Model Predictive Control

Silvan Stadelmann - 3. August 2025 - v0.3.0

github.com/silvasta/summary-mpc

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

1.1 Exact ODE Solution of a Linear System

Idea Create a model by solving the systems physical equations

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

Problem Most physical systems are nonlinear

Trick First Order Taylor expansion $f(\bar{x}) + \frac{\partial f}{\partial x^\top}\Big|_{\bar{x}} (x-\bar{x})$

1.2 Linearization

Idea Nonlinear system stable enough around an equilibrium

$$\dot{x_s} = g(x_s, u_s) = 0 \qquad A^c = \frac{\partial g}{\partial x^T}\Big|_{\substack{x_s\\u_s}}$$

$$y_s = h(x_s, u_s) \qquad B^c = \frac{\partial g}{\partial u^T}\Big|_{\substack{x_s\\u_s}}$$

$$\begin{aligned}\Delta\dot{x} &= \dot{x} - \dot{x_s}\\ &= A^c\Delta x + B^c\Delta u\end{aligned} \qquad C = \frac{\partial h}{\partial x^T}\Big|_{\substack{x_s\\u_s}}$$

$$\begin{aligned}\Delta y &= y - y_s\\ &= C\Delta x + D\Delta u\end{aligned} \qquad D = \frac{\partial h}{\partial u^T}\Big|_{\substack{x_s\\u_s}}$$

1.3 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

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

Exact Discretization of Linear Time-Invariant Models

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

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

1.4 Analysis of LTI Discrete-Time Systems

Controllabe if rank$(\mathcal{C}) = n,\ \mathcal{C} = \begin{bmatrix} B & \cdots & A^{n-1}B\end{bmatrix}$

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

Stabilizable iff all uncontrollable modes stable

Observable if rank$(\mathcal{O}) = n,\ \begin{bmatrix} C^\top & \cdots & (CA^{n-1})^\top\end{bmatrix}^\top$

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

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

Detectable iff all unobservable modes stable

Lyapunov

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

Definition 1 (Lyapunov Stability). $\bar{\mathbf{x}}$ is **Lyapunov stable** if:
$\forall\, \epsilon > 0\, \exists\, \delta(\epsilon)$ s.t. $|x(0) - \bar{x}|_2 < \delta(\epsilon) \rightarrow |x(k) - \bar{x}|_2 < \epsilon$

Definition 2 (Globally asymptotic stability). If $\bar{\mathbf{x}}$ is attractive, i.e., $\lim_{k\to\infty} ||x(k) - \bar{x}|| = 0,\ \forall x(0)$ and Lyapunov stable then $\bar{\mathbf{x}}$ is **globally asymptotically stable**.

Definition 3 (Global Lyapunov function). For the equilibrium $\bar{\mathbf{x}} = 0$ 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 1. If a system admits a Lyapunov function $V(x)$, then $\bar{\mathbf{x}} = 0$ is **globally asymptotically stable**.

Theorem 2 (Lyapunov indirect method). System linearized around $\bar{\mathbf{x}} = 0$ with resulting matrix A and eigenvalues λ_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 $\qquad A^T P A - P = -Q, \quad Q > 0$

Theorem 3 (Existence of solution, DT Lyapunov equation). 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.

1.6 Optimization

Mathematical Optimization Problem

Decision variable $x \in \mathbb{R}^n$
Objectivce function $f : \text{dom}(f) \to \mathbb{R}$
Inequality constraints g_i $(i\in\#\text{constraints})$
Equality constraints h_i $(i\in\#\text{constraints})$
Fesabile set $\mathcal{X}{:=}\{x|g(x){\le}0, h(x){=}0\}$

minimize $f(x)$
subject to:
$g_i(x) \le 0$
$h_i(x) = 0$

Feasible point $x \in \text{dom}(f)$ with $g_i(x) \le 0,\ h_i(x) = 0$
Strictly feasible point x with strict inequality $g_i(x) < 0$
Optimal value $f^\star$ (or $p^\star$) $= \inf\{f(x)|g_i(x) \le 0, h_j = 0\}$
$f^\star = +\infty$: OP infeasible, $f^\star = -\infty$: OP unbound below
Optimizer set: $\text{argmin}_{x\in\mathcal{X}} f(x) := \{x \in \mathcal{X}|f(x) = f^\star\}$

