Model Predictive Control

Chapter 5: Invariance

Prof. Melanie Zeilinger

ETH Zurich

Spring 2020

Coauthors: Prof. Manfred Morari, University of Pennsylvania
Prof. Colin Jones, EPFL
Prof. Francesco Borrelli, UC Berkeley

Outline

Outline

1. Objectives of Constrained Control

2. Invariance

3. Controlled Invariance

4. Summary

5. Practical Computation of Invariant Sets

Constrained Control

$$x(k+1) = g(x(k), u(k)) \qquad\qquad x, u \in \mathcal{X}, \mathcal{U}$$

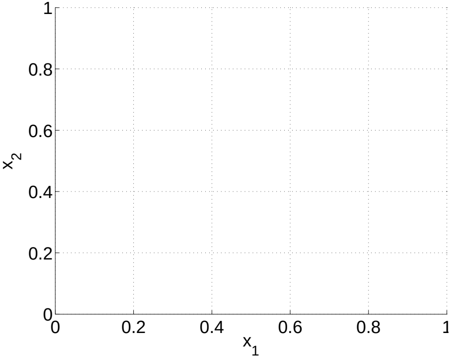
Design control law $u(k) = \kappa(x(k))$ such that the system:

1. Satifies constraints : $\{x(k)\} \subset \mathcal{X}$, $\{u(k)\} \subset \mathcal{U}$
2. Is stable: $\lim_{k\to\infty} x(k) = 0$
3. Optimizes "performance"
4. Maximizes the set $\{x(0) \,|\, \text{Conditions 1-3 are met}\}$

This lecture is about how to ensure #1

(Remaining lectures cover 2-4)

Limitations of Linear Controllers



System:

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

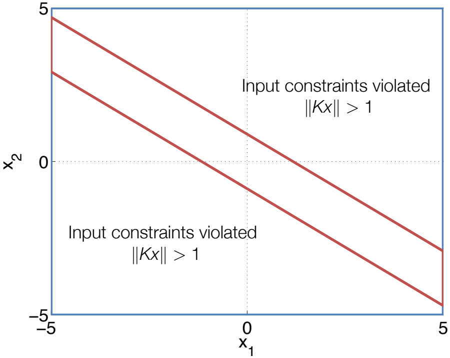
Constraints:

$$\mathcal{X} := \{x \mid \|x\|_\infty \le 5\}$$
$$\mathcal{U} := \{u \mid \|u\|_\infty \le 1\}$$

Consider an LQR controller, with $Q = I$, $R = 1$.

Does linear control work?

Limitations of Linear Controllers



System:

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

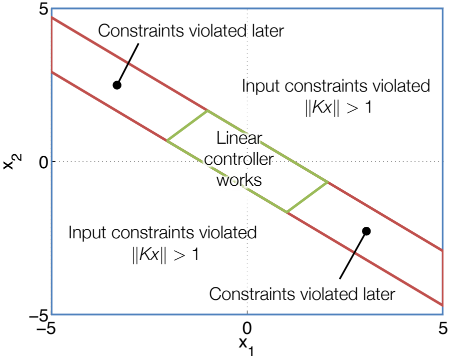
Constraints:

$$\mathcal{X} := \{x \mid \|x\|_\infty \le 5\}$$
$$\mathcal{U} := \{u \mid \|u\|_\infty \le 1\}$$

Consider an LQR controller, with $Q = I$, $R = 1$.

Does linear control work?

Limitations of Linear Controllers



System:

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

Constraints:

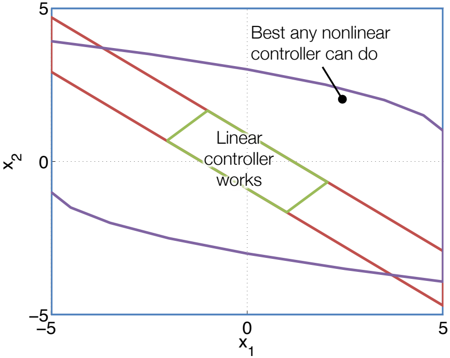
$$\mathcal{X} := \{x \mid \|x\|_\infty \leq 5\}$$
$$\mathcal{U} := \{u \mid \|u\|_\infty \leq 1\}$$

Consider an LQR controller, with $Q = I$, $R = 1$.

