x_N \quad (5)$$

#### Terminal weight

$P \succeq 0$  symmetric

#### State weight

$Q \succeq 0$  symmetric

#### Input weight

$R \succ 0$  symmetric

#### Constraints

No input or state constraints!

$$x(k+1) = Ax_k + Bu_k$$

Only dynamics matter.

#### Batch Approach

express cost function in terms of  $x(0)$  and input sequence  $U$

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

$$X := \mathcal{S}^x x(0) + \mathcal{S}^u U \quad J(x(0), U) = X^\top \bar{Q} X + U^\top \bar{R} U$$

$$\bar{Q} := \text{blockdiag}(Q, \dots, Q, P) \quad \bar{R} := \text{blockdiag}(R, \dots, R)$$

**Optimal Input set**  $\nabla_U J(x(0), U) = 2HU + 2F^\top x(0) = 0$

$$U^*(x(0)) = -\underbrace{\left((\mathcal{S}^u)^\top \bar{Q} \mathcal{S}^u + \bar{R}\right)}_{H(\text{Hessian})^{-1}} \underbrace{(\mathcal{S}^u)^\top \bar{Q} \mathcal{S}^x}_{F^\top} x(0)$$

**Optimal Cost** ( $x_0 = x(0)$ )

$$J^*(x_0) = x_0^\top (\mathcal{S}_x^\top \bar{Q} \mathcal{S}_x - \mathcal{S}_x^\top \bar{Q} \mathcal{S}_u (\mathcal{S}_u^\top \bar{Q} \mathcal{S}_u + \bar{R})^{-1} \mathcal{S}_u^\top \bar{Q} \mathcal{S}_x) x_0$$

### Recursive Approach

use dynamic programming to solve problem backwards from  $N$

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

**From Principle Of Optimality**

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

**Optimal Control Policy** use Riccati

$$u_i^* = F_i x_i = -(B^\top P_{i+1} B + R)^{-1} B^\top P_{i+1} A \cdot x(i)$$

**Optimal Cost-To-Go**

$$J_i^*(x_i) = x_i^\top P_i x_i$$

### Comparison of Batch and Recursive Approach

Dynamic programming yields **feedback policies**  $u_i^* = F_i x_i$  depending on each  $x_i$ . Batch optimization returns sequence  $U^*(x(0))$  of **numeric values** depending only on  $x(0)$ ,

## 7.2 Infinite Horizon Control Problem

### Linear Quadratic Regulator

**Cost Function**

$$J_\infty^*(x(k)) = \min_{u(\cdot)} \sum_{i=0}^{\infty} x_i^\top Q x_i + u_i^\top R u_i \quad (6)$$

**Constraints**

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

$$x_0 = x(k)$$

**Optimal Input**

$$u^*(k) = F_\infty x(k) \quad (\text{use Riccati})$$

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

**Lemma 3** (Lyapunov function for LQR). If  $(A, B)$  is stabilizable and  $Q, R \succ 0$  then  $J^*(x) = x^\top P_\infty x$  is a **Lyapunov function** for the system  $x^+ = (A + BF_\infty)x$  where  $F_\infty$  is the constant feedback matrix and  $P_\infty \succ 0$  solves the Riccati equation..

### Lyapunov

**Stability** is a property of an **equilibrium point**  $\bar{x}$  of a system

**Definition 21** (Lyapunov Stability).  $\bar{x}$  is **Lyapunov stable** if:

$$\forall \epsilon > 0 \exists \delta(\epsilon) \text{ s.t. } |x(0) - \bar{x}|_2 < \delta(\epsilon) \rightarrow |x(k) - \bar{x}|_2 < \epsilon$$

**Definition 22** (Globally asymptotic stability). If  $\bar{x}$  is attractive, i.e.,  $\lim_{k \rightarrow \infty} \|x(k) - \bar{x}\| = 0$ ,  $\forall x(0)$  and Lyapunov stable then  $\bar{x}$  is **globally asymptotically stable**.

**Definition 23** (Global Lyapunov function). For the equilibrium  $\bar{x} = 0$  of a system  $x(k+1) = g(x(k))$ , a function  $V$ , continuous at the origin, finite and such that

$\forall x \in \mathbb{R}^n$ :

$$|x| \rightarrow \infty \Rightarrow V(x) \rightarrow \infty$$

$$V(x) = 0 \text{ if } x = 0 \quad \text{else} \quad V(x) > 0$$

$$V(g(x)) - V(x) \leq -\alpha(x)$$

for continuous positive definite  $\alpha : \mathbb{R}^n \rightarrow \mathbb{R}$

then  $V : \mathbb{R}^n \rightarrow \mathbb{R}$  is called **Lyapunov function**.

**Theorem 11.** If a system admits a Lyapunov function  $V(x)$ , then  $\bar{x} = 0$  is **globally asymptotically stable**.

**Theorem 12** (Lyapunov indirect method). System linearized around  $\bar{x} = 0$  with resulting matrix  $A$  and eigenvalues  $\lambda_i$ .

If  $\forall |\lambda_i| < 1$  then the origin is asymptotically stable.

