CS159 Lecture 4 Supplementary Material: Synthesizing Terminal Components for MPC

Ugo Rosolia

Caltech

Spring 2021

Adapted from Berkeley ME231A
Original slide set by F. Borrelli, M. Morari, C. Jones

Table of Contents

Table of Contents

Polyhedra and polytopes

Polyhedra and polytopes

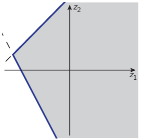
A polyhedron is the intersection of a *finite* number of closed halfspaces:

$$
\begin{aligned}
Z &= \{z \mid a_1^\top z \le b_1,\ a_2^\top z \le b_2, \dots, a_m^\top z \le b_m\} \\
&= \{z \mid Az \le b\}
\end{aligned}
$$

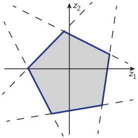
where $A := [a_1, a_2, \dots, a_m]^\top$ and $b := [b_1, b_2, \dots, b_m]^\top$.

A polytope is a *bounded* polyhedron.

Polyhedra and polytopes are always convex.



An (unbounded) polyhedron



A polytope

Polyhedra Representations

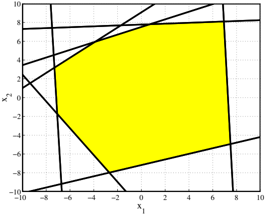
- An $\mathcal{H}$*-polyhedron* $\mathcal{P}$ in $\mathbb{R}^n$ denotes an intersection of a finite set of closed halfspaces in $\mathbb{R}^n$:

$$\mathcal{P} = \{x \in \mathbb{R}^n : \; Ax \leq b\}$$

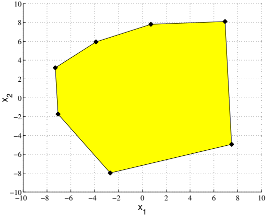
- A $\mathcal{V}$*-polytope* $\mathcal{P}$ in $\mathbb{R}^n$ is defined as

$$\mathcal{P} = \mathrm{conv}(V) = \{v \in \mathbb{R}^n | \exists \lambda \in \mathbb{R}^k, v = V\lambda, 1_k^\top \lambda = 1, \lambda \geq 0\}$$

for some $V = [V_1, \ldots, V_k] \in \mathbb{R}^{n \times k}$ and the vector of ones $1_k \in \mathbb{R}^k$.



$\mathcal{H}$-representation



$\mathcal{V}$-representation

Table of Contents

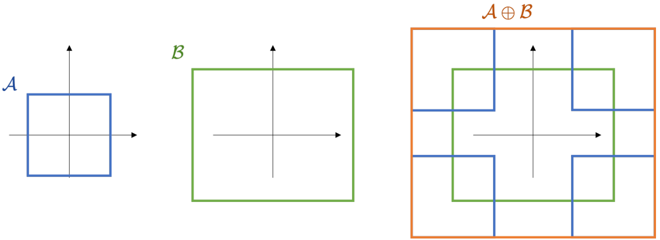
Basic Operations on Polytopes

- Given two sets $\mathcal{A} \subset \mathbb{R}^n$ and $\mathcal{B} \subset \mathbb{R}^n$, the Minkowski sum of $\mathcal{A}$ and $\mathcal{B}$ is defined as

$$\mathcal{A} \oplus \mathcal{B} = \{x + y \in \mathbb{R}^n \mid x \in \mathcal{A}, y \in \mathcal{B}\}$$

Furthermore, given the V-representations $\mathcal{A} = \text{conv}([v_1^a, \ldots, v_a^a])$ and $\mathcal{B} = \text{conv}([v_1^b, \ldots, v_b^b])$ the Minkowski sum

$$\mathcal{A} \oplus \mathcal{B} = \mathcal{A} = \text{conv}([v_{1,1}^{ab}, \ldots, v_{a,b}^{ab}]), \quad \text{where } v_{ij}^{ab} = v_i^a + v_j^b.$$



Basic Operations on Polytopes

- *Projection* Given a polytope $\mathcal{P} = \{[x'y']' \in \mathbb{R}^{n+m} \;:\; A^x x + A^y y \leq b\} \subset \mathbb{R}^{n+m}$ the projection onto the x-space $\mathbb{R}^n$ is defined as

$$\mathrm{proj}_x(\mathcal{P}) := \{x \in \mathbb{R}^n \mid \exists y \in \mathbb{R}^m : A^x x + A^y y \leq b\}.$$

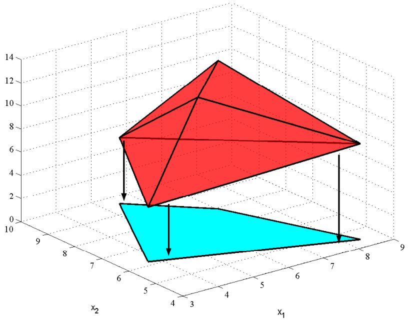


Table of Contents

Reachable Set

Reachable Set for a policy π

Consider the discrete-time system $x_{k+1} = f(x_k, u_k)$ and the state constraint set $\mathcal{X}$. The *Reachable Set for a policy π* from the set $\mathcal{S}$ is defined as

$$\mathrm{Reach}_\pi(\mathcal{S}) \triangleq \{x \in \mathbb{R}^n \mid \exists\ x_0 \in \mathcal{S} \text{ s.t. } x = f(x_0, \pi(x_0))\}$$