Does linear control work?

Yes, but the region where it works is very small

Limitations of Linear Controllers



System:

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

Constraints:

$$\mathcal{X} := \{x \mid \|x\|_\infty \leq 5\}$$
$$\mathcal{U} := \{u \mid \|u\|_\infty \leq 1\}$$

Consider an LQR controller, with $Q = I,\ R = 1$.

Does linear control work?

Yes, but the region where it works is very small

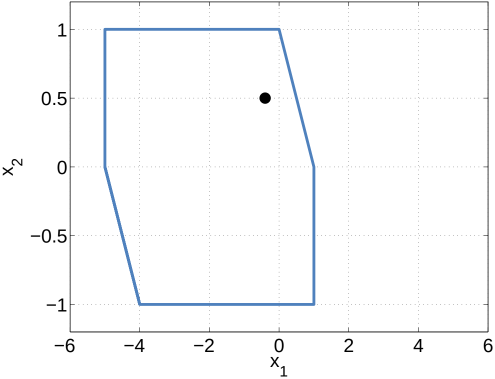
Use nonlinear control (MPC) to increase the region of attraction

Learning Objectives - Constrained Systems

- Learn definition and meaning of **invariance**
  - Region in which an **autonomous** system satisfies the constraints **for all time**
- Learn definition and meaning of **controlled invariance**
  - Region for which there **exists a controller** so that the system satisfies the constraints **for all time**
- Learn how to (conceptually) compute these sets
- Learn how to compute an ellipsoidal invariant set

Invariance: Which states are "good"?

The initial state is in the constraints. Is the next one?



System:

$$\dot{x} = \begin{bmatrix} -2\zeta\omega & -\omega^2 \\ 1 & 0 \end{bmatrix} x$$

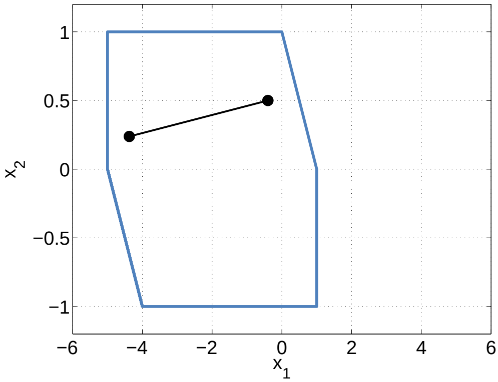
where $\omega = 10$, $\zeta = 0.01$, sampled at 10Hz.

Constraints:

$$\mathcal{X} := \left\{ x \,\middle|\, \begin{array}{l} -5 \leq x_1 \leq 1 \\ -1 \leq x_2 \leq 1 \\ -5 \leq x_1 + x_2 \leq 1 \end{array} \right\}$$

Invariance: Which states are "good"?

The initial state is in the constraints. Is the next one?
Yup, next one?



System:

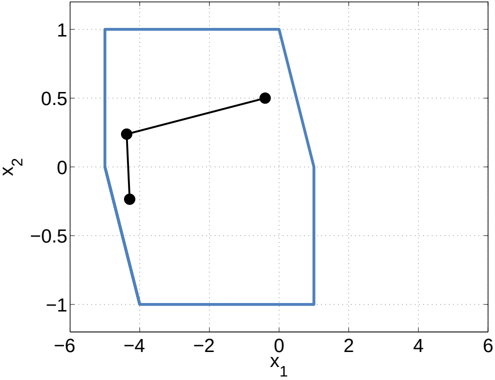
$$\dot{x} = \begin{bmatrix} -2\zeta\omega & -\omega^2 \\ 1 & 0 \end{bmatrix} x$$

where $\omega = 10$, $\zeta = 0.01$, sampled at 10Hz.
Constraints:

$$\mathcal{X} := \left\{ x \;\middle|\; \begin{array}{l} -5 \leq x_1 \leq 1 \\ -1 \leq x_2 \leq 1 \\ -5 \leq x_1 + x_2 \leq 1 \end{array} \right\}$$

Invariance: Which states are "good"?

The initial state is in the constraints. Is the next one?
Yup, next one? Yup...