$x^\star$ is a **Global Minimum** if $f(x^\star) \le f(x)$
$x^\star$ is a **Local Minimum** if $\exists\, \epsilon > 0$ s.t. $f(x^\star) \le f(x)$ $\forall x \in \mathcal{X} \cap B_\epsilon(x^\star)$, open ball with center $x^\star$ and radius ϵ

1.7 Convex Sets

Definition 4 (Convex Set). Set $\mathcal{C}$ is convex if and only if

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

Definition 5 (Hyperplanes). $\{x \in \mathbb{R}^n \mid a^\mathsf{T} x = b\}$
Definition 6 (Halfspaces). $\{x \in \mathbb{R}^n \mid a^\mathsf{T} x \le b\}$
can be **open** (strict inequality) or **closed** (non-strict inequality)
Definition 7 (Polyhedra). intersection of **finite** number of closed halfspaces: polyhedra $\{x \in \mathbb{R}^n \mid A^{q\times n} x \preceq b^{q\times 1},\}$
Definition 8 (Polytope). is a **bounded** polyhedron.
Definition 9 (Convex hull). for $\{v_1, \dots, v_k\} \in \mathbb{R}^d$ is:
$\text{co}(\{v_1, \dots, v_k\}) := \{x | x = \sum_i \lambda_i v_i, \lambda \ge 0, \sum_i \lambda_i = 1\}$
Definition 10 (Ellipsoid). set: $\{x|(x-x_c)^\mathsf{T} A^{-1}(x-x_c) \le 1\}$ 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 11 (Norm Ball). $B_r(x) := \{\xi \in \mathbb{R}^n : |\xi - x|_p < r\}$ where p defines the l_p norm, $p = \{1|2|..|\infty\}$
Intersection $\mathcal{C}_1, \mathcal{C}_2$ cv $\Rightarrow \mathcal{C}_1 \cap \mathcal{C}_2$ convex **(cv)**
Image under affine map $\mathcal{C} \subseteq \mathbb{R}^n$ cv $\Rightarrow \{Ax + b \mid x \in \mathcal{C}\}$ cv
Inverse loaM $\mathcal{C} \subseteq \mathbb{R}^m$ cv $\Rightarrow \{x \in \mathbb{R}^n \mid Ax + b \in \mathcal{C}\}$ cv

1.8 Convex Functions

Definition 12 (Convex Function). $f : \mathcal{C}_{\text{convex}} \to \mathbb{R}$ is convex iff

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

f is strictly convex if this inequality is strict.
Definition 13 (Epigraph). $f : \mathbb{R}^n \to \mathbb{R}$ cv $\Leftrightarrow$ epi(f) is cv set

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

Check Convexity f is convex if it is composition of simple convex function with convexity preserving operations or if
$f : \mathbb{R}^n \to \mathbb{R}$ twice differentiable, $\partial^2 f/\partial x^2 \succeq 0\ \forall\, x \in \mathbb{R}^n$
$g : \mathbb{R} \to \mathbb{R}$ with $g(t) = f(x + tv)$ convex in $t\ \forall\, x, v \in \mathbb{R}^n$ $\to f$ 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

1.9 Convex Optimization

Theorem 4. For a convex optimization problem, **any** locally optimal solution is globally optimal (local optima are global optima).
Linear Programming minimize $c^\mathsf{T} x$ s.t. $Ax - b \ge 0,\ x \ge 0$
Step 1: $\mathcal{L}(x, \lambda_1, \lambda_2) = c^\mathsf{T} x - \lambda_1^\mathsf{T}(Ax - b) - \lambda_2^\mathsf{T} x,\ \lambda_i \ge 0$
Step 2: $\inf_{x\in\mathbb{R}^n} \mathcal{L} = \lambda_1^\mathsf{T} b$, if $c - A^\mathsf{T}\lambda_1 - \lambda_2 = 0$, else $-\infty$
Step 3: Dual, maximize $b^\mathsf{T}\lambda$ s.t. $c - A^\mathsf{T}\lambda \ge 0, \lambda \ge 0$ (again LP)
Quadratic Programming min ...

