Model Predictive Control

GianAndrea Müller

2. August 2019

Inhaltsverzeichnis

Maths

$$\begin{aligned} M > 0 \Leftrightarrow & M = M^T,\ \underline{z}^T M \underline{z} > 0\ \forall \underline{z} \\ \Leftrightarrow & \lambda_{\min}(M) \geq 0 \\ \Leftrightarrow & M^{-1} > 0 \\ A > B \Leftrightarrow & A - B > 0 \end{aligned}$$

$$\boxed{\sum_{k=0}^{\infty} a_0 q^k = \frac{a_0}{1-q}} \text{ Geometric Series}$$

$$\boxed{||\underline{x}||_p = \left(\sum_{i=0}^{n} |x_i|^p \right)^{1/p}} \text{ p-norm}$$

$$\boxed{||Ax||_2 \leq ||A||_2 ||x||_2 = \sqrt{\lambda_{\max}(A^T A)} ||x||_2}$$

$$\boxed{||Ax||_2^2 = x^T A^T A x}$$

$$\boxed{Q > 0 \Leftrightarrow \exists Q^{1/2} > 0 \ s.t.\ Q^{1/2} Q^{1/2} = Q}$$

0 Definitions

Definition 1. ***Infimum***: *Greatest lower bound to a set.*

Definition 2. *A system is* ***feasible*** *if it does not violate any constraint.*

Definition 3. *A* ***recursively feasible*** *system is defined to allow a series of control inputs that drives it to the target state and keeps it there without violating any constraints.*

1 Basics

1.1 Requirements for MPC

1. A model of the system
2. A state estimator
3. Definition of the optimal control problem
4. Setup of the omptimization problem
5. Optimal control sequence
6. Verification of performance

1.2 General MPC Problem

$$\begin{aligned} U_k^*(x(k)) := \quad & \text{argmin} \quad \sum_{i=0}^{N-1} l(x_{k+i}, u_{k+i}) \\ & \text{subj. to} \\ \text{measurement} \quad & x_k = x(k) \\ \text{system model} \quad & x_{k+i+1} = A x_{k+i} + B u_{k+i} \\ \text{state constraints} \quad & x_{k+i} \in \mathcal{X} \\ \text{input constraints} \quad & u_{k+i} \in \mathcal{U} \\ \text{optimization variables} \quad & U_k = \{u_k, u_{k+1}, \ldots, u_{k+N-1}\} \end{aligned}$$

1.3 Models of Dynamic Systems

Abbreviation	System Specification
TI	Time Invariant
LTI	Linear Time Invariant
CT	Continuous Time
DT	Discrete Time
SS	State Space

1.3.1 Nonlinear, TI, CT, SS

$$\boxed{\begin{aligned} \dot{x} &= g(x, u) \\ y &= h(x, u) \end{aligned}}$$

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

1.3.2 LTI, CT, SS

$$\boxed{\begin{aligned} \dot{x} &= A^c x + B^c u \\ y &= C x + D u \end{aligned}}$$

$x \in \mathbb{R}^n$	state vector	$A^c \in \mathbb{R}^{n \times n}$	system matrix
$u \in \mathbb{R}^m$	input vector	$B^c \in \mathbb{R}^{n \times m}$	input matrix
$y \in \mathbb{R}^p$	output vector	$C \in \mathbb{R}^{p \times n}$	output matrix
		$D \in \mathbb{R}^{p \times m}$	throughput matrix

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

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

Linearization (first make first order sytem!)

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

$$\dot{x} = \underbrace{g(x_s, u_s)}_{=0} + \underbrace{\left. \frac{\partial g}{\partial x^T} \right|_{\substack{x=x_s \\ u=u_s}}}_{=A^c} \underbrace{(x - x_s)}_{=\Delta x} + \underbrace{\left. \frac{\partial g}{\partial u^T} \right|_{\substack{x=x_s \\ u=u_s}}}_{=B^c} \underbrace{(u - u_s)}_{=\Delta u}$$

$$y = \underbrace{h(x_s, u_s)}_{y_s} + \underbrace{\left. \frac{\partial h}{\partial x^T} \right|_{\substack{x=x_s \\ u=u_s}}}_{=C} (x - x_s) + \underbrace{\left. \frac{\partial h}{\partial u^T} \right|_{\substack{x=x_s \\ u=u_s}}}_{=D} (u - u - s)$$

1.3.3 TI, DT, SS Systems

$$\boxed{\begin{aligned} x(k+1) &= g(x(k), u(k)) \\ y(k) &= h(x(k), u(k)) \end{aligned}}$$

$x \in \mathbb{R}^n$	state v.	$g(x,u) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$	system dynamics
$u \in \mathbb{R}^m$	input v.	$h(x,u) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{B}^p$	output function
$y \in \mathbb{R}^p$	output v.		

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

1.3.4 Euler Discretization of Nonlinear, TI Systems

$$\begin{aligned} \dot{x}^c(t) &= g^c(x^c(t), u^c(t)) \\ y^c(t) &= h^c(x^c(t), u^c(t)) \end{aligned}$$

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

$$\boxed{\begin{aligned} x(k+1) &= x(k) + T_s g^c(x(k), u(k)) = g(x(k), u(k)) \\ y(k) &= h^c(x(k), u(k)) = h(x(k), u(k)) \end{aligned}}$$

1.3.5 Euler Discretization of LTI Systems

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(t) \end{aligned}$$

$$\boxed{A = I + T_s A^c,\ B = T_s B^c,\ C = C^c,\ D = D^c}$$

1.3.6 Exact Discretization of LTI Systems

$$\begin{aligned} x(t_{k+1}) &= e^{A^c T_s} x(t_k) + \int_{t_k}^{t_{k+1}} e^{A^c(t_{k+1}-\tau)} B^c d\tau u(t_k) \\ &= \underbrace{e^{A^c T_s}}_{\triangleq A} x(t_k) + \underbrace{\int_0^{T_s} e^{A^c(T_s - \tau')} B^c d\tau'}_{\triangleq B} u(t_k) \end{aligned}$$

if A invertible: $B = (A^c)^{-1}(A - I)B^c$
Note that the first order approximation to the exact discretization is equivalent to the euler discretization.

$$e^{A^c T_s} = I + A^c T_s$$

1.4 Analysis of LTI DT Systems

1.4.1 Coordinate Transformations

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned}$$

Consider: $\tilde{x}(k) = Tx(k)$ s.t. T is invertible ($det(T) \neq 0$).

$$\begin{aligned} T^{-1}\tilde{x}(k+1) &= AT^{-1}\tilde{x}(k) + Bu(k) \\ y(k) &= CT^{-1}\tilde{x}(k) + Du(k) \end{aligned}$$

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

1.4.2 Stability of Linear Systems

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

is globally asymptotically stable

$$\lim_{k \to \infty} x(k) = 0, \forall x(0) \in \mathbb{R}^n \Leftrightarrow |\lambda_j| < 1,\ \forall j = 1, \ldots, n$$

For continuous systems: $\text{Re}(\lambda_i) < 0$

1.4.3 Controllability

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

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

$$x^* = x(N) = A^N x(0) + \begin{bmatrix} B & AB & \cdots & A^{N-1}B \end{bmatrix} \begin{bmatrix} u(N-1) \\ u(N-2) \\ \vdots \\ u(0) \end{bmatrix}$$

Cayley-Hamilton Theorem: A^k can be expressed as linear combinations of A^j, $j \in 0, 1, \ldots, n-1$ for $k \geq n$. Hence for all $N \geq n$:

$$\text{range}\begin{bmatrix} B & AB & \cdots & A^{N-1}B \end{bmatrix} = \text{range}\begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}$$

Thus if the system cannot be controlled to x^* in n steps, then it cannot in an arbitrary number of steps.

$$\boxed{\mathcal{C} = \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}} \text{ Controllability Matrix}$$

The system is controllable if $\mathcal{C} \begin{bmatrix} u(n-1) \\ u(n-2) \\ \vdots \\ u(0) \end{bmatrix} = x^* - A^n x(0)$ has a solution for all right-hand sides.

$$\boxed{\text{The system } (A, B) \text{ is controllable if } \mathcal{C} \text{ has full rank.}}$$

A system is called **stabilizable** if there exits a input sequence that returns the state to the origin asymptotically, starting from any point. True if all uncontrollable modes are stable.

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

where Λ_A^+ is the set of all eigenvalues of A lying on or outside the unit circle.

1.4.4 Observability

$$\begin{aligned} x(k+1) &= Ax(k) \\ y(k) &= Cx(k) \end{aligned}$$

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

Are the linear equations $\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(N-1) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{N-1} \end{bmatrix} x(0)$ unique?

$$\boxed{\mathcal{O} = \begin{bmatrix} C^T & (CA)^T & \cdots & (CA^{n-1})^T \end{bmatrix}^T}$$

Observability matrix

$$\boxed{\text{The system } (C, A) \text{ is observable if } \mathcal{O} \text{ has full rank.}}$$

First replace N by n (**Cayley-Hamilton**). Then the solution is unique if the columns of $\mathcal{O}$ are linearly independent.

A system is called **detectable** if it is possible to construct from the measurement sequence a sequence of state estimates that converges to the true state asymptotically, starting from an arbitrary initial estimate. True if all of its unobservable modes are stable.

$$\boxed{\text{rank}([A^T - \lambda_j I | C^T]) = n \ \forall \lambda_j \in \Gamma_A^+ \Rightarrow (A, C)}$$

where Γ_A^+ is the set of all eigenvalues of A lying on or outside the unit circle.

1.5 Stability of nonlinear DT Systems

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

with an equilibrium point at 0 i.e. $g(0) = 0$.

$$\boxed{\forall \ \epsilon > 0 \ \exists \ \delta(\epsilon) \ : \ ||x(0)|| < \delta(\epsilon) \rightarrow ||x(k)|| < \epsilon, \forall k \geq 0}$$

Lyapunov stability

An equilibrium point is **asymptotically stable** in $\Omega \subseteq \mathbb{R}^n$ if it is Lyapunov stable and **attractive**.

$$\boxed{\lim_{k \to \infty} x(k) = 0, \ \forall x(0) \in \Omega} \text{ Attractivity}$$

and **globally asymptotically stable** if it is asymptotically stable and $\Omega = \mathbb{R}^n$.

1.5.1 Lyapunov function

Definition 4. *Equilibrium point $x = 0$, $\Omega \subset \mathbb{R}^n$ a closed an bounded set containing the origin. A function $V : \mathbb{R}^n \rightarrow \mathbb{R}$, continuous at the origin, finite for every $x \in \Omega$, and such that:*

$$\boxed{\begin{aligned} &V(0) = 0 \textit{ and } V(x) > 0, \ \forall x \in \Omega \setminus \{0\} \\ &V(g(x)) - V(x) \leq -\alpha(x) \forall x \in \Omega \setminus \{0\} \end{aligned}}$$

*where $\alpha : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuous positive definite, is called a **Lyapunov function.***

Theorem 1. *If a system admits a Lyapunov function $V(x)$, then $x = 0$ is **asymptotically stable** in Ω.*

Theorem 2. *If a system admits a Lyapunov function $V(x)$ for $\Omega = \mathbb{R}^n$, which additionally satisfies $||x|| \rightarrow \infty \Rightarrow V(x) \rightarrow \infty$ then $x = 0$ is **globally asymptotically stable***

Theorem 3. *If the linearization of a nonlinear system around an isolated equilibrium point x_e is stable (unstable),then this equilibrium is an asymptotically stable (unstable) equilibrium of the nonlinear system as well. (**Lyapunov indirect method.**)*

Sum of L. funs is an L. fun as well.

1.5.2 Lyapunov Stability of LTI DT Systems

Candidate Lyapunov function: $V(x) = x^T P x$ with P positive definite.

$$\begin{aligned} V(Ax(k)) - V(x(k)) &= x^T(k) A^T P A x(k) - x^T P x(k) \\ &= x^T(k)(A^T P A - P) x(k) \leq -\alpha(x(k)) \end{aligned}$$

Can choose $\alpha(x(k)) = x^T Q x(k), Q > 0$, need to find $P > 0$ solving

$$\boxed{A^T P A - P = -Q, \ \ Q > 0} \text{ DT Lyapunov equation}$$

Theorem 4. *The discrete-time Lyapunov equation has a unique solution $P > 0$ if and only if A has all eigenvalues inside the unit circle, i.e. if and only if the system $x(k+1) = Ax(k)$ is stable.*

- Note that stability is always „global“ for linear systems.
- The infinite horizon cost-to-go for an asymptotically stable autonomous system $x(k+1) = Ax(k)$ with a quadratic cost function

$$\boxed{\begin{aligned} \phi(x(0)) &= \sum_{k=0}^{\infty} x(k)^T Q x(k) = \sum_{k=0}^{\infty} x(0)^T (A^k)^T Q A^k x(0) \\ &= x(0)^T P x(0) \end{aligned}}$$

To prove existence of Lyapunov function

$$x(k)^T \left(\sum_{i=1}^{\infty} (A^i)^T Q A^i \right) x(k) = x(k+1)^T \left(\sum_{i=0}^{\infty} (A^i)^T Q A^i \right) x(k+1)$$

Approach: 1. Linearize 2. Design LQR 3. If LQR closed-loop is stable (with observability and controllability) the system is stable.