System:

$$\dot{x} = \begin{bmatrix} -2\zeta\omega & -\omega^2 \\ 1 & 0 \end{bmatrix} x$$

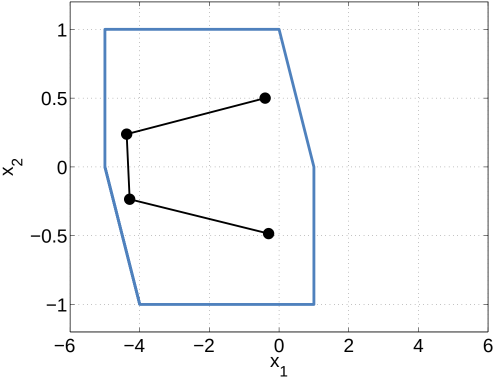
where $\omega = 10$, $\zeta = 0.01$, sampled at 10Hz.

Constraints:

$$\mathcal{X} := \left\{ x \,\middle|\, \begin{array}{l} -5 \leq x_1 \leq 1 \\ -1 \leq x_2 \leq 1 \\ -5 \leq x_1 + x_2 \leq 1 \end{array} \right\}$$

Invariance: Which states are "good"?

The initial state is in the constraints. Is the next one?
Yup, next one? Yup... Yup...



System:

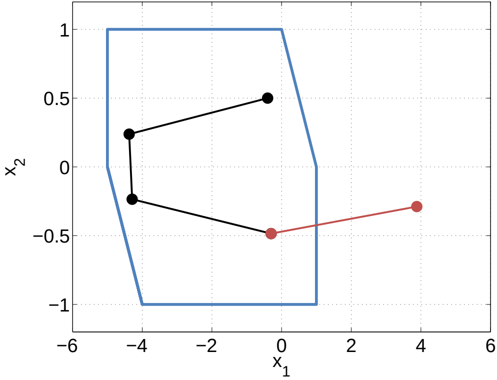
$$\dot{x} = \begin{bmatrix} -2\zeta\omega & -\omega^2 \\ 1 & 0 \end{bmatrix} x$$

where $\omega = 10$, $\zeta = 0.01$, sampled at 10Hz.
Constraints:

$$\mathcal{X} := \left\{ x \;\middle|\; \begin{array}{l} -5 \le x_1 \le 1 \\ -1 \le x_2 \le 1 \\ -5 \le x_1 + x_2 \le 1 \end{array} \right\}$$

Invariance: Which states are "good"?

The initial state is in the constraints. Is the next one?
Yup, next one? Yup... Yup... Uh oh



System:

$$\dot{x} = \begin{bmatrix} -2\zeta\omega & -\omega^2 \\ 1 & 0 \end{bmatrix} x$$

where $\omega = 10$, $\zeta = 0.01$, sampled at 10Hz.

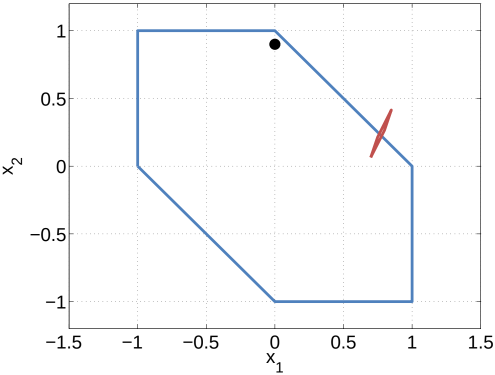
Constraints:

$$\mathcal{X} := \left\{ x \;\middle|\; \begin{aligned} -5 \le x_1 \le 1 \\ -1 \le x_2 \le 1 \\ -5 \le x_1 + x_2 \le 1 \end{aligned} \right\}$$

Look an infinite distance into the future to determine if the trajectory beginning at the current state always remains in the constraints.

Controlled Invariance: Does a good input exist?

The initial state is in the constraints. Can we choose the next one to be?