Reachable Set

Consider the discrete-time system $x_{k+1} = f(x_k, u_k)$, the state constraint set $\mathcal{X}$ and input constraint set $\mathcal{U}$. The *Reachable Set* from the set $\mathcal{S}$ is defined as

$$\mathrm{Reach}(\mathcal{S}) \triangleq \{x \in \mathbb{R}^n \mid \exists\ x_0 \in \mathcal{S},\ \exists\ u_0 \in \mathcal{U} \text{ s.t. } x = f(x_0, u_0)\}$$

Reachable Set

Reachable Set for a policy π

Consider the discrete-time system $x_{k+1} = f(x_k, u_k)$ and the state constraint set $\mathcal{X}$. The *Reachable Set for a policy* π from the set $\mathcal{S}$ is defined as

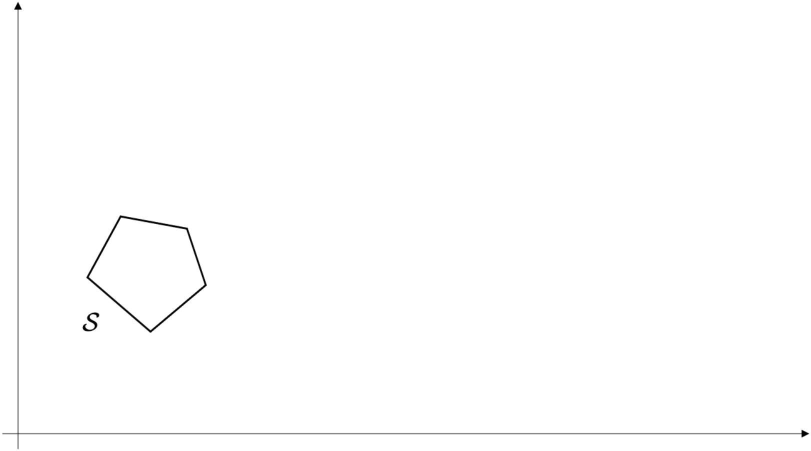
$$\text{Reach}_\pi(\mathcal{S}) \triangleq \{x \in \mathbb{R}^n \mid \exists\, x_0 \in \mathcal{S} \text{ s.t. } x = f(x_0, \pi(x_0))\}$$

Reachable Set

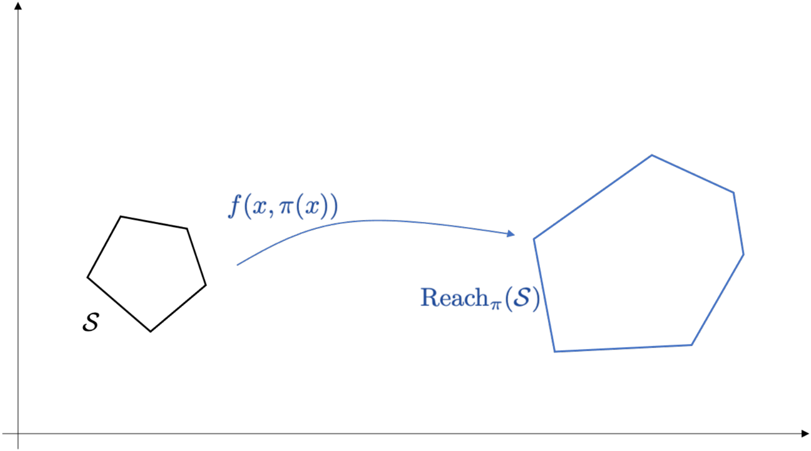
Consider the discrete-time system $x_{k+1} = f(x_k, u_k)$, the state constraint set $\mathcal{X}$ and input constraint set $\mathcal{U}$. The *Reachable Set* from the set $\mathcal{S}$ is defined as

$$\text{Reach}(\mathcal{S}) \triangleq \{x \in \mathbb{R}^n \mid \exists\, x_0 \in \mathcal{S},\ \exists\, u_0 \in \mathcal{U} \text{ s.t. } x = f(x_0, u_0)\}$$