1.5.3 Region of Attraction

1. Determine Lyapunov function.
2. Find for what regions it is valid.

2 Optimal Control

$$J^*(x(0)) := \min_U J(x(0), U)$$

$$\begin{aligned} \text{subj. to} \quad x_{i+1} &= g(x_i, u_i), \ i = 0, \ldots, N-1 \\ h(x_i, u_i) &\leq 0, \ i = 0, \ldots, N-1 \\ x_N &\in \mathcal{X}_f \\ x_0 &= x(0) \end{aligned}$$

2.1 Linear Quadratic Optimal Control

$$x(k+1) = Ax(k) + Bu(k) \quad \text{linear DT TI systems}$$

$$J(x_0, U) := x_N^T P x_N + \sum_{i=0}^{N-1} (x_i^T Q x_i + u_i^T R u_i)$$

quadratic cost functions

$Q = C^T C$ and $R = \rho I$

$$\sum_{i=0}^{N} ||y_i||_2^2 + \rho ||u_i||_2^2 \quad \text{energy in input and output signals}$$

Large $\rho \Rightarrow$ small input energy, output weakly controlled

Small $\rho \Rightarrow$ large input energy, output strongly controlled

2.2 Unconstrained Finite Horizon Control

$$\begin{aligned} J^*(x(0)) &:= \min_U x_N^T P x_N + \sum_{i=0}^{N-1} (x_i^T Q x_i + u_i^T R u_i) \\ \text{subj. to } x_{i+1} &= A x_i + B u_i,\ i = 0, \dots, N-1 \\ x_0 &= x(0) \end{aligned}$$

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

2.2.1 Batch approach

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

$$X := \mathcal{S}^x x(0) + \mathcal{S}^u U$$

Then write the cost function using

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

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

Replacing $X = \mathcal{S}^x x(0) + \mathcal{S}^u U$:

$$J(x(0), U) = U^T H U + 2x(0)^T F U + x(0)^T (\mathcal{S}^x)^T \bar{Q} \mathcal{S}^x x(0)$$

where $H := (\mathcal{S}^u)^T \bar{Q} \mathcal{S}^u + \bar{R}$ and $F := (\mathcal{S}^x)^T \bar{Q} \mathcal{S}^u$.

- Note that $H \succ 0$ since $R \succ 0$ and $(\mathcal{S}^u)^T \bar{Q} \mathcal{S}^u \succeq 0$, thus H^{-1} is guaranteed to exist.

Setting the gradient equal to zero:

$$\begin{aligned} \nabla_U J(x(0), U) &= 2HU + 2F^T x(0) = 0 \\ U^*(x(0)) &= -((\mathcal{S}^u)^T \bar{Q} \mathcal{S}^u + \bar{R})^{-1} (\mathcal{S}^u)^T \bar{Q} \mathcal{S}^x x(0) \end{aligned}$$

Optimal cost:

$$\begin{aligned} &J^*(x(0)) = \\ &x(0)^T \Big((\mathcal{S}^x)^T \bar{Q} \mathcal{S}^x - (\mathcal{S}^x)^T \bar{Q} \mathcal{S}^u ((\mathcal{S}^u)^T \bar{Q} \mathcal{S}^u + \bar{R})^{-1} \bullet \\ &\bullet (\mathcal{S}^u)^T \bar{Q} \mathcal{S}^x \Big) x(0) \end{aligned}$$

2.2.2 Verification of the Batch approach with quadprog

Write the cost as a function of U.

$$\begin{aligned} J_0(x(0), U) &= (\mathcal{S}^x x(0) + \mathcal{S}^u U)^T \bar{Q} (\mathcal{S}^x x(0) + \mathcal{S}^u U) + U^T \bar{R} U \\ &U^T H U + 2x(0)^T F U + x(0)^T \mathcal{S}^{xT} \bar{Q} \mathcal{S}^x x(0) \end{aligned}$$

where $H := \mathcal{S}^{uT} \bar{Q} \mathcal{S}^u$ and $F := \mathcal{S}^{xT} \bar{Q} \mathcal{S}^u$

2.2.3 Recursive Approach

Defining the j-step optimal cost-to-go: $J_j^*(x(j))$

The minimum of the cost attainable for the remainder of the horizon after step j.

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

$$\begin{aligned} \text{subj. to } \quad & x_{i+1} = A x_i + B u_i,\ i = j, \dots, N-1 \\ & x_j = x(j) \end{aligned}$$

see page 18-21 for an example

Procedure:

1. Start at step N $J_N^*(x_N) := l_f(x_N)$
2. Iterate **backwards** for $i = N-1, \dots, 0$ (DP iteration)

$$J_I^*(x_i) := \min_{u_i} l(x_i, u_i) + J_{i+1}^*(A x_i + B u_i)$$

3. $J^*(x_0) := J_0^*(x_0)$, optimal controller is the optimizer $u_0^*(x_0)$

Requirements:

- Closed-form representation of the function $J_i^*(x)$.
- Ability to compute a DP iteration.

Often not possible, except for some special cases (e.g. LQR)

2.2.4 Bellman's Principle of Optimality

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.

$$\boxed{\begin{aligned} J_j^*(x_j) &= \min_{u_j} l(x_i, u_i) + J_{j+1}^*(x_{j+1}) \\ &\text{subj. to } x_{j+1} = A x_j + B u_j \end{aligned}}$$

2.2.5 LQR

One step problem:

$$\begin{aligned} J_{N-1}^*(x_{N-1}) = \quad & \min_{u_{N-1}} x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1} + x_N^T P_N x_N \\ & \text{s.t. } x_N = A x_{N-1} + B u_{N-1} \\ & P_N = P \end{aligned}$$

where $x_j^T P_j x_j$ refers to the optimal cost-to-go. $P_N = P$

Substitution and setting the derivative equal to 0 yields the optimality condition:

$$u_{N-1}^* = -(B^T P_N B + R)^{-1} B^T P_N A x_{N-1} := F_{N-1} x_{N-1}$$

[..1-step cost to go → same for 2-step cost to go → recognise recursion..]

$$\boxed{u_i^* = -(B^T P_{i+1} B + R)^{-1} B^T P_{i+1} A x(i) := F_i x_i \text{ for } i = 1:N}$$

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

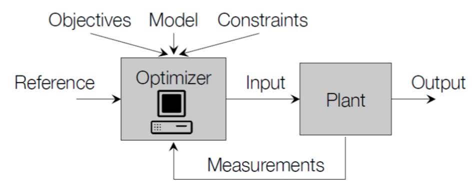
Discrete Time Riccati equation (RDE)

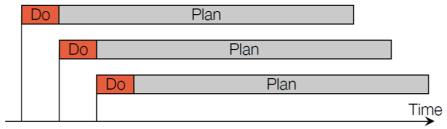
Evaluation down to P_0 we obtain the N-step cost-to-go.

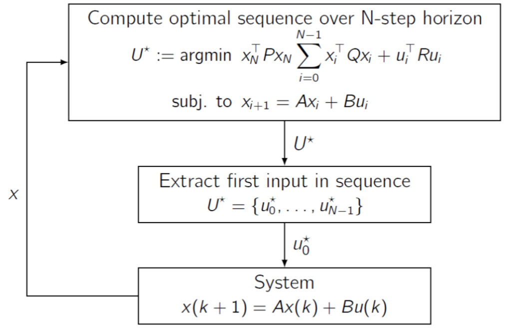
2.3 Comparison Batch/Recursive Approach

- Batch optimization returns a sequence of numeric values depending only on the inital state whereas the recursive approach yields feedback policies $u_i^* = F_i x_i$ depending on each x_i.
- They are identical if there are no disturbances.
- The recursive approach is more robust to disturbances and model errors, because if future states deviate the optimal input can still be computed.
- The recursive approach is computationally more attractive because it divides the problem into small calculations.
- Neither one method can deal with inequality constraints.

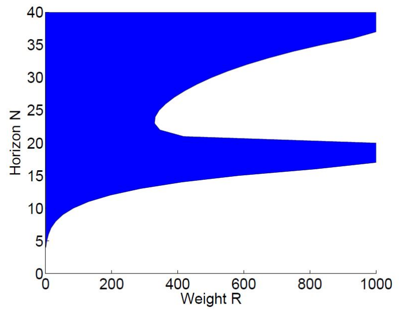
2.3.1 Receding Horizon







The stability of finite horizon controllers highly depends on the chosen horizon length, as well as on the costs R, Q and can be unintuitive.



Blue = stable, white = unstable

2.3.2 Infinite Horizon

$$J_\infty(x(0)) = \lim_{u(\cdot)} \sum_{i=0}^{\infty} (x_i^T Q x_i + u_i R u_i)$$
$$\text{subj. to } x_{i+1} = Ax_i + Bu_i, \quad i = 0, 1, 2 \ldots, \infty$$
$$x_0 = x(0)$$

$$\boxed{\begin{aligned} u^*(k) &= -(B^T P_\infty B + R)^{-1} B^T P_\infty A x(k) \\ &:= F_\infty(x(k) - x_s) + u_s \end{aligned}}$$

$$\boxed{J_\infty(x(k)) = x^T(k) P_\infty x(k)}$$

The matrix P_∞ comes from an infinity recursion of the RDE.

Assuming that the RDE does converge to some constant matrix P_∞ it must satisfy the following (with $P_i = P_{i+1} = P_\infty$):

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

- The constant feedback matrix F_∞ is referred to as the asymptotic form of the **Linear Quadratic Regulator (LQR)**.
- The closed-loop system with $u(k) = F_\infty x(k)$ is **guaranteed** to be stable if (A, B) is stabilizable and $(Q^{\frac{1}{2}}, A)$ is detectable.
- The infinite-horizon cost to go is actually a Lyapunov function for the system. Thus $\lim_{k\to\infty} x(k) = 0$
- Choices for the terminal cost:
  1. Equal to P_∞. To find it solve the are with $P_i = P_{i+1}$.
  2. Assuming no control action after the end of the horizon → solve the Lypunov equation for P:
  $$APA^T + Q = P$$
  This approach only makes sense if the system is asymptotically stable.
  3. If we want the state and the input both to be zero after the end of the finite horizon, no P but an additional constraint is needed:
  $$x_{i+N} = 0$$

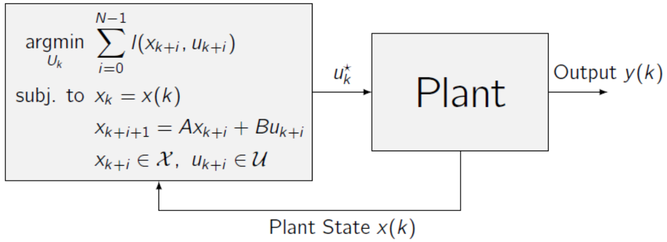
2.3.3 Bellman Equation

If we can find a function J such that

$$\boxed{J^*(x) := \min_u l(x, u) + J^*(Ax + Bu)} \text{ Bellman equation}$$

then $J^*(\cdot) = J^*_\infty(\cdot)$.

3 Convex Optimization



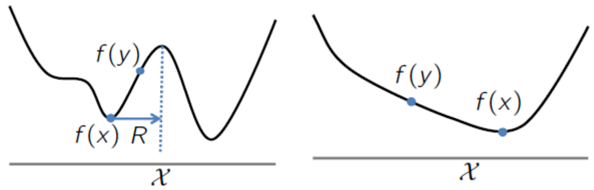
At each sample time:
1. Measure / estimate current state.
2. Find the optimal input sequence for N steps ahead.
3. Implement only the first control action u_k^*.

Mathematical Optimization:

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

with $\mathcal{X} := \{x \in \text{dom}(f) | g_i(x) \le 0, \ i = 0, \ldots, m, \ h_i(x) = 0, \ i = 0, \ldots, p\}$ the set of feasible decisions.

- **feasible point**: $x \in \text{dom}(f)$ satisfying the inequality and equality constraints.
- **strictly feasible point**: Feasible $x \in \text{dom}(f)$ strictly satisfying the inequality constraints.
- **Optimal value**: Lowest possible cost value $p^* = f(x^*) \triangleq \min_{x\in\mathcal{X}} f(x)$ also denoted by f^* or J^*.
- **Optimizer**: Any feasible x^* that achieves smalles cost p^*. The optimizer is not always unique.
- **Local/global optimality**:



- If $p^* = -\infty$ the problem is **unbounded below**.
- If $\mathcal{X}$ is empty the problem is **infeasible**.
- If $\mathcal{X} = \mathbb{R}^n$ the problem is **unconstrained**.
- The constraint $g_i(x) \le 0$ is **active** if $g_i(\bar{x}) = 0$. **Inactive** otherwise.
- A **redundant** constraint does not change the feasible set.

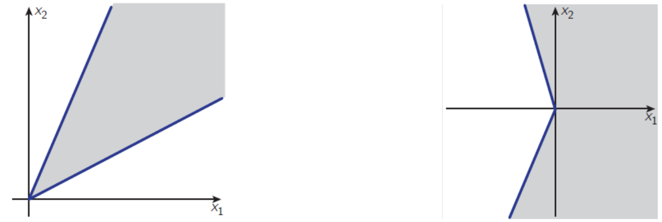
3.1 Convex sets

Definition 5. *A **convex set** can be mathematically defined as:*

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

This can be graphically illustrated by taking any two points in the set and connecting them with a line. If the line stays within the set for all combinations of points the set is convex.

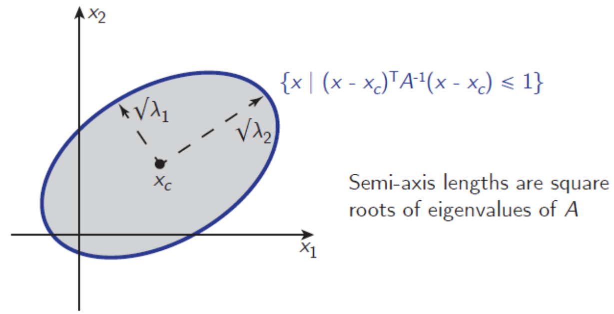
- **Affine Set**
  $\mathcal{X} = \{x \in \mathbb{R}^n \mid Ax = b\}$
  A subspace is an affine set with $b = 0$
- A **hyperplane** is defined by $\{x \in \mathbb{R}^n \mid a^T x = b\}$ for $a \ne 0$ where $a \in \mathbb{R}^n$ is the normal vector to the hyperplane.
- A **halfspace** is everything on one side of a hyperplane. It can be either **open** (strict inequality) or **closed** (non-strict inequality).
- An (unbounded) **polyhedron** is the intersection of a finite number of closed halfspaces.
  A **polytope** is a bounded polyhedron.
- **Cones**



A convex cone — A non-convex cone

A set $\mathcal{X}$ is a cone if for all $x \in \mathcal{X}$, and for all $\theta > 0$, $\theta x \in \mathcal{X}$. If the cone contains $x = 0$, it is **pointed**.

- **Ellipsoids** $\mathcal{X} := \{x | (x - x_c)^T A^{-1} (x - x_c) \le 1\}$



The **Euclidean Ball** $B(x_C, r)$ is a special case of the ellipsoid for which $A = r^2 \mathbb{I}$, such that $B(x_C, r) := \{x | \ ||x - x_c||_2 \le r\}$.
- The **intersection** of two or more convex sets is convex.
- The **union** of two complex sets is not necessarily convex.