System:

$$x(k+1) = 0.9 \begin{bmatrix} \sin(0.3) & \cos(0.3) \\ -\cos(0.3) & \sin(0.3) \end{bmatrix} x(k) + 0.25 \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} u(k)$$

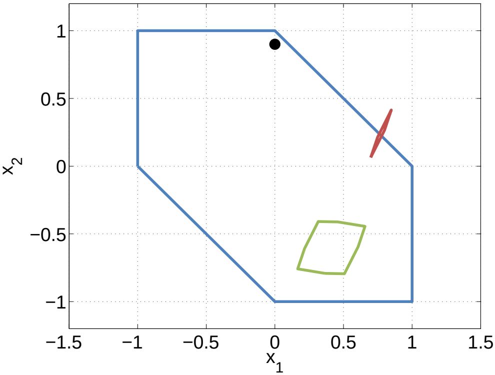
Constraints:

$$\|u(k)\|_\infty \leq 0.1$$
$$\|x(k)\|_\infty \leq 1$$
$$\| \begin{bmatrix} 1 & 1 \end{bmatrix} x(k) \|_\infty \leq 1$$

We can choose from a set of inputs $\Rightarrow$ Set of possible next states

Controlled Invariance: Does a good input exist?

The initial state is in the constraints. Can we choose the next one to be?



System:

$$x(k+1) = 0.9 \begin{bmatrix} \sin(0.3) & \cos(0.3) \\ -\cos(0.3) & \sin(0.3) \end{bmatrix} x(k) + 0.25 \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} u(k)$$

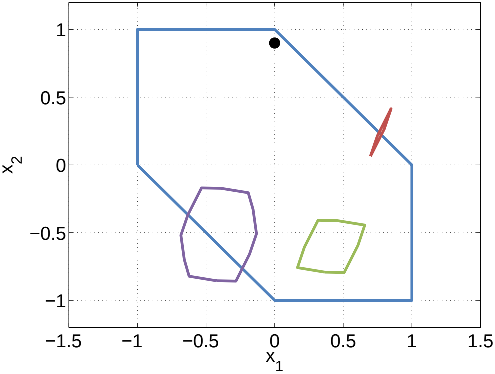
Constraints:

$$\|u(k)\|_\infty \leq 0.1$$
$$\|x(k)\|_\infty \leq 1$$
$$\| \begin{bmatrix} 1 & 1 \end{bmatrix} x(k) \|_\infty \leq 1$$

We can choose from a set of inputs $\Rightarrow$ Set of possible next states

Controlled Invariance: Does a good input exist?

The initial state is in the constraints. Can we choose the next one to be?



System:

$$x(k+1) = 0.9 \begin{bmatrix} \sin(0.3) & \cos(0.3) \\ -\cos(0.3) & \sin(0.3) \end{bmatrix} x(k) + 0.25 \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} u(k)$$

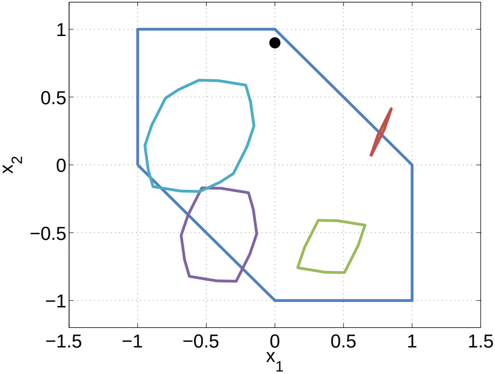
Constraints:

$$\|u(k)\|_\infty \leq 0.1$$
$$\|x(k)\|_\infty \leq 1$$
$$\|\begin{bmatrix} 1 & 1 \end{bmatrix} x(k)\|_\infty \leq 1$$

We can choose from a set of inputs ⇒ Set of possible next states

Controlled Invariance: Does a good input exist?

The initial state is in the constraints. Can we choose the next one to be?