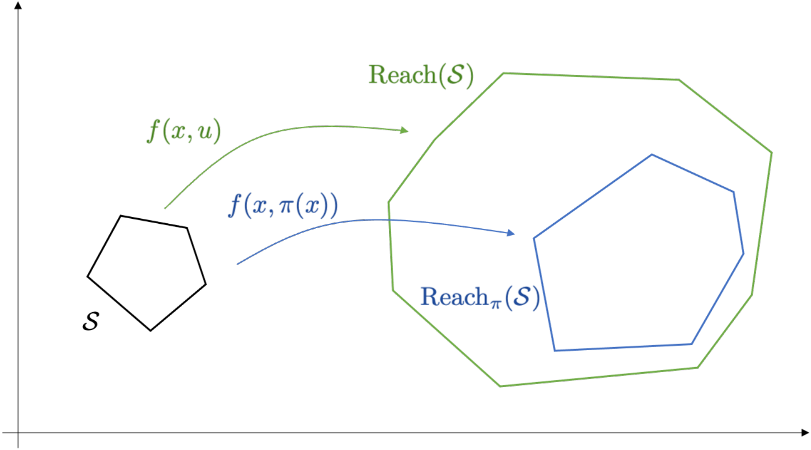
Reachable Set – Example



Reachable Set – Example



Reachable Set – Example



Reach Set Computation

- Consider the polyhedron $\mathcal{X} = \text{conv}(V_x)$ and the linear discrete time system
$$x(t+1) = Ax(t) + Bu(t)$$
where the input $u \in \mathcal{U} = \text{conv}(V_u)$ and define
$$A \circ \mathcal{X} = \mathrm{conv}(AV_x).$$

- Then for the policy $\pi(x) = Kx$
$$\text{Reach}_\pi(\mathcal{X}) = (A - BK) \circ \mathcal{X}$$
and
$$\begin{aligned} \text{Reach}(\mathcal{X}) &= \{\bar{x} + \bar{u} \mid \bar{x} \in A \circ V_x,\ \bar{u} \in B \circ \mathcal{U}\} \\ &= (A \circ \mathcal{X}) \oplus (B \circ \mathcal{U}). \end{aligned}$$

N-Step Reachable Sets

Definition (N-Step Reachable Set $\mathcal{R}_N(\mathcal{S})$)

Consider the discrete-time system $x_{k+1} = f(x_k, u_k)$, the state constraint set $\mathcal{X}$ and input constraint set $\mathcal{U}$. For a given initial set $\mathcal{S} \subseteq \mathcal{X}$, the N-step reachable set $\mathcal{R}_N(\mathcal{S})$ is

$$\mathcal{R}_{i+1}(\mathcal{S}) \triangleq \text{Reach}(\mathcal{R}_i(\mathcal{S})), \quad \mathcal{R}_0(\mathcal{S}) = \mathcal{S}, \qquad i = 0, \ldots, N-1$$

By definition all states $x_0 \in \mathcal{S}$ will evolve to the N-step reachable set $\mathcal{R}_N(\mathcal{S})$ in N time steps.

N-Step Reachable Sets

Definition (N-Step Reachable Set $\mathcal{R}_N(\mathcal{S})$)

Consider the discrete-time system $x_{k+1} = f(x_k, u_k)$, the state constraint set $\mathcal{X}$ and input constraint set $\mathcal{U}$. For a given initial set $\mathcal{S} \subseteq \mathcal{X}$, the N-step reachable set $\mathcal{R}_N(\mathcal{S})$ is

$$\mathcal{R}_{i+1}(\mathcal{S}) \triangleq \text{Reach}(\mathcal{R}_i(\mathcal{S})), \quad \mathcal{R}_0(\mathcal{S}) = \mathcal{S}, \qquad i = 0, \ldots, N-1$$

By definition all states $x_0 \in \mathcal{S}$ will evolve to the N-step reachable set $\mathcal{R}_N(\mathcal{S})$ in N time steps.

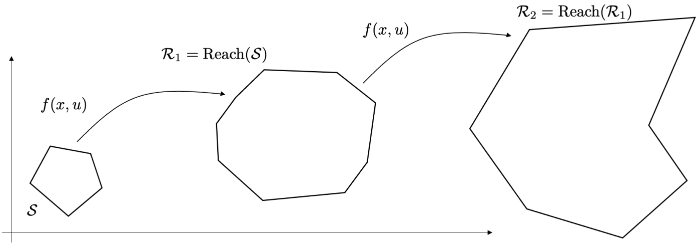


Table of Contents

Pre Set Definition

Pre Set for the policy π

Consider the discrete-time system $x_{k+1} = f(x_k, u_k)$ and the state constraint set $\mathcal{X}$. The *Pre Set for a policy π* from the set $\mathcal{S}$ is defined as

$$\text{Pre}_\pi(\mathcal{S}) \triangleq \{x \in \mathbb{R}^n \mid f(x, \pi(x)) \in \mathcal{S}\}$$

Pre Set

Consider the discrete-time system $x_{k+1} = f(x_k, u_k)$, the state constraint set $\mathcal{X}$ and input constraint set $\mathcal{U}$. The *Pre Set* from the set $\mathcal{S}$ is defined as

$$\text{Pre}_\pi(\mathcal{S}) \triangleq \{x \in \mathbb{R}^n \mid \exists u \in \mathcal{U}, f(x, u) \in \mathcal{S}\}$$

Pre Set Definition

Pre Set for the policy π

Consider the discrete-time system $x_{k+1} = f(x_k, u_k)$ and the state constraint set $\mathcal{X}$. The *Pre Set for a policy π* from the set $\mathcal{S}$ is defined as