3.1.1 Norms

A **norm** is any function $f : \mathbb{R}^n \to \mathbb{R}$ satisfying:
1. $f(x) \ge 0$ and $f(x) = 0 \Rightarrow x = 0$
2. $f(tx) = |t| f(x)$ for scalar t
3. $f(x + y) \le f(x) + f(y)$, for all $x, y \in \mathbb{R}^n$

$$\boxed{||x||_p := \left[\sum_{i=1}^{n} |x_i|^p\right]^{1/p}} \ l_p \text{ norm}$$

$$\boxed{\{x| \ ||x - x_c|| \le r\}} \text{ Norm Ball}$$

where x_c is the centre of the ball and r its radius. A ball is always convex for any norm.

3.1.2 Convex Hull

The **convex hull** is the smallest convex set that contains $\mathcal{X}$.

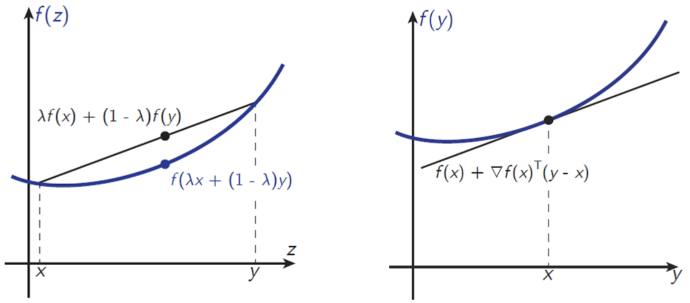
$$\boxed{\text{conv}(\mathcal{X}) = \left\{ \lambda_1 x_1 + \dots \lambda_q x_q \,|\, \lambda_i \geq 0, i = 1, \dots, 1, \ \sum_{i=1}^{q} \lambda_i = 1 \right\}}$$

3.2 Convex function

A function $f \ : \text{dom}(f) \to \mathbb{R}$ is **convex** iff $\text{dom}(f)$ is convex and

$$\boxed{\begin{array}{r} f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y), \\ \forall \lambda \in (0,1), \ \forall x, y \in \text{dom}(f) \end{array}}$$

The function f:dom$(f) \to \mathbb{R}$ is strictly convex if the inequality is strict.
The function f is **concave** iff dom(f) is convex and -f(x) is convex.



First order condition for convexity
A function $f \ : \text{dom}(f) \to \mathbb{R}$ with a convex domain is **convex** iff

$$\boxed{f(y) \geq f(x) + \nabla f(x)^T (y - x), \ \forall \ x, y \in \text{dom}(f)}$$

First order approximation of f around any point x is a global underestimator of f

Second order condition for convexity
A twice-differentiable function f:dom$(f) \to \mathbb{R}$ with convex domain is convex iff:

$$\boxed{\nabla^2 f(x) \succeq 0, \ \forall \ x \in \text{dom}(f)}$$

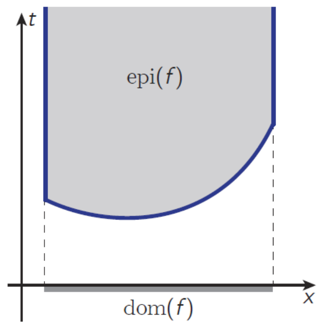
where $\nabla^2 f(x)_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$

If $\text{dom}(f)$ is convex and $\nabla^2 f(x) \succ 0 \ \forall x \in \text{dom}(f)$, then f is strictly convex.

Epigraph of a function The **epigraph** of a function f:dom$(f) \to \mathbb{R}$ is the set

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

A function is convex iff its epigraph is a complex set.



- Constant/Linear functions are both convex and concave.

3.2.1 Level and sublevel sets

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

$$\boxed{L_\alpha := \{x | x \in \text{dom}(f), f(x) = \alpha\}}$$

For $f(x) : \mathbb{R}^2 \to \mathbb{R}$ these are **contour lines** of constant height.

The **sublevel set** C_α of a function f for value α is:

$$\boxed{C_\alpha := \{x | x \in \text{dom}(f), f(x) \leq \alpha\}}$$

Function f is covex $\Rightarrow$ sublevel sets of f are complex for all α but not the other way around.

See the script for examples of convex functions [p. 38-40].

3.3 Convexity preserving operations

- **Non-negative weighted sum**
  If f is a function convex, then αf is convex for $\alpha \geq 0$. For several complex functions g_i, $\sum_i \alpha_i g_i$ is convex if all $\alpha_i \geq 0$.
- **Composite with affine function**
  If f is a convex function, then $f(Ax + b)$ is convex.
- **Pointwise maximum**
  If $f_1, \dots, f_m$ are convex functions, then $f(x) = \max\{f_1(x), \dots, f_m(x)\}$ is convex.
- **Pointwise supremum**
  If $f(x, y)$ is convex in x for every $y \in \mathcal{Y}$, then $g(x) = \sup_{y \in \mathcal{Y}} f(x, y)$ is convex.
- **Parametric minimization**
  If $f(x, y)$ is convex in (x, y) and the set $\mathcal{C}$ is convex, then
  $$g(x) = \min_{y \in \mathcal{C}} f(x, y)$$
  is convex
- **Composition with scalar functions**
  For $g : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R} \to \mathbb{R}$, $f(x) = h(g(x))$ is convex if:
  – g is convex, h is convex, $\tilde{h}$ is non-decreasing.
  – g is concave, h is convex, $\tilde{h}$ is non-decreasing.
- **Composition with vector functions**
  For $g \ : \ \mathbb{R}^n \to \mathbb{R}^k$ and $h \ : \ \mathbb{R}^k \to \mathbb{R}$, $f(x) = h(g(x)) = h(g_1(x), \dots, g_k(x))$ is convex if:
  – Each g_i is convex, h is convex, $\tilde{h}$ is non-decreasing in each argument.
  – Each g_i is concave, h is convex, $\tilde{j}$ is non-decreasing in each argument.

3.4 Convex Optimization Problem

$$\begin{array}{rll} \min\limits_{x \in \text{dom}(f)} & f(x) & \\ \text{subj. to} & g_i(x) \leq 0 & i = 1, \dots, m \\ & h_i(x) = 0 & i = 1, \dots, p \end{array}$$

where it is necessary that, f convex, $\text{dom}(f)$ convex, g_i convex, $h_i(x) = a_i^T x - b$ affine!

Thus the problem can be rewritten as:

$$\begin{array}{rl} \min\limits_{x \in \text{dom}(f(x))} & f(x) \\ \text{subj. to} & g_i(x) \leq 0 \ i = 1, \dots, m \\ & Ax = b \quad A \in \mathbb{R}^{p \times m} \end{array}$$

Feasible set of a convex optimization problem is convex.

3.4.1 Local and Global Optimality

For a convex optimization problem, any locally optimal solution is globally optimal!

3.4.2 Equivalent Optimization Problems

Two problems are called equivalent if the solution to one can be inferred from the solution to the other.
Introducing equality constraints:

$$\begin{array}{rl} & \min\limits_{x} f(A_0 x + b_0) \\ \text{subj. to} & g_i(A_i x + b_i) \leq 0 \ i = 1, \dots, m \end{array}$$

is equivalent to

$$\begin{array}{rl} & \min\limits_{x, y_i} f(y_0) \\ \text{subj. to} & g_i(y_i) \leq 0 \ i = 1, \dots, m \\ & A_i x + b_i = y_i \ i = 0, 1, \dots, m \end{array}$$

Although the second version has a nicer cost function it features more constraints.

Introducing slack variables s_i for linear inequalities:

$$\begin{array}{rl} & \min\limits_{x} f(x) \\ \text{subj. to} & A_i x \leq b_i \ i = 1, \dots, m \end{array}$$

is equivalent to

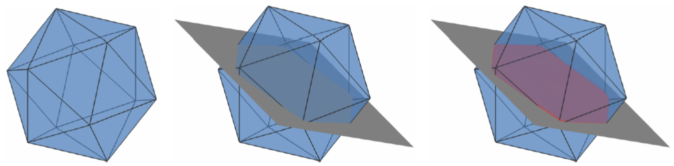
$$\begin{array}{rl} & \min\limits_{x, s_i} f(x) \\ \text{subj. to} & A_i x + s_i = b_i \ i = 1, \dots, m \\ & s_i \geq 0 \ i = 1, \dots, m \end{array}$$

3.5 General Linear Program (LP)

Affine cost and constraint functions.

$$\begin{aligned} \min_{x\in\mathbb{R}} c^T x \\ \text{subj. to } Gx \le h \\ Ax = b \end{aligned}$$

- Feasible set is a polyhedron.
- If P is empty the problem is infeasible.
- Each row of A defines a half space.



(a) $Gx \le h$ (b) $A_i^\top x = b_i$ (c) $Gx \le h \cap A_i^\top x = b_i$

Types of solutions:

1. The LP solution is unbounded, i.e. $p^* = -\infty$.
2. The LP solution is bounded, i.e. $p^* > -\infty$ and the optimizer is unique. X_{opt} is a singleton.
3. The LP solution is bounded and there are multiple optima. X_{opt} is a subset of $\mathbb{R}^s$, which can be bounded or unbounded.

3.6 General Quadratic Program

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \tfrac{1}{2}x^T Hx + q^T x + r \\ \text{subj. to } Gx \le h \\ Ax = b \end{aligned}$$

- Constant r can be left out since it does not effect the optimum.
- Convex if $H \succ 0$, otherwise there are non-unique solutions.
- Problems with concave objective $H \not\succeq 0$ are quadratic programs, but hard.

Types of solutions:

1. The optimizer lies strictly inside the feasible polyhedron.
2. The optimizer lies on the boundary of the feasible polyhedron.

4 Duality

4.1 The Lagrange Dual Problem

Primal Problem

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

with (primal) decision variable x, domain dom(f) and optimal value p^*.

Lagrangian function: L :dom$(f) \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$

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

- λ_i: inequality Lagrange multiplier for $g_i(x) \le 0$.
- ν_i: inequality Lagrange multiplier for $h_i(x) = 0$.
- Lagrangian is a weighted sum of the objective and constraint functions.

Lagrange Dual function:

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

- To find d solve $\nabla_x L(x,\lambda,\nu) = 0$ and insert back into L.
- The dual function is always a **convex** function.
- $d(\lambda,\nu)$ is the pointwise infimum of affine functions.
- dual function generates lower bounds for p^*.

$$d(\lambda,\nu) \le p^*, \;\forall(\lambda \ge 0, \nu \in \mathbb{R}^p)$$

- $d(\lambda,\nu)$ might be $-\infty$

$$\text{dom}(d) := \{\lambda,\nu | d(\lambda,\nu) > -\infty\}$$

If $d(\lambda,\nu)$ is close to $f(x)$ we know that we are close the the optimum.

$$\boxed{d(\lambda,\nu) \le d^* \le p^* \le f(x) \;\forall\; x \in \mathcal{X}}$$

Dual Problem

Every $\nu \in \mathbb{R}^p, \lambda \ge 0$ produces a lower bound for p^*. Which is the best?

$$\begin{aligned} \max_{\lambda,\nu} d(\lambda,\nu) \\ \text{subj. to } \lambda \ge 0 \end{aligned}$$

- Problem (D) is **convex** even if (P) is not.
- Problem (D) has optimal value $d^* \le p^*$.
- The point (λ,ν) is **dual feasible** if $\lambda \ge 0$ and (λ,ν) ∈dom(d).
- (λ,ν) ∈dom$(d$ can often be imposed explicitly in (D).

4.1.1 Example: Least norm solution to linear system

$$\begin{aligned} \min_{x\in\mathbb{R}} x^T x \\ \text{subj. to } Ax = b \end{aligned}$$

Lagrangian: $L(x,\nu) = x^T x + \nu^T(Ax - b)$

Dual function:

1. Minimize the Lagrangian by setting its gradient zero $\nabla_x L(x,\nu) = 2x + A^T\nu = 0 \Rightarrow x = -\frac{1}{2}A^T\nu$
2. Substitute back into Lagrangian to get Dual function:

$$\boxed{d(\nu) = -\tfrac{1}{4}\nu^T AA^T\nu - b^T\nu \le p^* \text{ for every } \nu}$$

4.1.2 Example: (Recitation) Duality of an LP

$$\begin{aligned} \min_x \underbrace{c^T x}_{f(x)} \\ \text{sb.t. } \underbrace{A'x - b'}_{g(x)} \le 0 \\ \underbrace{A''x - b''}_{h(x)} = 0 \end{aligned}$$

1. **Rewrite equality constraint as inequality constraint:**

$$h(x) = 0 \Leftrightarrow h(x) \le 0 \&\& - h(x) \le 0$$

$$\min_x c^T x$$
$$\text{sb.t. } Ax - b \le 0 \quad A = \begin{bmatrix} A' \\ A'' \\ -A'' \end{bmatrix}, \; b = \begin{bmatrix} b' \\ b'' \\ -b'' \end{bmatrix}$$

2. **Lagrangian primal and dual function**

$$L(x,\lambda) = c^T x + \sum_{i=1}^n \lambda_i(A_i x - b_i)$$

$$d(\lambda) = \max_x \inf_{x\in\text{dom}(f)} (c^+\lambda^T A)x - \lambda^T b$$

3. **KKT**

Primal feasibility	$Ax^* - b \le 0$
Dual feasibility	$\lambda_i^* \ge 0 \forall i = 1\dots m$
Complementary Slackness	$\lambda_i^*(A_i x^* - b_i)$
Stationarity Condition	$\nabla_x L(x^*,\lambda^*) = 0 = c + A^T\lambda^*$

4. **Dual Problem**

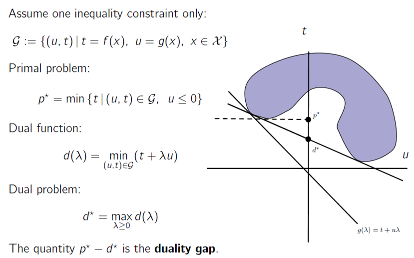
$$\begin{aligned} \max -\lambda^T b \\ \text{sb.t. } \lambda \ge 0 \\ \lambda^T A = -c^T \end{aligned}$$

Geometric Interpretation

$$\begin{aligned} t = f(x) \\ u = g(x) \le 0 \\ \mathcal{G} = \{(u,t) : t = f(x),\; u = g(x),\; x \in \mathcal{X}\} \end{aligned}$$

$$\textbf{primal: } p^* \le min\,\{t : (u,t) \in \mathcal{G}, u \le 0\}$$

$$\textbf{dual: } d(\lambda) = \min_{(u,t)\in\mathcal{G}} (t + \lambda u)$$
$$d^* = \max_{\lambda\ge 0} d(\lambda)$$



t-axis for $f(x)$, u-axis for $g(x)$ therefore only points with $u < 0$ are feasible.