System:

$$x(k+1) = 0.9 \begin{bmatrix} \sin(0.3) & \cos(0.3) \\ -\cos(0.3) & \sin(0.3) \end{bmatrix} x(k) + 0.25 \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} u(k)$$

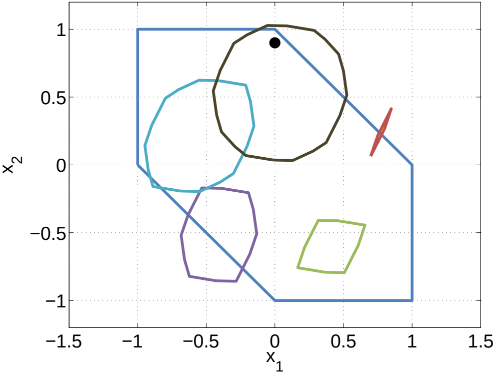
Constraints:

$$\|u(k)\|_\infty \leq 0.1$$
$$\|x(k)\|_\infty \leq 1$$
$$\| \begin{bmatrix} 1 & 1 \end{bmatrix} x(k)\|_\infty \leq 1$$

We can choose from a set of inputs $\Rightarrow$ Set of possible next states

Controlled Invariance: Does a good input exist?

The initial state is in the constraints. Can we choose the next one to be?



System:

$$x(k+1) = 0.9 \begin{bmatrix} \sin(0.3) & \cos(0.3) \\ -\cos(0.3) & \sin(0.3) \end{bmatrix} x(k) + 0.25 \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} u(k)$$

Constraints:

$$\|u(k)\|_\infty \leq 0.1$$

$$\|x(k)\|_\infty \leq 1$$

$$\| \begin{bmatrix} 1 & 1 \end{bmatrix} x(k) \|_\infty \leq 1$$

We can choose from a set of inputs $\Rightarrow$ Set of possible next states

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

Outline

Invariance

Constraint satisfaction, for an **autonomous** system $x(k+1) = g(x(k))$, or **closed-loop** system $x(k+1) = g(x(k), \kappa(x(k)))$ for a **given** controller κ.

Positive Invariant set
A set $\mathcal{O}$ is said to be a positive invariant set for the autonomous system $x(k+1) = g(x(k))$ if $$x(k) \in \mathcal{O} \;\; \Rightarrow \;\; x(k+1) \in \mathcal{O} \;\; , \;\; \forall k \in \{0, 1, \ldots\}$$

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.

Maximal Positive Invariant Set $\mathcal{O}_\infty$
The set $\mathcal{O}_\infty \subset \mathcal{X}$ is the maximal invariant set with respect to $\mathcal{X}$ if $0 \in \mathcal{O}_\infty$, $\mathcal{O}_\infty$ is invariant and $\mathcal{O}_\infty$ contains all invariant sets that contain the origin.

The maximal invariant set is the set of all states for which the system will remain feasible if it starts in $\mathcal{O}_\infty$.

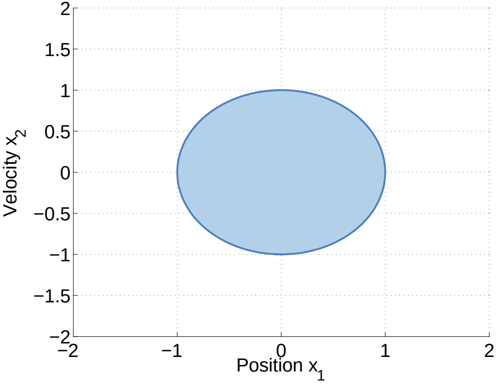
Pre-Sets

Pre Set

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

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

Pre-Set Example : Pendulum



Pendulum:

$$x(k+1) = x(k) + \begin{bmatrix} x_2(k) \\ -9.8\sin x_1(k) - x_2(k) \end{bmatrix}$$

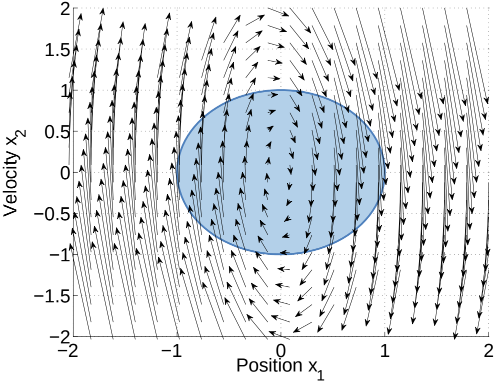
(Discretized with forward Euler at 1Hz)

Target set:

$$T := \{x \mid \|x\|_2 \le 1\}$$

Which states will be in the target set at the next point in time?