1.10 Optimality Conditions

Lagrange Duality

Consider $f^\star = \inf_{x\in\mathbb{R}^n} f(x)$ s.t. $g(x) \le 0, h(x) = 0$ (1)

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

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

Proposition 1 (Weak Duality). $d(\lambda, \nu) \le f^\star, \forall \lambda \ge 0, \nu \in \mathbb{R}^h$
Definition 14 (Constraint qualification). **Slaters 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 $\Rightarrow \exists \lambda \ge 0, \nu \in \mathbb{R}^{n_h}$ s.t. $d(\lambda, \nu) = f^\star$

KKT Conditions (Karush-Kuhn-Tucker)

Theorem 5 (KKT Conditions). If Slater's condition holds and (1) is convex $\to x^\star \in \mathbb{R}^n$ is a minimizer of the primal (1) and $(\lambda^\star \ge 0, \nu^\star) \in \mathbb{R}^{n_g} \times \mathbb{R}^{n_h}$ is a maximizer of the dual $\Leftrightarrow$ is equivalent to the following statements:

KKT-1 (Stationary Lagrangian) $\nabla_x \mathcal{L}(x^\star, \lambda^\star, \nu^\star) = 0$
KKT-2 (primal feasibility) $g(x^\star) \le 0, h(x^\star) = 0$
KKT-3 (dual feasibility) $\lambda^\star, \nu^\star \in \mathbb{R}^{n_h} \ge 0$
KKT-4 (compementary slackness) $\lambda^{\star T} g(x^\star) = 0$, $\nu^{\star T} h(x^\star) = 0$

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

Remark Without Slater, KKT1-4 still implies $x^\star$ minimizes (1) and λ, ν maximizes dual, but the converse is no longer true. There can be primal-minimizer/dual-maximizer not satisfy KKT.

1.11 Invariance

Definition 15 (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 16 (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 17 (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 $\mathcal{S} \Rightarrow$ pre$(\mathcal{S}) := \{x \mid g(x) \in \mathcal{S}\}$
Theorem 6 (Geometric condition for invariance). Set $\mathcal{O}$ is positively invariant set iff $\mathcal{O} \subseteq$ pre$(\mathcal{O}) \Leftrightarrow$ pre$(\mathcal{O}) \cap \mathcal{O} = \mathcal{O}$

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

1.12 Computing Invariant Sets

$\Omega_0 \leftarrow \mathcal{X}$
loop
  $\Omega_{i+1} \leftarrow$ pre$(\Omega_i) \cap \Omega_i$
  if $\Omega_{i+1} = \Omega_i$ **then**
    return $\mathcal{O}_\infty = \Omega_i$
  end if
end loop

(Same but much harder for control invariat sets)

System for Pre-Set Computation

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

Invariant Pre-Set

$$\text{pre}(\mathcal{S}) := \{x \mid Ax \in \mathcal{S}\} = \{x \mid FAx \le f\}$$

Control Invariant Pre-Set

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

This is a **projection** operation

1.13 Control Invariance

Definition 18 (Control Invariant Set). $\mathcal{C} \subseteq \mathcal{X}$ control invariant if

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

Definition 19 (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 pre$(\mathcal{S}) := \{x \mid \exists u \in \mathcal{U}$ s.t $g(x, u) \in \mathcal{S}\}$
Set $\mathcal{C}$ is control invariant iff: $\mathcal{C} \subseteq$ pre$(\mathcal{C}) \Leftrightarrow$ 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

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

We can use this fact to **synthesize** control law κ

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

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

Theorem 7. 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 $\mathcal{P} = \{x \mid Ax \le b\}$
$\mathcal{P}$ finitely generated, $\exists$ finite set $\{v_i\}$ s.t $\mathcal{P} = \text{co}(\{v_1, \dots, v_s\})$

MOST COMMON Polytopic

1

Lemma 1. Invariant Sets from Lyapunov Functions
If $V : \mathbb{R}^n \to \mathbb{R}$ is a Lyapunov function for $x(k+1) = g(x(k))$, then $Y := \{x \mid V(x) \le \alpha\}$ is an invariant set for all $\alpha \ge 0$

Proof. Lyapunov property $V(g(x)) - V(x) < 0$ implies that once $V(x(k)) \le \alpha, V(x(j)) < \alpha, \forall j \ge k \to$ Invariance □

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 α s.t the invariant set $Y_\alpha \in \mathcal{X}$

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

Equivalent to $\max_\alpha \alpha$ s.t. $h_{Y_\alpha}(F_i) \le f_i\ \forall i \in \{1 \dots n\}$
...
USE ELIPSOID