4.1.3 Example: Dual of a Linear Program (LP)

$$\begin{aligned} \min_{x\in\mathbb{R}^n} c^T x \\ \text{subj. to } Ax = b \\ Cx \le e \end{aligned}$$

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

Thus the dual problem is:

$$\max_{\lambda,\nu} -b^T\nu - e^T\lambda$$
$$\text{subj. to } A^T\nu + C^T\lambda + c = 0$$
$$\lambda \geq 0 \text{ dual feasibility}$$

$\boxed{\text{The dual of an LP is also an LP}}$

4.1.4 Example: Norm minimization with equality constraint

$$\min_x ||x||_2$$
$$\text{subj. to } Ax = b$$

The dual function is:

$$\begin{aligned} d(\lambda) &= \min_x \left[||x|| - (A^T\nu)^T x + b^T\nu\right] \\ &= \begin{cases} b^T\nu & \text{if } ||A^T\nu||_2 \leq 1 \\ -\infty & \text{otherwise} \end{cases} \end{aligned}$$

The dual problem is:

$$\max_\nu b^T\nu$$
$$\text{subj. to } ||A^T\nu||_2 \leq 1$$

Lower bound: $b^T\nu \leq p^*$ whenever $||A^T\nu||_2 \leq 1$

4.1.5 Example: Dual of a Quadratic Program

$$\min_{x\in\mathbb{R}^n} \tfrac{1}{2}x^TQx + c^Tx$$
$$\text{subj. to } Cx \leq e$$

with $Q \succ 0$

The dual function is:

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

The optimal x satisfies: $Qx + c + C^T\lambda = 0$

Substituting $x = -Q^{-1}(c + C^T\lambda)$ into $d(\lambda)$

$$d(\lambda) = -\tfrac{1}{2}(c + C^T\lambda)^TQ^{-1}(c + C^T\lambda) - e^T\lambda$$

The **dual** problem is then:

$$\min_\lambda \tfrac{1}{2}\lambda^TC^TQ^{-1}C\lambda + (CQ^{-1}c + e)^T\lambda + \tfrac{1}{2}c^TQ^{-1}c$$
$$\text{subj. to } \lambda \geq 0$$

$\boxed{\text{The dual of a QP is a QP as well!}}$

4.1.6 Example: Dual of a Mixed-Integer Linear Problem (MILP)

$$\min_{x\in\mathcal{X}} c^Tx$$
$$\text{subj. to } Ax \leq b$$
$$\mathcal{X} = -1, 1^n$$

The dual function is:

$$\begin{aligned} d(\lambda) &= \min_{x_i\in\{-1,1\}} \left[c^Tx + \lambda^T(Ax - b)\right] \\ &= -||A^T\lambda + c||_1 - b^T\lambda \end{aligned}$$

The dual problem is:

$$\max_\lambda -||A^T\lambda + c||_1 - b^T\lambda$$
$$\text{subj. to } \lambda \geq 0$$

The dual of a mixed-integer LP is an LP (without integers).

4.2 Weak Duality

- It is **always** true that $d^* \leq p^*$.
- Sometimes the dual is much easier to solve than the primal (or vice-versa).

If $p^* \neq d^*$ then $p^* - d^*$ is the **duality gap**.

4.3 Strong Duality

- It is sometimes true that $d^* = p^*$.
- Strong duality does not hold for non-convex problems.

4.3.1 Slater Condition (valid for conv.opt. prob.)

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

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

$\boxed{\text{Then } p^* = d^*}$

4.4 Primal and Dual Solution Properties

Assume that strong duality holds, with optimal solution x^* and (λ^*, ν^*).
1. From strong duality: $d^* = p^* \Rightarrow d(\lambda^*, \nu^*) = f(x^*)$
2. From the definition of the dual function:

$$f(x^*) = d(\lambda^*, \nu^*) = f(x^*) + \underbrace{\sum_{i=1}^m \lambda_i^* g_i(x^*)}_{=0} + \underbrace{\sum_{i=1}^p \nu_i^* h_i(x^*)}_{=0}$$

3. **Complementary Slackness**

$$\lambda_i^* = 0 \text{ for every } g_i(x^*) < 0 \rightarrow \textbf{inactive constraint}$$
$$g_i(x^*) = 0 \text{ for every } \lambda_i^* > 0 \rightarrow \textbf{active constraint}$$

4.5 Karush-Kuhn-Tucker Conditions

Assume that all g_i and h_i are differentiable. **Necessary** conditions for optimality:
1. Primal Feasibility:

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

2. Dual Feasibility:

$$\lambda^* \geq 0$$

   - Introduce constraints in D s.t. $(\lambda, \nu) \in \mathrm{dom}(d)$!

3. Complementary Slackness:

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

4. Stationarity:

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

For a convex optimization problem:
1. If (x^*, λ^*, ν^*) satisfy the KKT conditions, then $p^* = d^*$.
   - $p^* = f(x^*) = L(x^*, \lambda^*, \nu^*)$ (due to complementary slackness.
   - $d^* = g(\lambda^*, \nu^*) = L(x^*, \lambda^*, v^*)$ (due to convexity of the functions and stationarity)
2. If the Slater conditions holds ($\rightarrow$ strong duality), then
   - x^* is optimal **if and only if** there exist (λ^*, ν^*) satisfying the KKT conditions.

4.5.1 Example: KKT Conditions for a QP

Provide a possibility to check whether a point is an optimum.

$$\min_{x\in\mathbb{R}^2} \tfrac{1}{2}x^TQx + c^Tx$$
$$\text{subj. to } Ax = b$$
$$x \geq 0$$

The **Lagr.** is $L(x, \lambda, \nu) = \frac{1}{2}x^TQx + c^Tx + \nu^T(Ax - b) - \lambda^Tx$
The KKT conditions are:

$$\begin{aligned} \nabla_x L(x, \lambda, \nu) = Qx + A^T\nu - \lambda + c &= 0 && \text{[stationarity]} \\ Ax &= b && \text{[primal feasibility]} \\ x &\geq 0 && \text{[primal feasibility]} \\ \lambda &\geq 0 && \text{[dual feasibility]} \\ x_i\lambda_i = 0 \quad i &= 1..n && \text{[complementarity]} \end{aligned}$$

The final three conditions can be written as $0 \leq x \perp \lambda \geq 0$

4.6 Sensitivity Analysis

What effect does changing a constraint have on the optimal solution?

General optimization problem and its dual:

$$\min_x f(x) \qquad\qquad \max_{\nu,\lambda} d(\nu\lambda)$$
$$\text{subj. to } g_i(x) \leq 0 \ i = 1 \ldots m \qquad \text{sb. to } \lambda \geq 0$$
$$h_i(x) = 0 \ i = 1 \ldots p$$

A perturbed optimization and its dual:

$$\min_x f(x) \qquad\qquad \max_{\nu,\lambda} d(\nu, \lambda) - u^T\lambda - v^T\nu$$
$$\text{sb. to } g_i(x) \leq u_i \ i = 1 \ldots m \qquad \text{subj. to } \lambda \geq 0$$
$$h_i(x) = v_i \ i = 1 \ldots p$$

where the perturbations are u_i and v_i

Assume **strong duality** for the unperturbed problem with (ν^*, λ^*) dual optimal. Weak duality for the perturbed problem then implies

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

4.6.1 Global Sensitivity Analysis

λ_i^* large and $u_i < 0 \;\Rightarrow p^*(u,v)$ incr. greatly.
λ_i^* small and $u_i > 0 \;\Rightarrow p^*(u,v)$ does not decr. much.
$\left\{\begin{array}{l} v^* \text{ large, positive, } v_i < 0 \\ v^* \text{ large, negative, } v_i > 0 \end{array}\right\} \Rightarrow p^*(u,v)$ incr. greatly.
$\left\{\begin{array}{l} v^* \text{ small, positive, } v_i > 0 \\ v^* \text{ small, negative, } v_i < 0 \end{array}\right\} \Rightarrow p^*(u,v)$ does not decr. much.
Note that the results are not symmetrical and that we only found a lower bound on $p^*(u,v)$.

4.6.2 Local Sensitivity Analysis

Assume **strong duality** for the unperturbed problem with (ν^*, λ^*) dual optimal. Weak duality for the perturbed problem then implies

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

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

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

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

4.7 Summary on Convex Optimization

- Convex optimization problem:
  - Convex cost function ($dom(f)$ also convex)
  - Convex inequality constraints
  - Affine equality constraints
- Benefit: Local = Global optimality
- If the slater condition holds x^* is optimal iff $\exists(\lambda^*, \nu^*)$ satisfying KKT conditions.
- The dual problem:
  - Is convex even if the primal is not.
  - Provides a lower bound for the primal problem: $d^* \leq p^*$ and thus a **suboptimality condition**.
  - Provides a certificate of optimality for convex problems via KKT.
  - Lagrange multipliers provide information about active constraints at the optimal solution and about the sensitivity of the optimal cost. How much will the cost increase if a constraint is tightened?

Guideline for dual problems

- Always:
  - Weak duality $d^* \leq p^*$
  - (D) convex, since dual function d always convex.
  - If x^*, (λ^*, ν^*) and $p^* = f^* \Rightarrow$ KKT satisfied.
- Primal problem nonconvex $\rightarrow$ **no strong duality**
- Primal problem convex
  - Strong duality guaranteed if Slater's Condition holds, which implies
    - Complementary Slackness
    - $p^* = d^*$
    - $\sum_{i=0}^{m} \lambda_i^* g_i(x^*) = \sum_{i=0}^{p} \nu_i^* h(x^*) = 0$
  - Given Slater's C check KKT to find the optimal solution.

5 Constrained Finite Time Optimal Control

We would like to solve constrained infinite time optimal control, but since there are an infinite number of variables this is not possible. Therefore the problem is reduced to constrained finite time optimal control which results in receding horizon optimal control.

5.1 Receding Horizon Optimal Control

DT model:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \\ x(k) \in \mathcal{X}, u(k) \in \mathcal{U}, \forall k \geq 0 \end{aligned}$$

The CFTOC writes as:

$$\begin{aligned} J^*_{k\to k+N|k}(x(k)) = \quad & \min_{U_{k\to k+N|k}} \; l_f(x_{k+N|k}) + \sum_{i=0}^{N-1} l(x_{k+i|k}, u_{k+i|k}) \\ \text{sb.t.} \quad & x_{k+i+1|k} = Ax_{k+i|k} + Bu_{k+i|k}, \; i = 0 \dots N-1 \\ & x_{k+i|k} \in \mathcal{X}, u_{k+i|k} \in \mathcal{U}, \; i = 0 \dots N-1 \\ & x_{k+N|k} \in \mathcal{X}_f \\ & x_{k|k} = x(k) \end{aligned}$$

is solved at time k with $U_{k\to N|k} = \{u_{k|k}, \dots, u_{k+N-1|k}\}$

- $x_{i+k|k}$ is the state of the model at time $k+i$, predicted at time k obtained by starting from the current state $xk|k = x(k)$ an applying to the system model the input sequence $u_{k|k}, \dots, u_{k+i-1|k}$
- Similarly $u_{k+i|k}$ is the input u at time $k+i$ computed at time k.
- Let $U^*_{k\to k+N|k} = \{u^*_{k|k}, \dots, u^*_{k+N-1|k}\}$ be the optimal solution. The first element of $U^*_{k\to k+N|k}$ is applied to the system. Then the CFTOC problem is reformulated and solved at time $k+1$ with the new state $x(k+1)$.

$$\boxed{\kappa_k(x(k)) = u^*_{k|k}(x(k))} \text{ Receding Horizon Control Law}$$

$$x(k+1) = Ax(k) + B\kappa_k(x(k)) := g_{cl}(x(k)), k \geq 0$$
Closed loop system

RHC: Time-invariant systems
System, Constraints and Cost function time invariant!

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

where $U = \{u_0 \dots, u_{N-1}\}$

5.2 Constrained Linear Optimal Control

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

- $U := \{u_0, \dots, u_{N-1}\}$
- Squared Eulcidean Norm:
  $l_f(x_N) = x_N^T P x_N$ and $l(x_i, u_i) = x_i^T Q x_i + u_i^T R u_i$
- $p = 1$ or $p = \infty$:
  $l_f(x_N) = ||Px_N||_p$ and $l(x_i, u_i) = ||Qx_i||_p + ||Ru_i||_p$

CFTOC:

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

Where N is the time horizon and $\mathcal{X}, \; \mathcal{U}, \; \mathcal{X}_f$ are polyhedral regions.

5.2.1 Feasible Set

Set of inital states $x(0)$ for which the optimal control problem is feasible:

$$\begin{aligned} \mathcal{X}_0 = \quad & \{x_0 \in \mathbb{R}^n | \exists (u_0 \dots u_{N-1}) \text{ s.t. } x_i \in \mathcal{X}, \; u_i \in \mathcal{U}, \\ & i = 0 \dots N-1, \; x_N \in \mathcal{X}_f, \text{ where } x_{i+1} = Ax_i + Bu_i\} \end{aligned}$$

In general $\mathcal{X}_j$ is the set of states x_j at time j for which the control problem is feasible, i.e. for which we can find a trajectory to $\mathcal{X}_f$ within N steps. **Independent of the cost.**

5.3 Constrained Optimal Control: Quadratic Cost

$$J(x_0, U) = x_N^T P x_N + \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i$$

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

CFTOC:

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

5.3.1 Construction of the QP with substitution

- Dense matrices, N optimization variables.