Pre-Set Example : Pendulum



Pendulum:

$$x(k+1) = x(k) + \begin{bmatrix} x_2(k) \\ -9.8\sin x_1(k) - x_2(k) \end{bmatrix}$$

(Discretized with forward Euler at 1Hz)

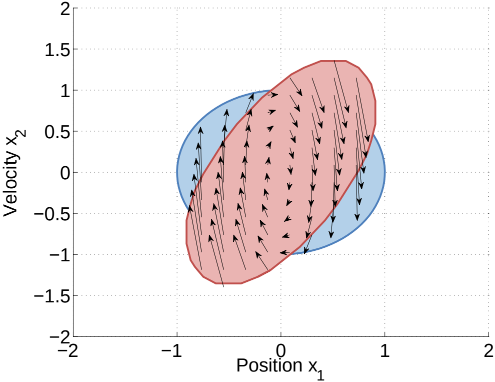
Target set:

$$T := \{x \mid \|x\|_2 \leq 1\}$$

Which states will be in the target set at the next point in time?

Consider the phase diagram.

Pre-Set Example : Pendulum



Pendulum:

$$x(k+1) = x(k) + \begin{bmatrix} x_2(k) \\ -9.8 \sin x_1(k) - x_2(k) \end{bmatrix}$$

(Discretized with forward Euler at 1Hz)

Target set:

$$T := \{x \mid \|x\|_2 \leq 1\}$$

Which states will be in the target set at the next point in time?

Consider the phase diagram.

Pre-set is those states that will be in T in one time-step

Extremely difficult to compute, except in special cases.

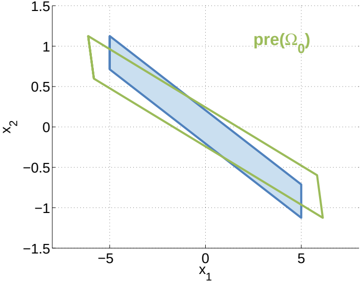
Pre-Set Computation: Linear Autonomous System

Pre Set

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

$$\mathrm{pre}(S) := \{x \,|\, Ax \in S\}$$

If $S := \{x \,|\, Fx \leq f\}$, then $\mathrm{pre}(S) = \{x \,|\, FAx \leq f\}$



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

$$\begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix}, \quad \|u(k)\|_\infty \leq 0.1$$

Where $u(k) = Kx(k)$, with K the optimal LQR controller for $Q = I$, $R = 90$.

Invariant Set Conditions

Theorem: Geometric condition for invariance

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

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

We prove the contrapositive for both the necessary and sufficient conditions.

Necessary If $\mathcal{O} \nsubseteq \mathsf{pre}(\mathcal{O})$, then $\exists \bar{x} \in \mathcal{O}$ such that $\bar{x} \notin \mathsf{pre}(\mathcal{O})$. From the definition of $\mathsf{pre}(\mathcal{O})$, $g(\bar{x}) \notin \mathcal{O}$ and thus $\mathcal{O}$ is not a positive invariant set.

Sufficient If $\mathcal{O}$ is not a positive invariant set, then $\exists \bar{x} \in \mathcal{O}$ such that $g(\bar{x}) \notin \mathcal{O}$. This implies that $\bar{x} \in \mathcal{O}$ and $\bar{x} \notin \mathsf{pre}(\mathcal{O})$ and thus $\mathcal{O} \nsubseteq \mathsf{pre}(\mathcal{O})$.

□

Note that $\mathcal{O} \subseteq \mathsf{pre}(\mathcal{O}) \Leftrightarrow \mathsf{pre}(\mathcal{O}) \cap \mathcal{O} = \mathcal{O}$

Computing Invariant Sets

Conceptual Algorithm to Compute Invariant Set

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

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

Computing Invariant Sets

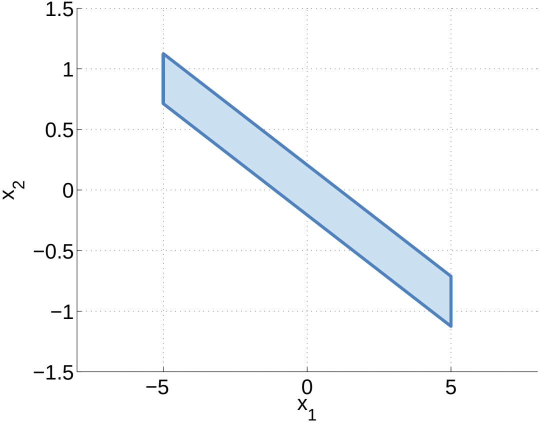
Conceptual Algorithm to Compute Invariant Set