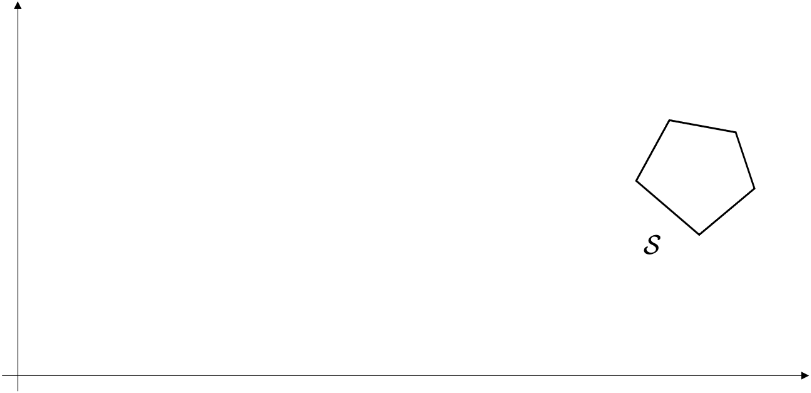
$$\text{Pre}_\pi(\mathcal{S}) \triangleq \{x \in \mathbb{R}^n \mid f(x, \pi(x)) \in \mathcal{S}\}$$

Pre Set

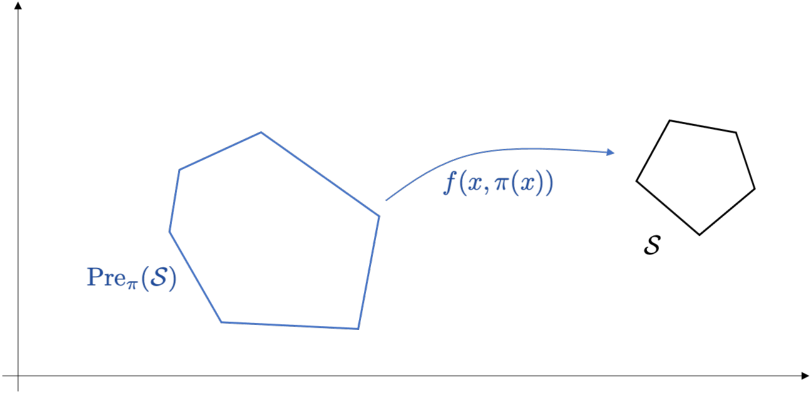
Consider the discrete-time system $x_{k+1} = f(x_k, u_k)$, the state constraint set $\mathcal{X}$ and input constraint set $\mathcal{U}$. The *Pre Set* from the set $\mathcal{S}$ is defined as

$$\text{Pre}_\pi(\mathcal{S}) \triangleq \{x \in \mathbb{R}^n \mid \exists u \in \mathcal{U}, f(x, u) \in \mathcal{S}\}$$

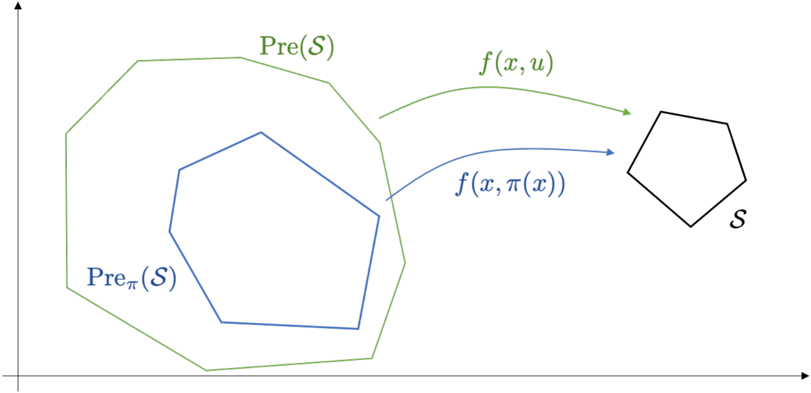
Pre Set – Example



Pre Set – Example



Pre Set – Example



Pre Set Computation - Autonomous Systems

- Consider the polyhedron $\mathcal{X} = \{x \mid H_x x \leq h_x\}$ and the linear discrete time autonomous system

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

- Then

$$\mathrm{Pre}(\mathcal{S}) = \{x \mid HAx \leq h\}$$

Pre Set Computation - System with Inputs

- Consider the polyhedron $\mathcal{X} = \{x \mid H_x x \leq h_x\}$ and the linear discrete time system

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

where the input $u \in \mathcal{U} = \{u \mid H_u u \leq h_u\}$ and define

$$A \circ \mathcal{X} = \mathrm{conv}(AV_x).$$

- Then

$$\mathrm{Pre}(\mathcal{S}) = \left\{ x \in \mathbb{R}^n \,|\, \exists u \in \mathbb{R} \mid \begin{bmatrix} H_x A & H_x B \\ 0 & H_u \end{bmatrix} \begin{pmatrix} x \\ u \end{pmatrix} \leq \begin{bmatrix} h_x \\ h_u \end{bmatrix} \right\}$$

which is the projection onto the x-space (with dimension $\mathbb{R}^n$) of the polyhedron

$$\mathcal{T} := \left\{ \begin{bmatrix} HA & HB \\ 0 & H_u \end{bmatrix} \begin{pmatrix} x \\ u \end{pmatrix} \leq \begin{bmatrix} h \\ h_u \end{bmatrix} \right\}.$$

N-Step Controllable Sets

N-Step Controllable Set $\mathcal{K}_N(\mathcal{S})$

For a given target set $\mathcal{S} \subseteq \mathcal{X}$, the N-step controllable set $\mathcal{K}_N(\mathcal{S})$ is defined as:

$$\mathcal{K}_N(\mathcal{S}) \triangleq \text{Pre}(\mathcal{K}_{N-1}(\mathcal{S})) \cap \mathcal{X}, \quad \mathcal{K}_0(\mathcal{S}) = \mathcal{S}, \qquad N \in \mathbb{N}^+.$$

By definition all states $x_0 \in \mathcal{K}_N(\mathcal{S})$ can be driven, through a time-varying control law, to the target set $\mathcal{O}$ in N steps, while satisfying input and state constraints.