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

1.14 Optimal Control

Unconstrained Finite Horizon Control Problem

$$J^\star(x(0)) := \min_U x_N^\top Px_N + \textstyle\sum_{i=0}^{N-1} x_i^\top Qx_i + u_i^\top Ru_i$$
$$\text{subject to } x_{i+1} = Ax_i + Bu_i \quad i = 0, \dots, N-1$$
$$x_0 = x(0)$$

$P \succeq 0$, with $P = P^T$ terminal weight
$Q \succeq 0$, with $Q = Q^T$ state weight
$R \succ 0$, with $R = R^T$ input weight

1.15 Batch Approach

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

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

$\overline{Q} :=$ blockdiag$(Q, \dots, Q, P)$ $\quad \overline{R} :=$ blockdiag$(R, \dots, R)$
Optimal Input (from $\nabla_U J(x(0), U) = 2HU{+}2F^\top x(0) = 0$)

$$U^\star(x(0)) = -\big(\underbrace{(\mathcal{S}^u)^\top \overline{Q}\mathcal{S}^u + \overline{R}}_{H\text{(Hessian)}^{-1}}\big)\underbrace{(\mathcal{S}^u)^\top \overline{Q}\mathcal{S}^x}_{F^\top} x(0)$$

Optimal Cost

$J^\star(x(0))=x(0)^\top(\mathcal{S}_x^\top\overline{Q}\mathcal{S}_x-\mathcal{S}_x^\top\overline{Q}\mathcal{S}_u(\mathcal{S}_u^\top\overline{Q}\mathcal{S}_u+\overline{R})^{-1}\mathcal{S}_u^\top\overline{Q}\mathcal{S}_x)x(0)$

1.16 Recursive Approach

uses dynamic programming to solve problem backwards from N

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

From Principle Of Optimality

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

Optimal Cost-To-Go

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

Optimal Control Policy

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

1.17 Comparison of Batch and Recursive Approaches

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

1.18 Infinite Horizon LQR

LQR

$$J_\infty^\star(x(k)) = \min \sum_{i=0}^{\infty} x_i^\top Q x_i + u_i^\top R u_i$$

subj. to $x_{i+1} = Ax_i + Bu_i, \quad x_0 = x(k)$

Same u as for finite problem but with ARE Constant Feedback Matrix $F\infty$ asymptotically stable for.. Q,R,stabi,detect

Choice of P

1. Match infinite solution, use ARE
2. Assume no control needed after N, use Lyapunov Equation (makes only sense when asymptotically stable, otherwise P not positive definite)
3. set constraint $x_{i+N} = 0$

1.19 CFTOC

CFTOC Constrained Finite Time Optimal Control problem

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

$$J^\star(x(k)) = \min_U I_f(x_N) + \sum_{i=0}^{N-1} I(x_i,u_i)$$

s.t $x_{i+1} = Ax_i + Bu_i,\ i = 0,\dots,N-1$
$x_i \in \mathcal{X},\ u_i \in \mathcal{U},\ \mathcal{X}_N \in \mathcal{X}_f,\ x_0 = x(k)$

N is the time horizon and X , U, Xf are polyhedral regions

1.20 Transform Quadratic Cost CFTOC into QP

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

1.20.1 Construction of QP without substitution

Idea Keep state equations as equality constraints

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

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

$$G_{eq} = \left[\begin{array}{cccc|ccc} \mathbb{I} & & & & -B & & \\ -A & \mathbb{I} & & & & \ddots & \\ & \ddots & \ddots & & & & \\ & & -A & \mathbb{I} & & & -B \end{array}\right] \quad E_{eq} = \begin{bmatrix} A \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Inequalities $G_{in} z \le w_{in} + E_{in} x(k)$ from $\mathcal{X} = \{x \mid A_x x \le b_x\}, \mathcal{U} = \{u \mid A_u u \le b_u\}, \mathcal{X}_f = \{x \mid A_f x \le b_f\}$

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

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

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

Finally the resulting quadratic optimization problem

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

1.20.2 Construction of QP with substitution

Idea Substitute the state equations.