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

Requirements:

- Represent set Ω_i (Often as polytopes)
- Pre-set computation
- Intersection
- Equality test (bi-directional subset)

Computing Invariant Sets



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

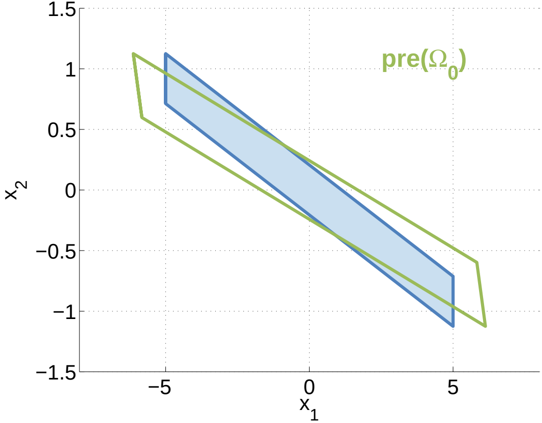
$\Omega_0 \leftarrow \mathcal{X}$
loop
  $\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$
  if $\Omega_{i+1} = \Omega_i$ **then**
    return $\mathcal{O}_\infty = \Omega_i$
  end if
end loop

System: $x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k)$, $\begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix}$

$-0.1 \leq u(k) \leq 0.1$

Where $u(k) = Kx(k)$, with K the optimal LQR controller for $Q = I$, $R = 90$.

Computing Invariant Sets



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

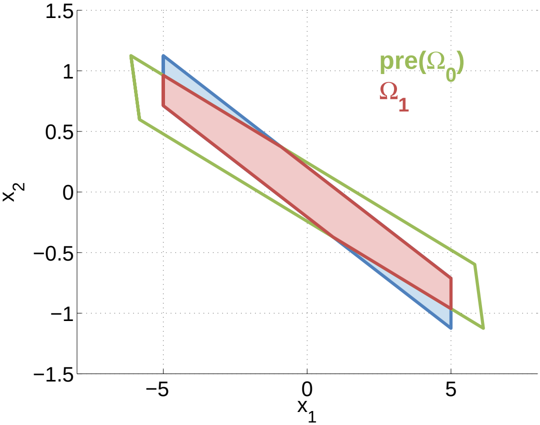
$\Omega_0 \leftarrow \mathcal{X}$
loop
  $\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$
  if $\Omega_{i+1} = \Omega_i$ **then**
    return $\mathcal{O}_\infty = \Omega_i$
  end if
end loop

System: $x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k)$, $\begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix}$

$-0.1 \leq u(k) \leq 0.1$

Where $u(k) = Kx(k)$, with K the optimal LQR controller for $Q = I$, $R = 90$.

Computing Invariant Sets



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

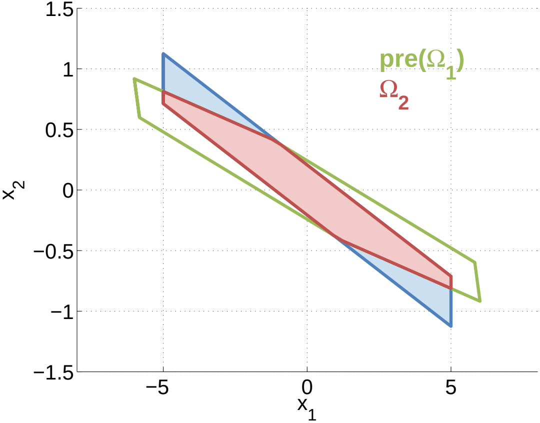
$\Omega_0 \leftarrow \mathcal{X}$
loop
  $\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$
  if $\Omega_{i+1} = \Omega_i$ **then**
    return $\mathcal{O}_\infty = \Omega_i$
  end if
end loop

System: $x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k)$, $\quad \begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix}$

$$-0.1 \leq u(k) \leq 0.1$$

Where $u(k) = Kx(k)$, with K the optimal LQR controller for $Q = I$, $R = 90$.