If  $\exists |\lambda_i| > 1$  then origin is unstable.

If  $\exists |\lambda_i| = 1$  we can't conclude anything about stability.

**Discrete-Time  
Lyapunov equation**

$$A^T P A - P = -Q, \quad Q > 0 \quad (7)$$

**Theorem 13** (Existence of solution, DT Lyapunov equation). The discrete-time Lyapunov equation has a unique solution  $P > 0$  iff the system  $x(k+1) = Ax(k)$  is stable.

### 7.3 Constrained Finite Time Optimal Control Problem

#### CFTOC

**Cost Function** equal to Linear Quadratic Optimal Control (5) **Constraints** as in (5)  $+ x_N \in \mathcal{X}_f, x_{i < N} \in \mathcal{X}, u_i \in \mathcal{U}$

#### Quadratic Cost CFTOC

$$J(x(k)) = x_N^\top P x_N + \sum_{i=0}^{N-1} x_i^\top Q x_i + u_i^\top R u_i$$

#### Goal: Transform into QP

$$\min_{z \in \mathbb{R}^n} \frac{1}{2} z^\top H z + q^\top z + r \quad \text{s.t. } Gz \leq h, \quad Az = b$$

#### Construction of QP without Substitution

**Idea** Keep state equations as equality constraints

**Define variable**  $z = [x_1^\top \dots x_N^\top \quad u_0^\top \dots u_{N-1}^\top]^\top$

**Equalities** from system dynamics  $x_{i+1} = Ax_i + Bu_i$

$$G_{eq} = \begin{bmatrix} \mathbb{I} & & \\ -A & \mathbb{I} & & \\ \ddots & \ddots & -B & \\ & -A & \ddots & -B \end{bmatrix} E_{eq} = \begin{bmatrix} A \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

**Inequalities**  $G_{in}z \leq w_{in} + E_{in}x(k)$  for  $\mathcal{X}, \mathcal{U}$

$$\mathcal{X} = \{x \mid A_x x \leq \mathcal{X}_f q b_x\} \quad \mathcal{U} = \{u \mid A_u u \leq b_u\} \quad \mathcal{X}_f = \{x \mid A_f x \leq b_f\}$$

$$E_{in} = \begin{bmatrix} -A_x \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$G_{in} = \left[ \begin{array}{cc|cc} 0 & & 0 & \\ \hline A_x & & 0 & \\ \ddots & & \ddots & \\ \hline & A_x & A_f & 0 & 0 \\ \hline 0 & & A_u & & \\ \ddots & & 0 & A_u & A_u \\ 0 & & 0 & & \end{array} \right] \quad w_{in} = \begin{bmatrix} b_x \\ b_x \\ \vdots \\ b_x \\ b_f \\ b_u \\ \vdots \\ b_u \\ b_u \end{bmatrix}$$

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

**Finally the resulting quadratic optimization problem**

$$\begin{aligned} J^*(x(k)) &= \min_z [z^\top x(k)^\top] \begin{bmatrix} \bar{H} & 0 \\ 0 & Q \end{bmatrix} [z^\top x(k)^\top]^\top \\ \text{s.t. } G_{in} z &\leq w_{in} + E_{in} x(k) \quad G_{eq} z = E_{eq} x(k) \end{aligned}$$

**Construction of QP with substitution**

**Idea** Substitute the state equations.

**Step 1** Rewrite cost as

$$\begin{aligned} J(x(k)) &= U^\top H U + 2x(k)^\top F U + x(k)^\top Y x(k) \\ &= [U^\top \quad x(k)^\top] \begin{bmatrix} H & F^\top \\ F & Y \end{bmatrix} [U^\top \quad x(k)^\top]^\top \end{aligned}$$

**Step 2** Rewrite constraints compactly as  $GU \leq w + Ex(k)$

$$G = \begin{bmatrix} A_u & 0 & \dots & 0 \\ 0 & A_u & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_u \\ 0 & 0 & \dots & 0 \\ A_x B & 0 & \dots & 0 \\ A_x A B & A_x B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_f A^{N-1} B & A_f A^{N-2} B & \dots & A_f B \end{bmatrix}, \quad E = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -A_x \\ -A_x A \\ -A_x A^2 \\ \vdots \\ -A_f A^N \end{bmatrix}, \quad w = \begin{bmatrix} b_u \\ b_u \\ \vdots \\ b_u \\ b_x \\ b_x \\ b_x \\ \vdots \\ b_f \end{bmatrix}$$

**Step 3** Rewrite constrained problem as

$$\begin{aligned} J^*(x(k)) &= \min_U [U^\top \quad x(k)^\top] \begin{bmatrix} H & F^\top \\ F & Y \end{bmatrix} [U^\top \quad x(k)^\top]^\top \\ \text{subj. to } GU &\leq w + Ex(k) \end{aligned}$$

### Choice of P

- 1 Match the infinite solution  $P_N = P_\infty$
- 2 Use solution of the Lyapunov Equation (7) if the system is asymptotically stable (otherwise P not positive definite), assumes no control needed after end of horizon.
- 3 Set  $P = 0$  und use instead constraint  $x_{i+N} = 0$

## 8 Implementation, needed?

CFTOC problem is **multiparametric quadratic program** mp-QP

Two options to solve it:

- Iterative optimization methods
  - Explicit solution

Let  $I := 1, \dots, m$  be the set of constraint indices.