1. Rewrite the cost as
$$\begin{aligned} J(x(k), U) &= U^T H U + 2x(k)^T F U + x(k)^T Y x(k) \\ &= \begin{bmatrix} U^T & x(k)^T \end{bmatrix} \begin{pmatrix} H & F^T \\ F & Y \end{pmatrix} \begin{bmatrix} U^T & x(k)^T \end{bmatrix}^T \end{aligned}$$
2. Rewrite the constraints compactly as
$\mathcal{X} = \{x | A_x x \leq b_x\} \quad \mathcal{U} = \{u | A_u u \leq b_u\} \quad \mathcal{X}_f = \{x | A_f x \leq b_f\}$
$$GU \leq w + Ex(k)$$

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

3. Rewrite the constrained optimal control problem as

$$\begin{aligned} J^*(x(k)) &= \min_U \begin{bmatrix} U^T & x(k)^T \end{bmatrix} \begin{pmatrix} H & F^T \\ F & Y \end{pmatrix} \begin{bmatrix} U^T & x(k)^T \end{bmatrix}^T \\ \text{sb.t.} \quad & GU \leq w + Ex(k) \end{aligned}$$

Then we can find a solution for every k which results in a piecewise affine solution.

Quadratic Cost State Feedback Solution Multiparametric quadratic program (mp-QP) with the following solution properties:

- First component of the solution has the form

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

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

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

- The polyhedral sets $CR^j = \{x \in \mathbb{R}^n | H^j x \leq K^j\}, j = 1, \dots, N^r$ are a partition of the feasible polyhedron $\mathcal{X}_0$.
- The value function $J^*(x(k))$ is convex and piecewise quadratic on polyhedra.
- The central polyhedron represents unconstrained control and thus the LQR solution.

5.3.2 Construction of the QP without substitution

- Sparse matrices, $2N$ variables.

Idea: Keep state equations as equality constraints (often more efficient)

Resulting QP problem:

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

where $z = \begin{bmatrix} x_1^T & \cdots & x_N^T & u_0^T & \cdots & u_{N-1}^T \end{bmatrix}^T$

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

$$G_{\text{eq}} = \left[\begin{array}{cccc:cccc} I & & & & -B & & & \\ -A & I & & & & -B & & \\ & -A & I & & & & -B & \\ & & \ddots & \ddots & & & & \ddots \\ & & & -A \ \ I & & & & -B \end{array}\right], E_{\text{eq}} = \begin{bmatrix} A \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Inequalities: $G_{in} z \leq w_{in} + E_{in} x(k)$

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

$$G_{\text{in}} = \left[\begin{array}{cccc:cccc} 0 & & & & 0 & & & \\ A_x & & & & & 0 & & \\ & \ddots & & & & & \ddots & \\ & & A_x & & & & & 0 \\ & & & A_f & & & & 0 \\ \hdashline 0 & & & & A_u & & & \\ & 0 & & & & A_u & & \\ & & \ddots & & & & \ddots & \\ & & & 0 & & & & A_u \\ & & & 0 & & & & A_u \end{array}\right] \qquad w_{\text{in}} = \left[\begin{array}{c} b_x \\ b_x \\ \vdots \\ b_x \\ b_f \\ \hdashline b_u \\ b_u \\ \vdots \\ b_u \\ b_u \end{array}\right]$$

$$E_{\text{in}} = \begin{bmatrix} -A_x^\top & 0 & \cdots & 0 \end{bmatrix}^\top$$

Cost function from MPC $x_N^T P x_N + \sum_{i=0}^{N-1} x_i^T Q x_i + u_i^T R u_i$

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

Matlab hint:

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

5.4 Constrained Optimal Control: 1-Norm and ∞-Norm

$$J(x_0, U) := ||PX_N||_p + \sum_{i=0}^{N-1} ||Qx_i||_p + ||Ru_i||_p$$

with $p = 1$ or $p = \infty$, P, Q, R full rank column matrices

CFTOC:

$$\begin{aligned} J^*(x(k)) &= \min_U J(x_0, U) \\ \text{sb.t.} \quad & x_{i+1} = Ax_i + Bu_i, i = 0 \dots N-1 \\ & x_i \in \mathcal{X},\ u_i \in \mathcal{U},\ i = 0 \dots N-1 \\ & x_N \in \mathcal{X}_f \\ & x_o = x(k) \end{aligned}$$

5.4.1 l_∞ (Chebyshev) minimization

$$\min_{x \in \mathbb{R}^n} ||x||_\infty \quad \text{sb.t. } Fx \leq g$$

$$\min_{x \in \mathbb{R}^n} [\max\{x_1, \dots, x_n, -x_1, \dots, -x_n\}] \quad \text{sb.t. } Fx \leq g$$

$$\begin{aligned} \min_{x,t}\ & t \\ \text{sb.t.} \quad & x_i \leq t \ \ i = 1 \dots n \\ & -x_i \leq t \ \ i = 1 \dots n \\ & Fx \leq g \end{aligned}$$

$$\begin{aligned} \min_{x,t} \\ \text{sb.t.} \quad & -\mathbf{1}t \leq x \leq \mathbf{1}t \\ & Fx \leq g \end{aligned}$$

5.4.2 l_1 minimization

Constrained l_1 minimization

$$\min_{x \in \mathbb{R}^n} ||x||_1 \quad \text{sb.t. } Fx \leq g$$

$$\min_{x \in \mathbb{R}^n} \left[\sum_{i=1}^n \max\{x_i, -x_i\}\right] \quad \text{sb.t. } Fx \leq g$$

$$\begin{aligned} \min_{x \in \mathbb{R}^n, t \in \mathbb{R}^n}\ & t_1 + \dots + t_n \\ \text{sb.t. } x_i & \leq t_i \ \ i = 1 \dots m \\ -x_i & \leq t_i \ \ i = 1 \dots m \\ Fx & \leq g \end{aligned}$$

$$\begin{aligned} \min_{x \in \mathbb{R}^n, t \in \mathbb{R}^n}\ & \mathbf{1}^T t \\ \text{sb.t. } & -t \leq x \leq t \\ & Fx \leq g \end{aligned}$$

5.4.3 Construction of the LP for l_∞

Following the procedure above the original problem can be written as:

$$\begin{aligned} \min_z \quad & \epsilon_0^x + \dots + \epsilon_N^x + \epsilon_0^u + \dots + \epsilon_{N-1}^u \\ \text{subj. to} \quad & -\underline{1}_n \epsilon_i^x \leq \pm Q \left[A^i x_0 + \sum_{j=0}^{i-1} A^j B u_{i-1-j}\right] \\ & -\underline{1}_r \leq \pm P \left[A^N x_0 + \sum_{j=0}^{N-1} A^j B u_{N-1-j}\right] \\ & \underline{1}_m \epsilon_i^u \leq \pm R u_i \\ & A^i x_0 + \sum_{j=0}^{i-1} A^j B u_{i-1-j} \in \mathcal{X},\ u_i \in \mathcal{U} \\ & A^N x_0 + \sum_{j=0}^{N-1} A^j B u_{N-1-j} \in \mathcal{X}_f \\ & x_0 = x(k),\ i = 0, \dots, N-1 \end{aligned}$$

which in standard LP form can be written as

$$\begin{aligned} \min_z \quad & c^T z \\ \text{subj. to} \quad & \bar{G} z \leq \bar{w} + \bar{S} x(k) \end{aligned}$$

where $z := \{\epsilon_0^x, \dots, \epsilon_N^x, \epsilon_0^u, \dots, \epsilon_{N-1}^u, u_0^T, \dots, u_{N-1}^T\} \in \mathbb{R}^s$ and $s = (m+1)N + N + 1$

$$\bar{G} = \begin{bmatrix} G_\epsilon & 0 \\ 0 & G \end{bmatrix}, \bar{S} = \begin{bmatrix} S_\epsilon \\ S \end{bmatrix}, \bar{w} = \begin{bmatrix} w_\epsilon \\ w \end{bmatrix}$$

1 – /∞–Norm State Feedback Solution Multiparametric linear program (mp-LP) and exhibits the same basic properties as the quadratic cost state feedback solution. The following distinctions have to be made:

- Quadratic cost solution is either:
  - unique and in the interior of feasible set → no constraints active
  - unique and on the boundary of feasible set → at least 1 active constraint
- Linear cost solution is either:
  - unbounded
  - unique at a vertex of feasible set → at least n active constraints
  - a set of multiple optima → at least 1 active constraint

6 Invariance

Definition 6. *Let A and B be subsets of $\mathbb{R}^n$. The **Minkowski Sum** is:*

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

$$[a, b] \oplus [c, d] = [a + c, b + d] \quad \textit{Scalar case}$$

Definition 7. *Let A and B be subsets of $\mathbb{R}^n$. The **Pontryagin Difference** is:*

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

$$[a, b] \ominus [c, d] = [a - c, b - d] \quad \textit{Scalar case}$$

6.1 Objectives of Constrained Control

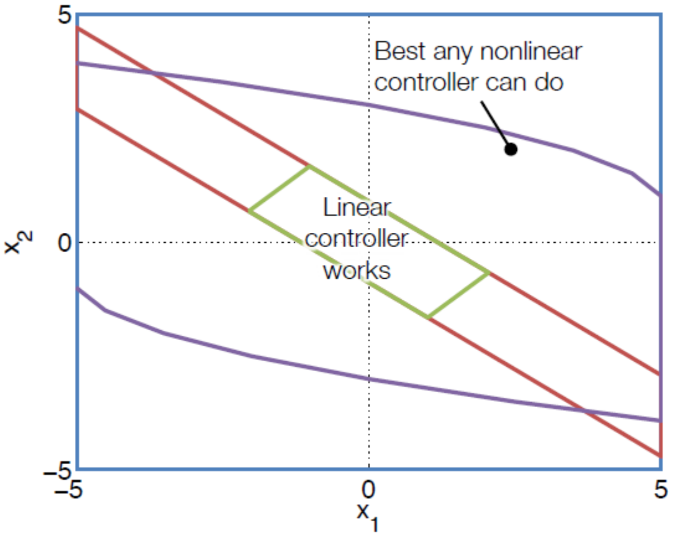
1. Constraint satisfaction
2. Stability
3. Optimal performance
4. Maximize the set $\{x(0)|$ Conditions 1-3 are met$\}$

6.2 Limitations of Linear Controlers

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

$\mathcal{X} := \{x| \; ||x||_\infty \leq 5\}$

$\mathcal{U} := \{u| \; ||u||_\infty \leq 1\}$



Controlled Invariance:
Will there always exist a valid input that will maintain constraints?

6.3 Invariance

$$\boxed{x(k) \in \mathcal{O} \Rightarrow x(k+1) \in \mathcal{O}, \; \forall \; k \in \{0, 1, \ldots\}}$$

Positive Invariant Set

If the invariant set is within the constraints, it provides a set of initial states from which the trajectory will never violate the system constraints.

$\boxed{\mathcal{O}_\infty \subset \mathcal{X}}$ Maximal Positive Invariant Set

if $0 \in \mathcal{O}_\infty$, $\mathcal{O}_\infty$ is invariant and contains all invariant sets that contain the origin.

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

$$\text{pre(s)} := \{x | g(x) \in S\}$$

Input: g, $\mathcal{X}$
Output: $\mathcal{O}_\infty$

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

Invariant Set Conditions: A set $\mathcal{O}$ is a positive invariant set if and only if:

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

6.4 Controlled Invariance

Control Invariant Set: A set $\mathcal{C} \subseteq \mathcal{X}$ is said to be control invariant if:

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

Maximum Control Invariant Set $\mathcal{C}_\infty$:
The set $\mathcal{C}_\infty$ is control invariant and contains all control invariant sets contained in $\mathcal{X}$.

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

- For box constraints and linear system check **all** corner points to find the invariant controller.

6.4.1 Pre-Set Computation: Controlled System

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

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

see pages 44-52, lecture 5 for an example.

6.4.2 Control Law Synthesis

A valid control law $\kappa(x(k))$ will ensure that a system $x(k+1) = g(x(k), u(k))$ always stays in the control invariant set:

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

This fact can be used to synthesize a controller:

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

where f is any function (including $f(x, u) = 0$)). This does not ensure convergence but will satisfy the constraints.

6.5 Practical Computation of Invariant Sets

$\boxed{E := \{x | (x - x_c)^T P (x - x_c) \leq 1\}}$ Ellipsoid

Lemma: If $V : \mathbb{R}^n \rightarrow \mathbb{R}$ is a Lyapunov function for the system $x(k+1) = g(x(k))$ then

$$Y := \{x | V(x) \leq \alpha\}$$

is an invariant set for all $\alpha \geq 0$, since it is a sublevel set of a Lyapunov function.

6.5.1 Example

$$A^T P A - P \succ 0$$

where $V(x) = x(k)^T P x(k)$ is a Lyapunov function.

Now we want to find the largest α s.t. the invariant set Y_α is contained within the system constraints $\mathcal{X}$:

$$Y_\alpha := \{x | x^T P x \leq \alpha\} \subset \mathcal{X} := \{x || Fx \leq f\}$$

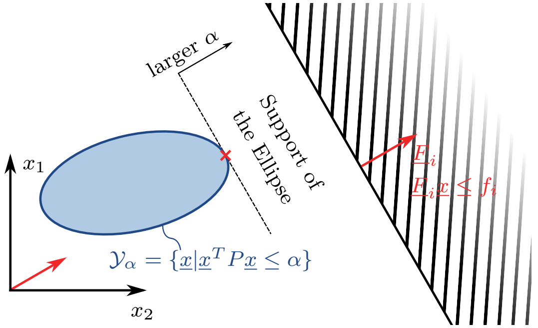
This is equivalent to the problem:

$$\begin{aligned} &\max_\alpha \alpha \\ &\text{sb.t. } h_{Y_\alpha}(F_i) \leq f_i \; \forall \; i \in \{1, \ldots, n\} \end{aligned}$$

Support of an ellipse:

$$\begin{aligned} h_{Y_\alpha} &= \max_x \gamma^T x \\ &\text{sb.t. } x^T P x \leq \alpha \end{aligned}$$

As long as $h_{Y_\alpha} < f_i$ the ellipse **does not violate the constraint**. The support identifies the point closest to the constraint relying on the scalar product, that effectively projects a certain point within the ellipse onto the direction of the constraint, thus returning a maximum for the point reaching towards the constraint the most.