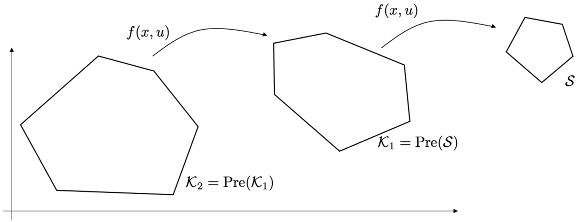
N-Step Controllable Sets

N-Step Controllable Set $\mathcal{K}_N(\mathcal{S})$

For a given target set $\mathcal{S} \subseteq \mathcal{X}$, the N-step controllable set $\mathcal{K}_N(\mathcal{S})$ is defined as:

$$\mathcal{K}_N(\mathcal{S}) \triangleq \text{Pre}(\mathcal{K}_{N-1}(\mathcal{S})) \cap \mathcal{X}, \quad \mathcal{K}_0(\mathcal{S}) = \mathcal{S}, \qquad N \in \mathbb{N}^+.$$

By definition all states $x_0 \in \mathcal{K}_N(\mathcal{S})$ can be driven, through a time-varying control law, to the target set $\mathcal{O}$ in N steps, while satisfying input and state constraints.



Maximal Controllable Set

Maximal Controllable Set $\mathcal{K}_\infty(\mathcal{S})$

For a given target set $\mathcal{O} \subseteq \mathcal{X}$, the maximal controllable set $\mathcal{K}_\infty(\mathcal{S})$ for the system $x(t+1) = f(x(t), u(t))$ subject to the constraints $x(t) \in \mathcal{X},\ u(t) \in \mathcal{U}$ is the union of all N-step controllable sets contained in $\mathcal{X}$ ($N \in \mathbb{N}$).

As we will be discussing, Maximal Controllable Set characterize the MPC region of attraction. However, computing these set may be challenging as these sets are computed using projections.

Table of Contents

Invariant Sets

Invariant sets

- are computed for ***autonomous systems***
- for a *given* feedback controller $u = \pi(x)$, will contain the evolution of the system for all times.

Positive Invariant Set

A set $\mathcal{O} \subseteq \mathcal{X}$ is said to be a positive invariant set for the autonomous system $x(t+1) = f(x(t), \pi(x(t))$ subject to the constraints $x(t) \in \mathcal{X}$, if

$$x(0) \in \mathcal{O} \quad \Rightarrow \quad x(t) \in \mathcal{O}, \quad \forall t \in \mathbb{N}^+$$

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

The set $\mathcal{O}_\infty$ is the maximal invariant set if $\mathcal{O}_\infty$ is invariant and $\mathcal{O}_\infty$ contains all the invariant sets contained in $\mathcal{X}$.

Invariant Sets

Invariant sets

- are computed for ***autonomous systems***
- for a *given* feedback controller $u = \pi(x)$, will contain the evolution of the system for all times.

Positive Invariant Set

A set $\mathcal{O} \subseteq \mathcal{X}$ is said to be a positive invariant set for the autonomous system $x(t+1) = f(x(t), \pi(x(t))$ subject to the constraints $x(t) \in \mathcal{X}$, if

$$x(0) \in \mathcal{O} \quad \Rightarrow \quad x(t) \in \mathcal{O}, \quad \forall t \in \mathbb{N}^+$$

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

The set $\mathcal{O}_\infty$ is the maximal invariant set if $\mathcal{O}_\infty$ is invariant and $\mathcal{O}_\infty$ contains all the invariant sets contained in $\mathcal{X}$.

Invariant Sets

Theorem (Geometric condition for invariance)

A set $\mathcal{O}$ is a positive invariant set if and only if $\mathcal{O} \subseteq Pre_{\pi}(\mathcal{O})$

NOTE: $\mathcal{O} \subseteq \text{Pre}_{\pi}(\mathcal{O}) \Leftrightarrow \text{Pre}_{\pi}(\mathcal{O}) \cap \mathcal{O} = \mathcal{O}$

Algorithm

Input: System model f, control policy π, constraint set $\mathcal{X}$
Output: $\mathcal{O}_\infty$
1. Let $\Omega_0 = \mathcal{X}$
2. Let $\Omega_{k+1} = \text{Pre}_{\pi}(\Omega_k) \cap \Omega_k$
3. If $\Omega_{k+1} = \Omega_k$ **then** $\Omega_\infty \leftarrow \Omega_{k+1}$
4. If else go to 2

The algorithm generates the set sequence $\{\Omega_k\}$ satisfying $\Omega_{k+1} \subseteq \Omega_k, \forall k \in \mathbb{N}$ and it terminates when $\Omega_{k+1} = \Omega_k$ so that Ω_k is the maximal positive invariant set $\mathcal{O}_\infty$ for $x(t+1) = f_a(x(t))$.