**Definition 24** (Active Set).  $A(x)$  and its complement  $NA(x)$

$$A(x) := \{j \in I : G_j z^*(x) - S_j x = w_j\}$$

$$NA(x) := \{j \in I : G_j z^*(x) - S_j x < w_j\}$$

**Definition 25** (Critical Region).  $CR_A$  is set of parameters  $x$  for which set  $A \subseteq I$  of constraints  $i$  active at the optimum. For given  $\bar{x} \in \mathcal{K}^*$  let  $(A, NA) := (A(\bar{x}), NA(\bar{X}))$ . Then

$$CR_A := \{x \in \mathcal{K}^*: A(x) = A\} \quad (\text{states share active set})$$

## Online evaluation: Point location

## Sequential search

## Logarithmic search

## Large Scale Convex Optimization

**Definition 26** (Lipschitz continuity).  $g: \mathbb{R}^n \rightarrow \mathbb{R}^m$  is Lipschitz with constant  $L$  if:  $|g(x) - g(y)| \leq L|x - y| \forall x, y \in \mathbb{R}^m$

$f$  is **Lipschitz** (Lip) with constant  $L \Leftrightarrow |\nabla f(x)|_2 \leq L$

**Theorem 14.**  $C \subseteq \mathbb{R}^n$  non-empty closed (**cl**) convex set,  $y \notin C \rightarrow \exists a \neq 0, b \in \mathbb{R}$  s.t.  $a^T x + b < a^T y + b, \forall x \in C$

**Definition 27** (smoothness).  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is  $L$ -smooth ( **$L$ -sm**) if  $\nabla f(x)$  satisfies  $|\nabla f(x) - \nabla f(y)| \leq L|x - y| \forall x, y \in \mathbb{R}^n$

$$\text{Taylor} \rightarrow f(y) \leq f(x) + \nabla f(x)^T(y - x) + \frac{L}{2}|x - y|^2$$

**Definition 28** (strong convexity).  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is  $\mu$ -strongly convex ( $\mu$ -scv) if  $f(y) \geq f(x) + \nabla f(x)^T(y - x) + \frac{\mu}{2}\|x - y\|^2$

**How to find**  $\mu/L$ , Spectra of Hessian  $\nabla^2 f$ , min/max eigenvalues

## Gradient Descent

$x_{k+1} = x_k - T \nabla f(x_k)$  for  $k = (k_0, \dots, k_N)$  given  $x_0, T$

**Line search** optimal step  $\nu_t^* = \operatorname{argmin}_{\nu \in \mathbb{R}} f(x_t - \nu \nabla f(x_t))$

**Definition 29.**  $\text{prox}_{\mathcal{C}}(x) = \operatorname{argmin}_{y \in \mathcal{C}} \frac{1}{2} \|x - y\|^2$   $\mathcal{C} \subset \mathbb{R}^n$

**Lemma 4.**  $\text{cl. cv } \mathcal{C} \subset \mathbb{R}^n \rightarrow |\text{prox}_{\mathcal{C}}(x) - \text{prox}_{\mathcal{C}}(y)| \leq \|x - y\| \leftarrow |\text{prox}_{\mathcal{C}}(x) - \text{prox}_{\mathcal{C}}(y)|^2 \leq (\text{prox}_{\mathcal{C}}(x) - \text{prox}_{\mathcal{C}}(y))^T(x - y)$

## Projected Gradient Descent

$$x_{k+1} = \text{prox}_{\mathcal{C}}(x_k - T \nabla f(x_k)), \text{ for } x_0, k_0 \in \mathbb{N}, T \in (0, 2/L)$$

```

def get_next_u(y: Measurement, r: Reference):
    """
        System handler for offset-free tracking
    """
    # approximate state, disturbance
    x, d = estimator(y)
    # find steady state und generate delta
    x_s, u_s = target_selector(x, r, d)
    x_delta = x - x_s
    # call solver with new parameter
    u_delta = mpc_regulator(x_delta, x_s, u_s)
    u = u_delta + u_s

    return u

```

## TRANSFORM P-NI

## $\ell_\infty$ -Minimization

$$\min_{x \in \mathbb{R}^m} \|x\|_\infty \iff \begin{array}{l} \min_{x,t} \\ \text{subj. to } Fx \leq q \\ -1 \mathbf{1}_m t \leq x \leq \mathbf{1}_m t, Fx \leq q \end{array}$$

**$\ell_1$ -Minimization**

$$\begin{array}{ll} \min_{x \in \mathbb{R}^m} ||x||_1 & \iff \\ \text{subj. to } Fx \leq g & \min_{x \in \mathbb{R}^m, t \in \mathbb{R}^m} \mathbf{1}_m^\top t \\ & \text{subj. to } -t \leq x \leq t, Fx \leq g \end{array}$$