Change of variables: $y := P^{\frac{1}{2}} x$

$$h_{Y_\alpha}(\gamma) = \max_y \gamma^T P^{-\frac{1}{2}} y$$
$$\text{sb.t. } y^T y \leq \sqrt{\alpha^2}$$

which can be solved by inspection:

$$h_{Y_\alpha} = \gamma^T P^{-\frac{1}{2}} \frac{P^{-\frac{1}{2}}\gamma}{||P^{-\frac{1}{2}}\gamma||} \sqrt{\alpha} = ||P^{-\frac{1}{2}}\gamma||\sqrt{\alpha}$$

The solution follows as:

$$\alpha^* = \max_\alpha \alpha \text{ s.t. } ||P^{-1/2} F_i T||^2 \alpha \leq f_i^2 \ \forall i \in \{1, \dots, n\}$$
$$= \min_{i \in \{1,\dots,n\}} \frac{f_i^2}{F_i P^{-1} F_i^T}$$

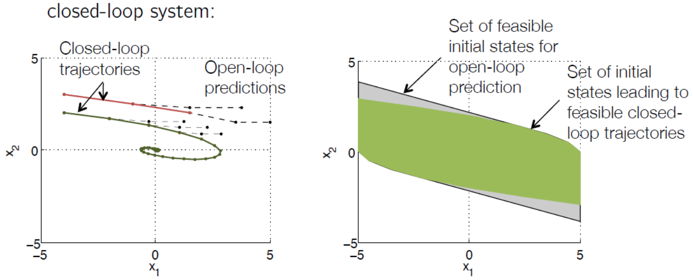
6.6 Summary Invariant Sets

- Core component of MPC problem.
- Special case: Linear System/Polyhedral Constraints
  - Polyhedral invariant set
    - Can represent the maximum invariant set
    - Can be complex (many inequalities) for more than ∼ 5 - 10 states
    - Resulting MPC optimization will be a quadratic program
  - Ellipsoidal invariant set
    - Smaller than polyhedral set (not maximum invariant set)
    - Easy to compute for large dimensions
    - Fixed complexity
    - Resulting MPC optimization will be quadratically constrained quadratic program

7 Terminal Cost, Constraint and Controller

Problems originate from the use of a 'short sighted' strategy

⇒ Finite horizon causes deviation between the open-loop prediction and the closed-loop system:



Ideally we would solve the MPC problem with an infinite horizon, but that is computationally intractable

⇒ Design finite horizon problem such that it approximates the infinite horizon

→ Introduce terminal cost and constraints to explicitly ensure feasibility and stability! $l_f()$ and $\mathcal{X}_f$ are chosen to mimic an infinite horizon.

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

- **Infinite-Horizon**
  Solution of the RHC problem with $N = \infty$ → open loop trajectories are the same as closed loop trajectories.
  - Problem feasible → closed loop trajectories will always be feasible.
  - Cost finite → states and inputs will converge to origin.
- **Finite-Horizon**
  RHC is „short-sighted“strategy approximating $N = \infty$-controller but:
  - **Feasibility**: After some steps the problem might become infeasible even without disturbance and modelling uncertainty.
  - **Stability**: The generated control inputs may not lead to convergent trajectories.

7.1 Proof of Feasibility and Stability

1. Prove recursive feasibility by showing the existence of a feasible control sequence at all time instants when starting from a feasible initial point.
2. Prove stability by showing that the optimal cost function is a Lyapunov function.

There are two possible cases:
1. Terminal constraint at zero: $x_N = 0$
2. Terminal constraint in some (convex) set: $x_N \in \mathcal{X}_f$

7.1.1 Proof of $x_N \in \mathcal{X}_f = 0$

First:
- Assume feasibility of $x(k)$ and let $\{u^*_{0|k}, \dots, u^*_{N-1|k}\}$ be the optimal control sequence and $\{x(k), x^*_{1|k}, \dots, x^*_{N|k}\}$ the corresponding trajectory.
- Apply $u^*_{0|k}$ and let the system evolve to $x(k+1) = Ax(k) + Bu^*_{0|k}$
- At $x(k+1) = x^*_{1|k}$ the shifted control sequence $\tilde{U} = \{u^*_{1|k}, \dots, u^*_{N-1|k}, 0\}$ is feasible (apply 0 control input $\Rightarrow x_{N+1} = 0$.

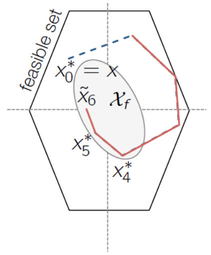
Second:
- Show $J^*(x(k+1)) < J^*(x(k)) \ \forall \ x(k) \neq 0$
- $J^*(x(k)) = \underbrace{l_f(x^*_{N|k})}_{=0} + \sum_{i=0}^{N-1} l(x^*_{i|k}, u^*_{i|k}$
- $J^*(x(k+1) \leq \tilde{J}(x(k+1)) = \sum_{i=1}^{N} l(x_i, \tilde{u}_i)$

$$\begin{aligned} \sum_{i=1}^{N} l(x_i, \tilde{u}_i) = \quad & \sum_{i=0}^{N-1} l(x^*_{i|k}, u^*_{i|k}) - l(x^*_{0|k}, u^*_{0|k}) + l(x_N + u_N) \\ & J^*(x(k)) - \underbrace{l(x(k), u^*_{0|k})}_{\text{stagecost k}} + \underbrace{l(0,0)}_{\text{final cost} = 0} \end{aligned}$$

Thus $J^*(x)$ is a Lyapunov function → Stability

Stability of MPC - Outline of the Proof

- Assume feasibility of $x(k)$ and let $\{u^\star_{0|k}, \ u^\star_{1|k}, \ \dots, \ u^\star_{N-1|k}\}$ be the optimal control sequence computed at $x(k)$ and $\{x(k), \ x^\star_{1|k}, \ \dots, \ x^\star_{N|k}\}$ the corresponding state trajectory



- At $x(k+1) = x^\star_{1|k}$, the control sequence $\tilde{U} = \{u^\star_{1|k}, \ u^\star_{2|k}, \ \dots, \ \kappa_f(x^\star_{N|k})\}$ is feasible:

  $x^\star_{N|k}$ is in $\mathcal{X}_f \to \kappa_f(x^\star_{N|k})$ is feasible

  and $x_{N+1} = Ax^\star_{N|k} + B\kappa_f(x^\star_{N|k})$ in $\mathcal{X}_f$

⇒ **Terminal constraint provides recursive feasibility**

Asymptotic Stability of MPC - Outline of the Proof

$$J^\star(x(k)) = \sum_{i=0}^{N-1} l(x^\star_{i|k}, u^\star_{i|k}) + l_f(x^\star_{N|k})$$

At $x(k+1) = x^\star_{1|k}$, $\tilde{U} = \{u^\star_{1|k}, \ u^\star_{2|k}, \ \dots, \ \kappa_f(x^\star_{N|k})\}$ is feasible & sub-optimal

$$\begin{aligned} J^\star(x(k+1)) &\leq \sum_{i=1}^{N} l(x_i, \tilde{u}_i) + l_f(Ax_N + B\kappa_f(x_N)) \\ &= \underbrace{\sum_{i=0}^{N-1} l(x^\star_{i|k}, u^\star_{i|k})}_{J^\star(x(k)) - l_f(x^\star_{N|k})} - l(x^\star_{0|k}, u^\star_{0|k}) + l(x_N, \kappa_f(x_N)) + l_f(Ax_N + B\kappa_f(x_N)) \\ &= J^\star(x(k)) - l(x(k), u^\star_{0|k}) + \underbrace{l_f(Ax_N + B\kappa_f(x_N)) - l_f(x_N) + l(x_N, \kappa_f(x_N))}_{\leq 0 \text{ by Assumption 3}} \\ &\Longrightarrow J^\star(x(k+1)) - J^\star(x(k)) \leq -l(x(k), u^\star_{0|k}), \quad l(x,u) > 0 \text{ for } x, u \neq 0 \end{aligned}$$

$J^\star(x)$ **is a Lyapunov function**

⇒ The closed-loop system under the MPC control law is asymptotically stable

7.2 Stability of MPC - Main Result

Assumptions:
1. Stage cost positive definite.
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 $\mathcal{X}_f$

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

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

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

Under those 3 assumptions:
Theorem: 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 the system.

7.3 Choice of terminal Sets and Cost - QP

- $\mathcal{X}_f = 0$ simplest choice but small region of attraction for small N
- Alternatively design LQR and get $F_\infty, \ P_\infty$.
- Terminal weight $P = P_\infty$, thus a possible invariant terminal set is

$$\mathcal{X}_f^\beta = \{x \in \mathbb{R}^n | \underline{x}^T P_\infty \underline{x} \leq \beta\}$$

  thus a sublevel set of a Lyapunov function (invariant by definition). Use the ellipse-support approach to find β.
- $\mathcal{X}_f \rightarrow$ maximum invariant set for closed-loop:

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

Then all the **Assumptions** of the Feasibility and Stability Theorem are verified.

7.4 Summary

- Finite-horizon MPC may be not stable!
- Finite-horizon MPC may not satisfy constraints for all time!
- An infinite-horizon provides stability and invariance.
- We fake infinite-horizon by forcing the final state to be in an invariant set for which there exists an invariance-inducing controller, whose infite-horizon cost can be expressed in closed form.
- These ideas extend to non-linear systems, but the sets are difficult to compute.

8 Practical Issues

8.1 Reference Tracking

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned}$$

where $x \in \mathbb{R}^{n_x}, \ u \in \mathbb{R}^{n_u}, \ y \in \mathbb{R}^{n_y}$
$\mathcal{X} = \{x | H_x x \leq k_x\}, \ \mathcal{U} = \{x | H_u u \leq k_u\}$

Goal: Track given reference r such that $y(k) \rightarrow r$ as $k \rightarrow \infty$
How to change the general MPC problem to achieve tracking?

$$\begin{aligned} U^*(x(k)) := \text{argmin}_{U_k} \quad & l_f(x_N) + \sum_{i=0}^{N-1} l(x_{k+i}, u_{k+i}) \\ \text{sb.t.} \quad & x_k = x(k) \\ & x_{k+i+1} = Ax_{k+i} + Bu_{k+i} \\ & x_{k+i} \in \mathcal{X} \\ & u_{k+i} \in \mathcal{U} \\ & U_k = \{u_k, u_{k+1}, \ldots, u_{k+N-1}\} \end{aligned}$$

$\Rightarrow$ Target condition, which is a steady state:

$$\begin{gathered} \min u_s^T R u_s \\ \text{sb.t.} \begin{bmatrix} I-A & -B \\ C & 0 \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix} \\ x_s \in \mathcal{X}, \ u_s \in \mathcal{U} \end{gathered}$$

where x_s and u_s represent the desired steady-state condition.

If no solution exists compute reachable set point that is closest to r:

$$\begin{gathered} \min (Cx_s - r)^T Q_s (Cx_s - r) \\ \text{sb.t. } x_s = Ax_s + Bu_s \\ x_s \in \mathcal{X}, \ u_s \in \mathcal{U} \end{gathered}$$

The new MPC is then designed as follows:

$$\begin{gathered} \min_U ||y_N - Cx_s||_{P_y}^2 + \sum_{i=0}^{N-1} ||y_i - Cx_s||_{Q_y}^2 + ||u_i - u_s||_R^2 \\ \text{sb.t. same constraints} \end{gathered}$$

Then the difference between x and x_s is defined as Δx and analogous for all other, such that we end up with:

$$\begin{aligned} \min \quad & \sum_{i=0}^{N-1} \Delta x_i^T Q \Delta x_i + \Delta u_i^T R \Delta u_i + V_f(\Delta x_n) \\ \text{sb.t.} \Delta x_0 &= \Delta x(k) \\ \Delta x_{i+1} &= A\Delta x_i + B\Delta u_i \\ H_x \Delta x_i &\leq k_x - H_x x_s \\ Hu \Delta u_i &\leq k_u - H_u u_s \\ \Delta x_N &\in \mathcal{X}_f \end{aligned}$$

Convergence

Assume feasibility in $x_s \in \mathcal{X}, \ u_s \in \mathcal{U}$ and choose terminal weight $V_f(x)$ and constraint $\mathcal{X}_f$ satisfying:
- $\mathcal{X}_f \subset \mathcal{X}, \ Kx \in \mathcal{U} \ \forall \ x \in \mathcal{X}_f$
- $V_f(x^+) - V_f(x) \leq -l(x, Kx) \ \forall \ x \in \mathcal{X}_f$

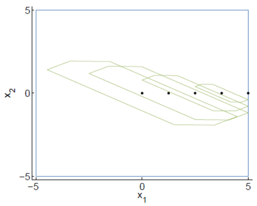
If in addition the target reference is such that

$$x_s \bigoplus \mathcal{X}_f \subset \mathcal{X}, \ K\Delta x + u_s \in \mathcal{U} \ \forall \ \Delta x \in \mathcal{X}_f$$

then the closed-loop system converges to the target reference.

8.2 Scaling the terminal Set

For tracking, if chosing $x_s \neq 0$ the terminal set has to be shifted with x_s. A large terminal set may only allow for a small set of feasible target since if it is moved to much its extreme states become infeasible. For that reason the moving terminal set is scaled down when getting close to state constraints.



8.2.1 Augmented Model

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

The augmented system is observable **iff** (A, C) is observable and

$$\begin{bmatrix} A-I & B_d \\ C & C_d \end{bmatrix} \text{ has full column rank}$$

$\Rightarrow$ Maximal dimension of the disturbance: $n_d \leq n_y$

State observer for augmented model

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

where $\hat{x}, \hat{d}$ are estimates of the state.

Error dynamics:

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

$\Rightarrow$ Choose $L = \begin{bmatrix} L_x \\ L_d \end{bmatrix}$ s.t. the error dynamics are stable and converge to zero.

8.3 Offset-free Tracking

Suppose the observer is stable and the number of outputs n_y equals the dimension of the constant disturbance n_d. The observer state satisfies:

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

where $y_{m,\infty}$ and u_∞ are the steady state measured outputs and inputs.
$\Rightarrow$ Observer output $C\hat{x}_\infty + C_d \hat{d}_\infty$ tracks the measurement $y_{m,\infty}$ without offset.