Table of Contents

Control Invariant Sets

Control invariant sets

- are computed for systems ***subject to external inputs***
- provide the set of initial states for which *there exists* a controller such that the system constraints are never violated.

Control Invariant Set

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

$$x(t) \in \mathcal{C} \quad \Rightarrow \quad \exists u(t) \in \mathcal{U} \text{ such that } f(x(t), u(t)) \in \mathcal{C}, \quad \forall t \in \mathbb{N}^+$$

Maximal Control Invariant Set

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

Control Invariant Sets

Control invariant sets

- are computed for systems ***subject to external inputs***
- provide the set of initial states for which *there exists* a controller such that the system constraints are never violated.

Control Invariant Set

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

$$x(t) \in \mathcal{C} \quad \Rightarrow \quad \exists u(t) \in \mathcal{U} \text{ such that } f(x(t), u(t)) \in \mathcal{C}, \quad \forall t \in \mathbb{N}^+$$

Maximal Control Invariant Set

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

Control Invariant Sets

Same geometric condition for control invariants holds: $\mathcal{C}$ is a control invariant set if and only if

$$\mathcal{C} \subseteq \mathrm{Pre}(\mathcal{C})$$

Algorithm

Input: System model f, constraint sets $\mathcal{X}$ and $\mathcal{U}$
Output: $\mathcal{O}_\infty$
1. Let $\Omega_0 = \mathcal{X}$
2. Let $\Omega_{k+1} = \mathrm{Pre}(\Omega_k) \cap \Omega_k$
3. If $\Omega_{k+1} = \Omega_k$ **then** $\mathcal{C}_\infty \leftarrow \Omega_{k+1}$
4. If else go to 2

The algorithm generates the set sequence $\{\Omega_k\}$ satisfying $\Omega_{k+1} \subseteq \Omega_k, \forall k \in \mathbb{N}$ and it terminates if $\Omega_{k+1} = \Omega_k$ so that Ω_k is the maximal control invariant set $\mathcal{C}_\infty$ for the constrained system.

Invariant Sets and Control Invariant Sets

TO DO: Add figure

- The set $\mathcal{O}_\infty$ ($\mathcal{C}_\infty$) is ***finitely determined*** if and only if $\exists\ i \in \mathbb{N}$ such that $\Omega_{i+1} = \Omega_i$.
- The smallest element $i \in \mathbb{N}$ such that $\Omega_{i+1} = \Omega_i$ is called the ***determinedness index***.
- For all states contained in the maximal control invariant set $\mathcal{C}_\infty$ there exists a control law, such that the system constraints are never violated.

Table of Contents

Loss of Feasibility

MPC policies compute control actions by solving finite time optimal control problems over shifted time windows:

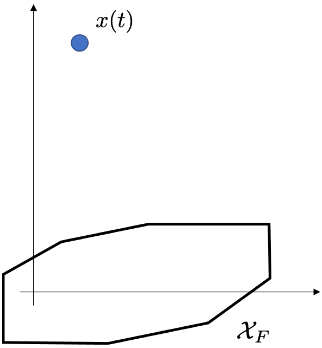
$$
\begin{aligned}
J_t^*(x(0)) = \min_{u_{t|t},\ldots,u_{t+N-1|t}} \quad & \sum_{k=0}^{T-1} h(x_{k|t}, u_{k|t}) + V(x_{t+T|t}) \\
\text{such that} \quad & x_{k+1|t} = Ax_{k|t} + Bu_{k|t}, \forall k \in \{t, \ldots, t+N-1\} \\
& x_{k|t} \in \mathcal{X}, u_{k|t} \in \mathcal{U}, \forall k \in \{t, \ldots, t+N-1\} \\
& x_{t|t} = x(0), x_N \in \mathcal{X}_F
\end{aligned}
$$

Solution: The terminal cost $V(x_{t+T|t})$ and terminal constraint $\mathcal{X}_F$, often referred to as *terminal components*, should approximate the tail of cost and constraints beyond the prediction horizon.

Recursive Feasibility

Let the terminal set $\mathcal{X}_F$ be *control invariant*. Next, we show by induction that the terminal set $\mathcal{X}_F$ guarantees that the controller is recursively feasible.

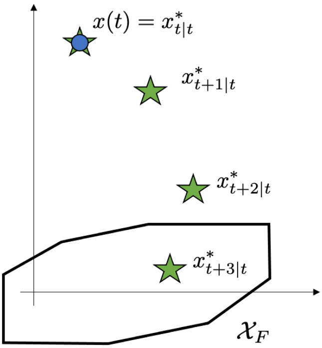
- At time step t assume that the MPC problem is feasible and let $\{u^*_{t|t}, \ldots, u^*_{t+N-1|t}\}$ and $\{x^*_{t|t}, \ldots, x^*_{t+N|t}\}$ be the optimal sequences of states and actions.
- At the next time step $t+1$, we have that $x(t+1) = x^*_{t+1|t}$.
- Therefore, at the next time step $t+1$ the sequences $\{u^*_{t+1|t}, \ldots, u^*_{t+N-1|t}, 0\}$ and $\{x^*_{t+1|t}, \ldots, x^*_{t+N|t}, 0\}$ are feasible, as $x^*_{t+N|t} = 0 \in \mathcal{X}_F$ and the origin is an unforced equilibrium point.