Step 1 Rewrite cost as

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

Step 2 Rewrite constraints compaclty as $GU \le w + Ex(k)$

Step 3 Rewrite constrained problem as

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

2 Nominal MPC

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

MPC Mathematical Formulation

V1

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

subj. to $x_{i+1} = Ax_i + Bu_i$
$x_0 = x(k) \quad x_i \in \mathcal{X}, \quad u_i \in \mathcal{U}$

Stability of MPC - Main Result

Theorem 8. The closed-loop system under the MPC control law $u_0^\star(x)$ is asymptotically stable and the set $\mathcal{X}_f$ is positive invariant for the system $x(k+1) = Ax(k) + Bu_0^\star(x(k))$ under the following assumptions:
1. Stage cost is positive definite, i.e. it is strictly positive and only zero at the origin
2. Terminal set is **invariant** under the local control law $\kappa_f(x_i)$:

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

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

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

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

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

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!

3 Practical MPC

3.1 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. $x_s = Ax_s + Bu_s$

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

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

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

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

3.2 Reference Tracking

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

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

Assume target feasible 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

- $\mathcal{X}_f \subseteq \mathcal{X}, Kx \in \mathcal{U} \quad \forall x \in \mathcal{X}_f$
- $V_f(x(k+1)) - V_f(x(k)) \le -l(x(k), Kx(k)) \quad \forall x \in \mathcal{X}_f$

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 CL system converges to target reference

$$x(k) \to x_s, z(k) = Hx(k) \xrightarrow{k\to\infty} r$$

Proof.
- Invariance under local ctrol law inherited from regulation case
- Constraint satisfaction provided by extra conditions
  - $x_s \oplus \mathcal{X}_f \subseteq \mathcal{X} \to x \in \mathcal{X} \forall \Delta \in \mathcal{X}_{\{}$
  - $K\Delta x + u_s \in \mathcal{U} \forall \Delta x \in \mathcal{X}_f \to u \in \mathcal{U}$
- Fron asympt stability of the regulation problem: $\Delta x(k) \xrightarrow{k\to\infty} 0$

□

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

3.3 Reference Tracking without Offset

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

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 2. Steady-state of an asym. stable observer satisfies:

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

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

```
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
```

3.4 Offset-free Tracking

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

$$\begin{aligned} x_s &= Ax_s + Bu_s + B_d\hat{d}_\infty \\ z_s &= H(Cx_s + C_d\hat{d}_\infty) = r \end{aligned} \quad \begin{bmatrix} A-I & B \\ HC & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r - HC_d\hat{d} \end{bmatrix}$$

Theorem 9 (Offset-free Tracking: Main Result). Assuming $n_d = n_y$, RHC recursively feasible, unconstrained for $k \geq j$, control law $\kappa(\cdot) = \kappa(\hat{x}(k), \hat{d}(k), r)$ and closed loop system

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

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

3.5 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 Problem Setup

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

$$\begin{aligned} \text{subj. to } & x_{i+1} = Ax_i + Bu_i \\ & H_x x_i \leq k_x + \epsilon_i \\ & H_u u_i \leq k_u \\ \text{slack variable } & \epsilon_i \geq 0 \end{aligned}$$

Quadratic penalty $l_\epsilon(\epsilon_i) = \epsilon_i^\top S \epsilon_i$ (e.g $S = Q$)

Linear Penalty $v|\epsilon_i|_{1/\infty}$

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

Theorem 10 (Exact Penalty Funtcion). $l_\epsilon(\epsilon) = v \cdot \epsilon$ satisfies requirement for any $v > \lambda^\star \geq 0$, where $\lambda^\star$ is optimal Lagrange multiplier for original problem

4 Robust MPC

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

4.1 Robust Invariance

Definition 20 (Robust Positive Invariant Set $\mathcal{O}^\mathcal{W}$). For the autonomous system $x(k+1) = g(x(k), w(k))$, the set $\mathcal{O}^\mathcal{W}$ is robust positive invariant if:

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

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

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

Definition 21 (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 Ω is the **pre-set** of $\Omega \Rightarrow \text{pre}^\mathcal{W}(\Omega) := \{x | g(x, w) \in \Omega \, \forall w \in \mathcal{W}\}$

Computing Robust Pre-Sets for Linear Systems

System $Ax(k) + w(k)$, set $\Omega := \{x \mid Fx \leq f\}$

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

where $h_{\mathcal{W}^i}(F)$ is the support function

Theorem 11 (Geometric condition for robust invariance). Set $\mathcal{O}^\mathcal{W}$ is robust positive invariant iff $\mathcal{O}^\mathcal{W} \subseteq \text{pre}^\mathcal{W}(\mathcal{O}^\mathcal{W})$

Definition 22 (Minkowski Sum). For $A, B \subset \mathbb{R}^n$, the Minkowski Sum is $A \oplus B := \{x + y | x \in A, y \in B\}$

Definition 23 (Pontryagin Difference). For $A, B \subset \mathbb{R}^n$, the Pontryagin Difference is $A \ominus B := \{x | x + e \in A, \forall e \in B\}$

4.2 Impact of Bounded Additive Noise

Defining a Cost to Minimize Expected value, worst case, max W nominal case w=0

4.3 Robust Constraint Satisfaction

The 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 that constraints are satisfied for the MPC sequence.