This leads to a new condition at steady-state:

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

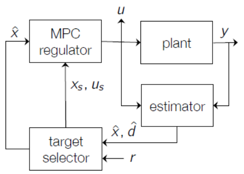
Thus we adapt the target condition according to the disturbance:

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

In practice:
1. Estimate state and disturbance, $\hat{x}, \hat{d}$
2. Obtain (x_s, u_s) from steady state target problem using disturbance estimate.
3. Solve MPC problem for tracking using disturbance estimate $\hat{d}$:

subj. to $x_0 = \hat{x}(k)$

$$
\begin{aligned}
& d_0 = \hat{d}(k) \\
& x_{i+1} = Ax_i + Bu_i + B_d d_i, && i = 0,\dots,N \\
& d_{i+1} = d_i, && i = 0,\dots,N \\
& x_i \in \mathcal{X},\ u_i \in \mathcal{U}, && i = 0,\dots,N-1 \\
& x_N - x_s \in \mathcal{X}_f
\end{aligned}
$$



Main result:

- $\kappa(\hat{x}(k), \hat{d}(k), r(k)) = u_0^*$

- $n_d = n_y$
- RHC recursively feasible and unconstrained for $k \geq j$ with $j \in \mathbb{N}^+$.
- Closed-loop system:

$$
\begin{aligned}
x(k+1) &= f(x(k), \kappa(\hat{x}, \hat{d}, r) \\
\hat{x}(k+1) &= (A + L_x C)\hat{x} + (B_d + L_x C_d)\hat{d} \\
&\quad + B\kappa(\hat{x}, \hat{d}, r) - L_x y_m(k) \\
\hat{d}(k+1) &= L_d C\hat{x}(k) + (i + L_d C_d)\hat{d}(k) - L_d y_m(k)
\end{aligned}
$$

converges, i.e. $\hat{x} \to \hat{x}_\infty, \hat{d} \to \hat{d}_\infty, y_m \to y_{m,\infty}$

Then $y_m(k) \to r$ as $k \to \infty$

8.4 Enlarging the Feasible Set

The introduction of a terminal set reduces the feasible set. → MPC **without terminal constraint**, with guaranteed stability.

Possible if:
- initial state lies in sufficiently small subset of feasible set.
- N is sufficiently large.

such that the terminal state satisfies the terminal constraint without enforcing it in the optimization. Thus the solution of the finite horizon MPC problem corresponds to the infinite horizon solution.

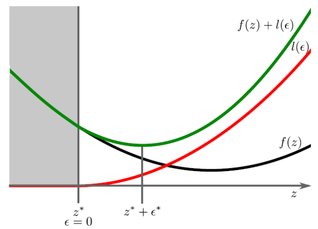
Advantage: Controller defined in larger feasible set. **Disadvantage**: Characterization of region of attraction or specification of required horizon length extremly difficult.

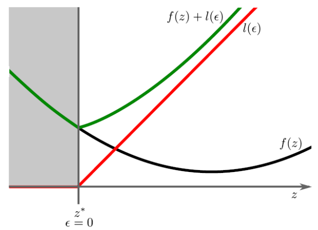
With larger horizon length N, region of attraction approaches maximum control invariant set.

8.5 Soft Constraints

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

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





Main result:

- $l_\epsilon(\epsilon) = v \cdot \epsilon$ satisfies the requirement for any $v \geq \lambda^* \geq 0$, where λ^* is the optimal Lagrange multiplier for the original problem.
- **Disadvantage**: $l_\epsilon(\epsilon) = v \cdot \epsilon$ renders the cost non-smooth.
- Solution: Combine quadratic and linear cost:

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

where $v \geq \lambda^*,\ s > 0$

$$\boxed{v_i > \lambda_i^*} \text{ Exactness}$$

- Extension to multiple constrains:

$$l_\epsilon(\epsilon) = \sum_{j=1}^{r} v_j \cdot \epsilon_j + s_j \cdot \epsilon_j^2$$

8.5.1 Simplification: Separation of Objectives

1. Minimize violation over horizon:

$$
\begin{aligned}
\epsilon^{\min} \quad &= \operatorname{argmin}_{U,\epsilon} \epsilon_i^T S \epsilon_i + v^T \epsilon_i \\
&\text{s.t. } x_{i+1} = Ax_i + Bu_i \\
&H_x x_i \leq K_x + \epsilon_i \\
&H_u u_i \leq K_u \\
&\epsilon_i \geq 0
\end{aligned}
$$

Now fix the slack variables!

2. Optimize controller performance:

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

- **Advantage:** Simplifies tuning, constraints will be satisfied if possible.
- **Disadvantage:** Requires the solution of two optimization problems.

8.6 Putting it all together

- In general state cannot be measured.
  - Use Kalman filter to estimate the state.
- Design tracking problem:
  - Rewrite problem in delta-formulation.
  - Setup target steady-state problem.
  - Calculate terminal weight and scale terminal constraint to guarantee convergence.
- Extend to offset-free tracking:
  - Augment model including disturbance model.
  - Augment the estimator to estimate the state and the disturbance.
  - Adapt target steady-state problem using the disturbance estimate.
- Possibly: Remove terminal constraint while choosing long horizon.
- Introduce soft constraints to ensure feasibility.
  - Introduce slack variables for constraint relaxation.
  - Choose penalty on slack variables (quadratic, linear).

9 Robust MPC

9.1 Uncertainty Models

- **Measurement / Input Bias:**

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

where θ is unknown, but constant

- **Linear Parameter Varying System:**

$$g(x(k), u(k), w(k); \theta) = \sum_{j=0}^{t} \theta_j A_j x(k) + \sum_{k=0}^{t} \theta_j B_j u(k)$$
$$\mathbf{1}^T \theta = 1,\ \ \theta \geq 0$$

where A_k, B_k known, θ_k unknown, but with fixed value at each sampling time

- **Additive Stochastic Noise:**

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

Distribution of w known

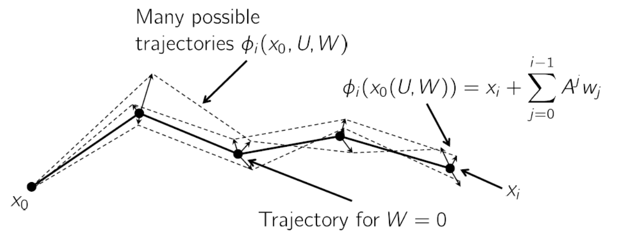
- Additive Bounded Noise

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

A,B known, w unknown and changing at each sampling instance
  - Dynamics are linear but impacted by random, bounded noise at each time step.
  - Can model many non-linearities in this fashion, but often a conservative model.
  - The noise is *persistent*, i.e. it does not converge to zero in the limit.

9.2 Impact of Bounded Additive Noise

Goal: 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 neighbourhood of the origin.
3. Optimizes (expected/worst case) „performance“.
4. Maximizes the set $\{x(0)|\text{Conditions 1-3 are met}\}$.



$$x(k+1) = Ax(k) + Bu(k)$$
Nominal System

$$\boxed{x(k+1) = Ax(k) + Bu(k) + w(k),\ w \in \mathcal{W}}$$
Uncertain System

9.2.1 Defining a Cost to Minimize

- Minimize the expected value (requires assumption on the distribution)

$$J_N(x_0, U) := E[J(x_0, U, W)]$$

- Take the worst-case

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

- Take the nominal case

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

In this lecture we will assume the nominal case for simplicity.

9.2.2 Constraint Satisfaction

In order to robustly enforce constraints of a linear system the concept of robust invariance is developed:

First the MPC prediction is broken into two parts:

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

- $i = 0 \dots N-1$
- Optimize over control actions.
- Enforce constraints explicitly by imposing $\phi_i \in \mathcal{X}$ and $u_i \in \mathcal{U}$ for all sequences W.

$$\begin{aligned} \phi_N & \in \mathcal{X}_f \\ \phi_{i+1} & = (A+BK)\phi_i + wi \end{aligned}$$

- $i = N, \dots$
- Assume control law to be linear $u_i = K\phi_i$.
- Enforce constraints implicitly by constraining ϕ_N to be in a **robust invariant set** $\mathcal{X}_f \subseteq \mathcal{X}$ and $K\mathcal{X}_f \subseteq \mathcal{U}$ for the system $\phi_{i+1} = (A+BK)\phi_i + w_i$.

9.2.3 Robust Invariance

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

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

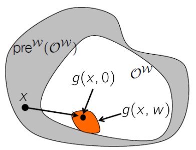
$$\text{pre}^{\mathcal{W}}(\Omega) := \{x | g(x, w) \in \Omega \text{ for all } w \in \mathcal{W}\}$$

See p 24-27 for an example on computing robust pre-sets for linear systems.

9.2.4 Robust Invariant Set Conditions

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

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



For computing the maximum robust invariant set use the algorithm from the nominal case, replacing $\text{pre}(\Omega)$ by $\text{pre}^{\mathcal{W}}(\Omega)$.

See p. 30-34 for an example on computing robust invariant sets.

9.2.5 Ensuring satisfaction of robust contstraints

Goal: Ensure that constraints are satisfied for the MPC sequence:

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

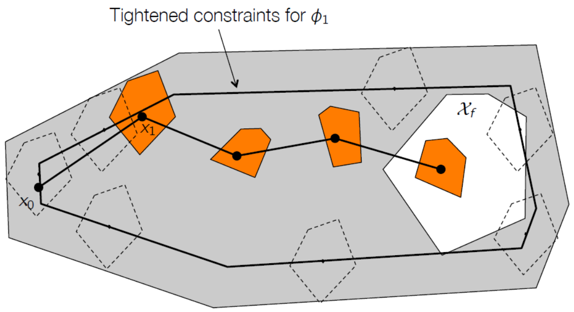
Assume that $\mathcal{X} = \{x | Fx \leq f\}$ then this is equivalent to:

$$Fx_i + F\sum_{j=0}^{i-1} A^j w_k \leq f \forall W \in \mathcal{W}^i$$

This leads to:

$$Fx_i \leq f - \max_{W \in \mathcal{W}^i} F \sum_{j=0}^{i-1} A^j w_j = f - h_{\mathcal{W}^i}\left(F \sum_{j=0}^{i-1} A^j\right)$$

What this results in is a tightening of the constraints on the nominal system!



For the terminal state constraint we can do exactly the same.

9.3 Open-Loop MPC

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

$$\begin{aligned} \text{subj. to } x_{i+1} &= Ax_i + Bu_i \\ x_i &\in \mathcal{X} \ominus (\mathcal{W} \oplus A\mathcal{W} \oplus \cdots \oplus A^{i-1}\mathcal{W}) \\ u_i &\in \mathcal{U} \\ x_N &\in \mathcal{X}_f \ominus (\mathcal{W} \oplus A\mathcal{W} \oplus \cdots \oplus A^{i-1}\mathcal{W}) \end{aligned}$$

where $\mathcal{X}_f$ is a robust invariant set for the system $x(k+1) = (A+BK)x(k)$ for some stabilizing K.

- We do **nominal** MPC but with tighter constraints on the states.
- If $U^*(x(k))$ is the optimizer of the robust open-loop MPC problem for $x(k) \in \mathcal{X}_0$ then the system $Ax(k) + Bu_0^*(x(k)) + w(k) \in \mathcal{X}_0$ for all $w \in \mathcal{W}$. This follows since the trajectory we computed at the current time is feasible for any disturbance.
- Potentially has a very small region of attraction, in particular for unstable systems.

9.4 Closed-Loop Predictions

Challenge: Cannot predict where the state of the system will evolve. We can only compute a set of trajectories that the system may follow.

Idea: Design a control law that will satisfy constraints and stabilize the system for all possible disturbances.
Possible structure of control-functions:

- **Pre-stabilization:** $\mu_i(x) = Kx + v_i$
  - Fixed K, s.t. A+BK is stable.
  - Simple, often conservative.
- **Linear feedback:** $\mu_i(x) = K_i x + v_i$
  - Optimize over K_i and v_i
  - Non-convex. Extremely difficult to solve.
- **Disturbance feedback:** $\mu_i(x) = \sum_{j=0}^{i-1} M w_j + v_i$
  - Optimize over M and v_j
  - Equivalent to linear feedback, but convex
  - Can be very effective, but computationally intense.
- **Tube-MPC** $\mu_i(x) = v_i + K(x - \bar{x}_i)$
  - fixed K, s.t. A+BK is stable
  - Optimize over $\bar{x}_i$ and v_i
  - Simple, and can be effective

9.5 Tube-MPC

Seperate the availabe control authority into two parts:

1. A portion that determines the optimal trajectory to the origin for the nominal system.

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

2. A portion that compenstates for deviations from this system, i.e. a 'tracking' controller, to keep the real trajectory close to the nominal.

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

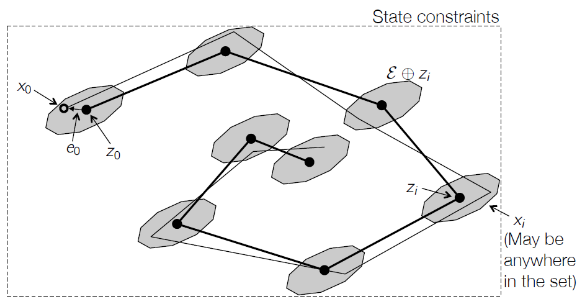
for some linear controller K, which stabilizes the nominal system.

9.5.1 Error dynamics

Define the error $e_i = x_i - z_i$ which gives the error dynamics:

$$\begin{aligned} e_{i+1} &= x_{i+1} - z_{i+1} \\ &= (A+BK)e_i + w_i \end{aligned}$$

There is some set that e will stay inside for all time. We want the smallest such set (the 'minimal invariant set').



To make it work:

- Compute the set $\mathcal{E}$ that the error will remain inside.
- Modify constraints on nominal trajectory $\{z_i\}$ so that $z_i \bigoplus \mathcal{E} \subset \mathcal{X}$ and $v_i \in \mathcal{U} \ominus K\mathcal{E}$.
- Formulate as convex optimization problem.

And prove that

- Constraints are robustly satisfied.
- The closed-loop system is robustly stable.