Recursive Feasibility

Let the terminal set $\mathcal{X}_F$ be *control invariant*. Next, we show by induction that the terminal set $\mathcal{X}_F$ guarantees that the controller is recursively feasible.

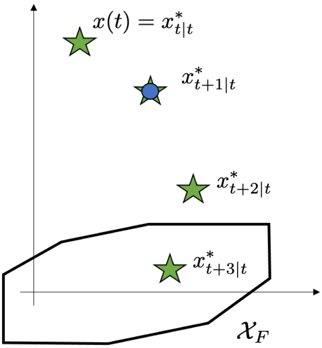
- At time step t assume that the MPC problem is feasible and let $\{u^*_{t|t}, \ldots, u^*_{t+N-1|t}\}$ and $\{x^*_{t|t}, \ldots, x^*_{t+N|t}\}$ be the optimal sequences of states and actions.
- At the next time step $t+1$, we have that $x(t+1) = x^*_{t+1|t}$.
- Therefore, at the next time step $t+1$ the sequences $\{u^*_{t+1|t}, \ldots, u^*_{t+N-1|t}, 0\}$ and $\{x^*_{t+1|t}, \ldots, x^*_{t+N|t}, 0\}$ are feasible, as $x^*_{t+N|t} = 0 \in \mathcal{X}_F$ and the origin is an unforced equilibrium point.



Recursive Feasibility

Let the terminal set $\mathcal{X}_F$ be *control invariant*. Next, we show by induction that the terminal set $\mathcal{X}_F$ guarantees that the controller is recursively feasible.

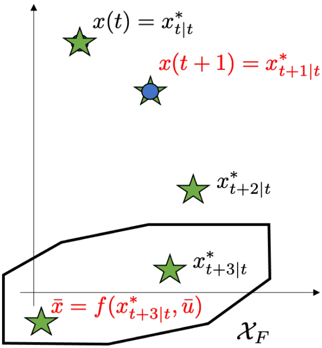
- At time step t assume that the MPC problem is feasible and let $\{u^*_{t|t}, \dots, u^*_{t+N-1|t}\}$ and $\{x^*_{t|t}, \dots, x^*_{t+N|t}\}$ be the optimal sequences of states and actions.
- At the next time step $t+1$, we have that $x(t+1) = x^*_{t+1|t}$.
- Therefore, at the next time step $t+1$ the sequences $\{u^*_{t+1|t}, \dots, u^*_{t+N-1|t}, 0\}$ and $\{x^*_{t+1|t}, \dots, x^*_{t+N|t}, 0\}$ are feasible, as $x^*_{t+N|t} = 0 \in \mathcal{X}_F$ and the origin is an unforced equilibrium point.



Recursive Feasibility

Let the terminal set $\mathcal{X}_F$ be <u>*control invariant*</u>. Next, we show by induction that the terminal set $\mathcal{X}_F$ guarantees that the controller is recursively feasible.

- At time step t assume that the MPC problem is feasible and let $\{u^*_{t|t}, \ldots, u^*_{t+N-1|t}\}$ and $\{x^*_{t|t}, \ldots, x^*_{t+N|t}\}$ be the optimal sequences of states and actions.
- At the next time step $t+1$, we have that $x(t+1) = x^*_{t+1|t}$.
- As $x^*_{t+N|t} \in \mathcal{X}_F$ there exists $\bar{u} \in \mathcal{U}$ such that $\bar{x} = f(x^*_{t+N|t}, \bar{u}) \in \mathcal{X}_F$. Thus, at the next time step $t+1$ the sequences $\{u^*_{t+1|t}, \ldots, u^*_{t+N-1|t}, \bar{u}\}$ and $\{x^*_{t+1|t}, \ldots, x^*_{t+N|t}, \bar{x}\}$ are feasible.



Stability – Assumptions

Let the following assumptions hold

- The stage cost satisfies

$$h(x,u) = 0 \forall x \in \mathcal{X} \setminus \{0\}, \forall u \in \mathcal{U} \setminus \{0\}$$

  and $h(0,0) = 0$.
- The terminal set $\mathcal{X}_F$ is a *control invariant set*
- The terminal cost function $V : \mathbb{R}^n \to \mathbb{R}$ is a control Lyapunov function for the set $\mathcal{X}_F$, i.e.,

$$\forall x \in \mathcal{X}_F, \exists u \in \mathcal{U} \text{ such that } V(f(x,u)) - V(x) \geq -h(x,u) \\ \text{and } f(x,u) \in \mathcal{X}_F.$$