Terminal State Constraint

...is called disturbance reachable set,

4.4 Robust open loop MPC

$$\begin{aligned} \min_U \; & \left[l_f(x_N) + \textstyle\sum_{i=0}^{N-1} l(x_i, u_i) \right] \\ \text{subj. to } & x_{i+1} = Ax_i + Bu_i \\ & x_i \in \mathcal{X} \ominus (\textstyle\bigoplus_{j=0}^{i-1} A^j \mathcal{W}), \quad u_i \in \mathcal{U} \\ & x_0 = x(k), \quad x_N \in \mathcal{X}_f \ominus (\textstyle\bigoplus_{j=0}^{N-1} A^j \mathcal{W}) \end{aligned}$$

4.5 Robust closed loop MPC

Idea Separate the available control authority into two parts:

1. $z(k+1) = Az(k) + Bv(k)$ steers noise-free *nominal* system to origin

2. $u_i = K(x_i - z_i) + v_i$ compensates for deviations, i.e. a *tracking* controller, to keep the real trajectory close to the nominal system.

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

Minimum Robust Invariant Set

$F_\infty = \bigoplus_{j=0}^\infty A_K^j \mathcal{W}, F_0 := \{0\} \Rightarrow F_n = F_{n+1} = F_\infty$

4.6 Robust Constraint-Tightening MPC

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

$$\mathcal{F}_i := \mathcal{W} \oplus A_K \mathcal{W} \oplus \ldots A_K^i \mathcal{W}$$

Control Law $u(k) = v_0^\star + K(x(k) - z_0) = v_0^\star$

Motivation can robustly ensure constraint satisfactkon at each time step

Note need terminal set $\mathcal{X}_f$ that is robust invariant under tube controller K

4.7 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_∞

We know that the real trajectory stays 'nearby' the nominal onebecause we plan to apply the controllerin 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 (address input constraints later)

What do we need to make this work?

- Compute the set 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 $\{z_i\}$

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

- Formulate as convex optimization problem

BOX

... and then prove that

- Constraints are robustly satisfied

- The closed-loop system is robustly stable

Tube MPC

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

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

$$\text{Optimization:} \quad (V^\star(x_0), Z^\star(x_0)) = \text{argmin}_{V,Z}\{J(Z, V) | (Z, V) \in \mathcal{Z}(x_0)\}$$

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

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

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

$$\text{Optimization:} \quad (V^\star(x_0), Z^\star(x_0)) = \text{argmin}_{V,Z}\{J(Z, V) | (Z, V) \in \mathcal{Z}(x_0)\}$$

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

ASSUMPTIONS

Theorem 12 (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 13 (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 $\mathcal{E}$.

Putting it all together: Tube MPC

To implement tube MPC:

— Offline —

1. Choose a 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 the 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 the problem $(V^\star(x_0), Z^\star(x_0)) = \text{argmin}_{V,Z}\{J(Z, V) | (Z, V) \in \mathcal{Z}(x_0)\}$

3. Set the input to $u = K(x - z_0^\star(x)) + v_0^\star(x)$

5 Implementation

Two options:

- Iterative optimization methods

- Explicit solution

EXPLICIT:

The CFTOC problem is a **multiparametric quadratic program (mp-QP)**

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

Definition 24 (Active Set). $A(x)$ and it's complement $NA(x)$

$$\begin{aligned} A(x) :=& \{j \in I : G_j z^\star(x) - S_j x = w_j\} \\ NA(x) :=& \{j \in I : G_j z^\star(x) - S_j x < w_j\} \end{aligned}$$

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 $\bar{x} \in \mathcal{K}^\star$ let $(A, NA) := (A(\bar{x}), NA(\bar{X}))$. Then

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

Online evaluation: Point location

Sequential search

Logarithmic search

OPTIMIZATION

L-Smooth

(UN-)CONSTRAINED OPTIMIZATION

Projected Gradient Method