9.5.2 Compute $\mathcal{E}$

What is the set F_i that contains all possible states x_i?

$$F_i = \mathcal{W} \bigoplus A\mathcal{W} \ldots \bigoplus A^{i-1}\mathcal{W} = \bigoplus_{j=0}^{i-1} A^j \mathcal{W}, \ F_0 := \{0\}$$

Minimal Robust Invariant Set

Input: A
Output: F_∞
$\Omega_0 \leftarrow \{0\}$
loop
  $\Omega_{i+1} \leftarrow \Omega_i \oplus A^i \mathcal{W}$
  if $\Omega_{i+1} = \Omega_i$ **then**
    return $F_\infty = \Omega_i$
  end if
end loop

- A finite n does not always exist, but a 'large' n is a good approximation
- If n is not finite, there are other methods of computing small invariant sets, which will be slightly larger than F_∞

$$[a, b] \oplus [c, d] = [a + c, b + d] \qquad \sum_{i=0}^{\infty} x^i = \frac{1}{1-x} \text{ if } |x| < 1$$

9.5.3 Constraint Tightening

We want to work with the nominal system but ensure that the noisy system satisfies constraints!

$$\boxed{z_i \bigoplus \mathcal{E} \subseteq \mathcal{X} \Leftarrow z_i \in \mathcal{X} \ominus \mathcal{E}} \text{ Sufficient condition}$$

The set $\mathcal{E}$ is known offline - thus the tightened constraints can be computed offline.

For the input:

$$\boxed{u_i \in K\mathcal{E} \bigoplus v_i \subset \mathcal{U} \Leftarrow v_i \in \mathcal{U} \ominus K\mathcal{E}}$$

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

9.5.4 Tube-MPC Problem Formulation

$$\mathcal{Z}(x_0) := \left\{ Z, V \,\middle|\, \begin{array}{ll} z_{i+1} = Az_i + Bv_i & i \in [0, N-1] \\ z_i \in \mathcal{X} \ominus \mathcal{E} & i \in [0, N-1] \\ v_i \in \mathcal{U} \ominus K\mathcal{E} & i \in [0, N-1] \\ z_N \in \mathcal{X}_f & \\ x_0 \in z_0 \bigoplus \mathcal{E} & \end{array} \right\}$$

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

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

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

- Optimizing the nominal system, with tightened state and input constraints.
- First tube center is optimization variable $\rightarrow$ has to be within $\mathcal{E}$ of x_0.
- The cost is with respect to the tube centers (nominal system).
- The terminal set is with respect to the tightened constraints.

9.5.5 Tube-MPC Assumptions

1. The stage cost is a positive function, i.e. it is strictly positive and only zero at the origin.
2. The terminal set is invariant **for the nominal system** under the local control law $\kappa_f(z)$:

   $$Az + B\kappa_f(z) \in \mathcal{X}_f \text{ for all } z \in \mathcal{X}_f$$

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

   $$\mathcal{X}_f \in \mathcal{X} \ominus \mathcal{E}, \ \kappa_f(z) \in \mathcal{U} \ominus K\mathcal{E} \text{ for all } z \in \mathcal{X}_f$$

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

   $$l_f(Az + B\kappa_f(z)) \leq -l(z, \kappa_f(z)) \text{ for all } z \in \mathcal{X}_f$$

   And thus $\mathcal{X}_f$ is a level set of l_f.

9.5.6 Tube-MPC Robust Invariance

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

Let $(\{v_0^*, \ldots, v_{N-1}^*\}, \{z_0^*, \ldots, z_N^*\})$ be the optimal solution for $x(k)$.

Now since by construction $x(k+1) \in z_1 \oplus \mathcal{E}$ the optimal sequence is feasible for all $x(k+1)$.

9.5.7 Tube-MPC Robust Stability

The state $x(k)$ of the system $x(k+1) = Ax(k) + B\mu_{tube}(x(k)) + w(k)$ converges in the limit to the set $\mathcal{E}$.

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

$$\begin{aligned} J^*(x(k+1)) &\leq \sum_{i=1}^{N} l(z_i^*, v_i^*) + l_f(z_N + 1) \\ &= J^*(x(k)) - \underbrace{l(z_0^*, v_0^*)}_{\geq 0} \\ &\quad \underbrace{-l_f(z_N^*) + l_f(z_{N+1} + l(z_N^*, \kappa_f(z_N^*))}_{\leq 0 (l_f \text{ is a Lyapunov function in } \mathcal{X}_f} \end{aligned}$$

This shows that $\lim_{k\to\infty} J(z_0^*(x(k))) = 0$ and therefore $\lim_{k\to\infty} z_0^*(x(k)) = 0$.

However $x(k)$ does not tend to zero but stay within a region $\mathcal{E}$ around zero.

9.6 Summary: Tube MPC

– **Offline** –

1. Choose a stabilizing controller K such that $||A + BK|| \leq 1$.
2. Compute the minimal robust invariant set $\mathcal{E} = F_\infty$ for the system $x(k+1) = (A + BK)x(k) + w(k), \ w \in \mathcal{W}^1$.
3. Compute the tightened constraints $\tilde{\mathcal{X}} := \mathcal{X} \ominus \mathcal{E}, \ \tilde{\mathcal{U}} := \mathcal{U} \ominus K\mathcal{E}$
4. Choose terminal weight function l_f and constraint $\mathcal{X}_f$ satisfying the assumptions made.

– **Online** –

1. Measure / estimate state x.
2. Solve the problem $(V^*(x), Z^*(x)) = \text{argmin}_{V,Z}\{J(Z, V)|(Z, V) \in \mathcal{Z}(x)\}$
3. Set the input to $u = K(x - z_0^*(x)) + v_0^*(x)$

Benefits:

- Less conservative than open-loop robust MPC, since we are actively compensation for the disturbance.
- Works for unstable systems.
- Optimization problems to solve are simple.

Cons:

- Sub-optimal MPC (optimal is extremely difficult).
- Reduced feasible set when compared to nominal MPC.
- We need to know what $\mathcal{W}$ is (this is usually not realistic).

9.7 Summary on Robust MPC for Uncertain Systems

- **Idea:** Compensate for noise in prediction to ensure all constraints are met.

- Complex (some schemes are simple to implement, like tubes, but complex to understand)
- Must know the largest noise $\mathcal{W}$
- Often conservative
- Feasible set may be small

\+ Feasible set is invariant - we know exactly when the controller will work
\+ Easier to tune - knobs to tradeoff robustness against performance

10 Robustness of Nominal MPC

We want to control the noisy system

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

Now running standard MPC on that gives us the following closed-loop system:

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

for which we can prove convergence to a neighbourhood of the origin (for linear systems), but depending on the noise realization it may not be feasible.

10.1 Do we still have Lyapunov decrease?

Nominally

$$J^*(Ax + Bu^*(x)) - J^*(x) \leq -l(x, u^*(x))$$

But now our state develops as follows:

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

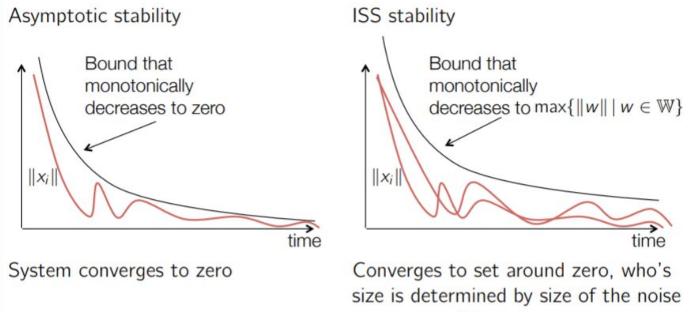
The optimal cost J^* is continuous for linear systems, convex constraints and continuous stage costs:

$$\begin{aligned} &|J^*(Ax + Bu^*(x) + w) - J^*(Ax + Bu^*(x))| \leq \\ &\gamma||Ax + Bu^*(x) + w - (Ax + Bu^*(x))|| = \gamma||w|| \end{aligned}$$

Thus the Lyapunov decrease can be bounded as:

$$\begin{aligned}
&J^*(Ax + Bu^*(x) + w) - J^*(x)\\
&\quad = J^*(Ax + Bu^*(x) + w) - J^*(x)\\
&\qquad - J^*(Ax + Bu^*(x)) + J^*(Ax + Bu^*(x))\\
&\quad \leq J^*(Ax + Bu^*(x)) - J^*(x) + \gamma ||w||\\
&\quad \leq -l(x, u^*(x)) + \gamma ||w||
\end{aligned}$$

- Amount of decrease grows with $||x||$
- Amount of increase is upper bounded by $\max\{||w|| \; |w \in \mathcal{W}\}$
- Thus we move towards the origin until there is a balance between the size of x and the size of w. Thus the system is **Input-to-State-Stable (ISS)**



10.2 Summary

\+ Simple
\+ No knowledge of the noise set $\mathcal{W}$ is required
\+ Often very effective in practice
\+ Feasible set is large (we can find a solution, but it might not work)
\+ Region of attraction may be larger than other approaches
\- Very difficult to determine region of attraction
\- Hard to tune - no obvious way to tradeoff robustness against performance
\- Works for linear systems, for nonlinear systems only under continuity assumptions

11 Implementation

11.1 Explicit MPC

- Linear MPC + Quadratic or linear-norm cost $\Rightarrow$ Controller is piecewise affine function
- We can pre-compute the controller offline
- Online evaluation of PWA is very fast
- This is only possible for very small systems (3-6 states)

When there is an explicit solution to the MPC problem posed, the optimization can be solved offline, resulting in a control law that is piecewise affine. Thus for finding the current control action, the system state has to be located within the partitioned feasible polyhedron. This search can be done sequentially or through a search tree:

11.1.1 Sequential Search vs. Search Tree

- **Sequential Search**
  - Very simple
  - Linear in number of regions
- **Search Tree**
  - Offline construction of a search tree by finding hyperplanes that separate regions into two equally sized parts and repeating that for the resulting subsets.
  - Potentially logarithmic
  - Significant offline processing (reasonable for < 1000 regions)

11.2 Iterative Optimization Methods

In all but the simplest cases no explicit solution can be obtained.
Iterative optimization methods:

$$x^{(i+1)} = \Psi(x^{(i)}, f, \mathbb{Q}), \; i = 0, 1, \ldots, m-1$$
$$\text{s.t.}$$
$$|f(x^{(m)}) - f(x^*)| \leq \epsilon \text{ and } \mathrm{dist}(x^{(m)}, \mathbb{Q}) \leq \delta$$

where ϵ and δ are user-defined tolerances.

11.2.1 Descent Methods

$$x^{(i+1)} = x^{(i)} + h^{(i)} \Delta x^{(i)} \text{ with } f(x^{(i+1)}) < f(x^{(i)})$$

- Δx is the **step** or **search direction**.
- $h^{(i)}$ is the **step size** or **step length**.
- $f(x^{(i+1)}) < f(x^{(i)})$, i.e. $\Delta x^{(i)}$ is a **descent direction**.
- There exists a $h^{(i)} > 0$ s.t. $f(x^{(i+1)} < f(x^{(i)})$if $\nabla f(x^{(i)}) \Delta x^{(i)} < 0$.

Input: starting point $x^{(0)} \in$ domain of f
repeat
1. Compute a *descent direction* $\Delta x^{(i)}$
2. *Line search*: Choose step size $h^{(i)} > 0$ such that $f(x^{(i)} + h^{(i)} \Delta x^{(i)}) < f(x^{(i)})$
3. *Update* $x^{(i+1)} := x^{(i)} + h \Delta x^{(i)}$

until $f(x^{(m)}) - f(x^*) \leq \epsilon_1$ *or* $||x^{(m)} - x^{(m-1)}|| \leq \epsilon_2$

11.2.2 Gradient Methods

Idea: Gradient ∇f gives direction of steepest local ascent. $\Rightarrow$ Make steps of size h into anti-gradient direction.

$$\boxed{x^{(i+1)} = x^{(i)} - h^{(i)} \nabla f(x^{(i)})}$$

11.2.3 Interior-point Methods

Constrained Minimization Problem:

$$\min f(x)$$
$$\text{s.t. } g_i(x) \leq 0, \; i = 1 \ldots m$$

Assumptions:
- f, g_i convex, twice continuously differentiable
- $f(x^*)$ is finite and attained
- strict feasiblity: there exists a $\tilde{x}$ with
$$\tilde{x} \in \mathrm{dom} f, \; g_i(\tilde{x}) < 0, \; i = 1 \ldots m$$
- feasible set is closed an compact

12 Nonlinear MPC

- Presented assumptions on the terminal set and cost did not rely on linearity
- Lyapunov stability is a general framework to analyze stability of nonlinear dynamic systems
- **Results can be directly extended to nonlinear systems**
- Computing the sets $\mathcal{X}_f$ and function l_f can be very difficult.

Practical approaches include:
- Choose zero terminal constraint (no terminal cost needed)
- Linearization (for quadratic cost)
  - Linearize system around origin, assuming the linearization is stabilizable.
  - Design auxiliary controller $\kappa_f(x) = Kx$, terminal cost $l_f(x) = x^T P x$ and constraint set $\mathcal{X}_f = \{x | x^T P x \leq \alpha\}$ for linearized system s.t.
    - $l_f(A' + B'K)x) - l_f(x) = -2x^T(Q + K^T RK)x \; \forall x \in \mathcal{X}_f$
    - All state and input constraints are satisfied in $\mathcal{X}_f$
    - α is small enough such that

$$\begin{aligned}
&l_f(g(x, Kx)) - l_f((A' + B'K)x) \\
&\quad \leq x^T(Q + K^T RK)x \qquad \forall x \in \mathcal{X}_f\\
\Rightarrow\; &l_f(g(x, Kx)) - l_f(x)\\
&\quad \leq -x^T(Q + K^T RK)x \qquad \forall x \in \mathcal{X}_f
\end{aligned}$$

Terminal cost is a Lyapunov function in the terminal set and terminal set is invariant also for the nonlinear system.
- At each time step: Linearize the system around a trajectory (usually solution from previous time step). Solve convex problem.
- Solve nonlinear program:
  - Sequential quadratic programming
    Solvers: SNOPT, ACADO, NPSOL, KNITRO
  - Interior-point method
    Solvers: IPOPT, FORCES Pro, KNITRO
  - Non-convex problem, convergence only to locally optimal solution (under some assumptions)