Next, we show by induction that the open-loop cost $J_t^*(x(t))$ is a Lyapunov function for the closed-loop system, i.e.,

$$J_{t+1}^*(x(t+1)) < J_t^*(x(t)), \forall x(t) \in \mathcal{X} \setminus \{0\}.$$

Table of Contents

Stability – Proof (1/2)

- At time step t, assume that the MPC problem is feasible and let $\{u^*_{t|t}, \ldots, u^*_{t+N-1|t}\}$ and $\{x^*_{t|t}, \ldots, x^*_{t+N|t}\}$ be the optimal sequences of states and actions. Then the open-loop cost is

$$\begin{aligned} J^*_t(x(t)) &= \sum_{k=t}^{N-1} h(x^*_{k|t}, u^*_{k|t}) + V(x^*_{t+N|t}) \\ &\geq \sum_{k=t}^{N-1} h(x^*_{k|t}, u^*_{k|t}) + h(x^*_{t+N|t}, \bar{u}) + V(f(x^*_{t+N|t}, \bar{u})) \end{aligned}$$

for $\bar{u} \in \mathcal{U}$ such that $f(x^*_{t+N|t}, \bar{u})$.

Stability – Proof (2/2)

- At the next time step $t+1$,

$$\bar{J} = \sum_{k=t+1}^{N-1} h(x^*_{k|t}, u^*_{k|t}) + h(x^*_{t+N|t}, \bar{u}) + V(f(x^*_{t+N|t}, \bar{u}))$$

  is the cost associated with the feasible sequence of inputs $\{u^*_{t+1|t}, \dots, u^*_{t+N-1|t}, \bar{u}\}$, thus

$$J^*_t(x(t)) = h(x^*_{k|t}, u^*_{k|t}) + \bar{J} \geq h(x^*_{t|t}, u^*_{t|t}) + J^*_{t+1}(x(t+1)).$$

- Concluding, the open-loop cost satisfies

$$J^*_{t+1}(x(t+1)) - J^*_t(x(t)) \leq -h(x(t), u(t))$$

  as $x^*_{t|t} = x(t)$ and $u^*_{t|t} = x(t)$, and it is a Lyapunov function for the closed-loop system.

Table of Contents

Constrained Linear Quadratic Regulator

Consider the following finite time optimal control problem:

$$
\begin{aligned}
J_t^*(x(0)) = \min_{u_{t|t},\ldots,u_{t+N-1|t}} \quad & \sum_{k=0}^{T-1} h(x_{k|t}, u_{k|t}) + x_{t+T|t}^\top P x_{t+T|t} \\
\text{such that} \quad & x_{k+1|t} = Ax_{k|t} + Bu_{k|t}, \forall k \in \{t, \ldots, t+N-1\} \\
& x_{k|t} \in \mathcal{X}, u_{k|t} \in \mathcal{U}, \forall k \in \{t, \ldots, t+N-1\} \\
& x_{t|t} = x(0), x_N \in \mathcal{X}_F
\end{aligned}
$$

where $h(x, u) = x^\top Qx + u^\top Ru$.

Next, we discuss how to construct the terminal cost $V(x) = x^\top Px$ and the terminal set $\mathcal{X}_F$ to guarantee recursive feasibility and closed-loop stability.

Design Rules

1. Design unconstrained LQR control law

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

   where P_∞ is the solution to the discrete-time algebraic Riccati equation:

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

2. Choose the terminal weight $P = P_\infty$
3. Choose the terminal set $\mathcal{X}_F$ to be the maximum invariant set for the closed-loop system $x_{k+1} = (A - BK_\infty)x_k$:

$$x_{k+1} = (A - BK_\infty)x_k \in \mathcal{X}_F, \quad \text{for all } x_k \in \mathcal{X}_F$$

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

$$\mathcal{X}_F \subseteq \mathcal{X}, \; F_\infty x_k \in \mathcal{U}, \quad \text{for all } x_k \in \mathcal{X}_F$$

Stability and Feasibility Proof

By construction all the Assumptions of the required to guarantee recursive feasibility and stability are verified:

1. The stage cost is a positive definite function
2. By construction the terminal set is **invariant** under the local control law $v = -K_\infty x$
3. Terminal cost is a continuous **Lyapunov function** in the terminal set $\mathcal{X}_F$ and satisfies:

$$\begin{aligned}
&x_{k+1}^\top P x_{k+1} - x_k^\top P x_k \\
&\qquad = x_k'(-P_\infty + A'P_\infty A - A'P_\infty B(B'P_\infty B + R)^{-1}B'P_\infty A)x_k \\
&\qquad = -x_k' Q x_k
\end{aligned}$$

Summary of Safety and Stability Properties

Key Message: When the MPC terminal components are not designed correctly, the closed-loop system may violate safety constraints and convergence to the goal state/set is not guaranteed

Solution: We have shown that given a terminal set set $\mathcal{X}_F$ which is control invariant, and a terminal cost function $V(x)$ which is a control Lyapunov function.

- The MPC problem is feasible at all times
- The closed-loop system is stable as for the positive definite open-loop cost we have $J^*_{t+1}(x(t+1)) < J^*_t(x(t)), \forall x(t) \notin \mathcal{X}_F$

Main drawback: These terminal components are hard to compute even for linear constrained deterministic systems.