Computing Invariant Sets



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

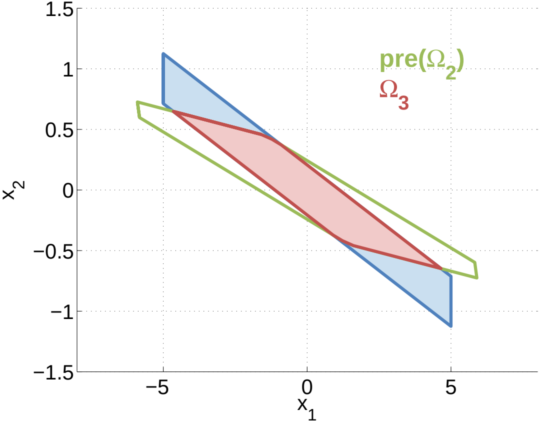
$\Omega_0 \leftarrow \mathcal{X}$
loop
  $\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$
  if $\Omega_{i+1} = \Omega_i$ **then**
    return $\mathcal{O}_\infty = \Omega_i$
  end if
end loop

System: $x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k)$, $\quad \begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix}$

$$-0.1 \leq u(k) \leq 0.1$$

Where $u(k) = Kx(k)$, with K the optimal LQR controller for $Q = I$, $R = 90$.

Computing Invariant Sets



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

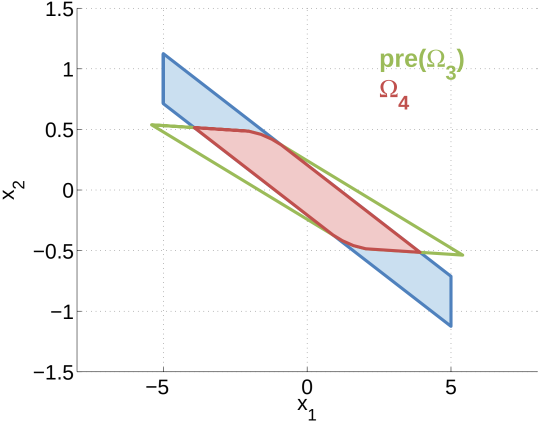
$\Omega_0 \leftarrow \mathcal{X}$
loop
  $\Omega_{i+1} \leftarrow \mathrm{pre}(\Omega_i) \cap \Omega_i$
  if $\Omega_{i+1} = \Omega_i$ **then**
    return $\mathcal{O}_\infty = \Omega_i$
  end if
end loop

System: $x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k)$, $\begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix}$

$-0.1 \leq u(k) \leq 0.1$

Where $u(k) = Kx(k)$, with K the optimal LQR controller for $Q = I$, $R = 90$.

Computing Invariant Sets



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

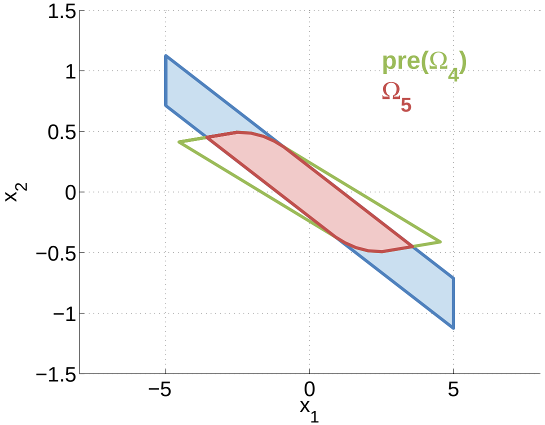
$\Omega_0 \leftarrow \mathcal{X}$
loop
  $\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$
  if $\Omega_{i+1} = \Omega_i$ **then**
    return $\mathcal{O}_\infty = \Omega_i$
  end if
end loop

System: $x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k)$, $\begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix}$

$$-0.1 \leq u(k) \leq 0.1$$

Where $u(k) = Kx(k)$, with K the optimal LQR controller for $Q = I$, $R = 90$.

Computing Invariant Sets



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

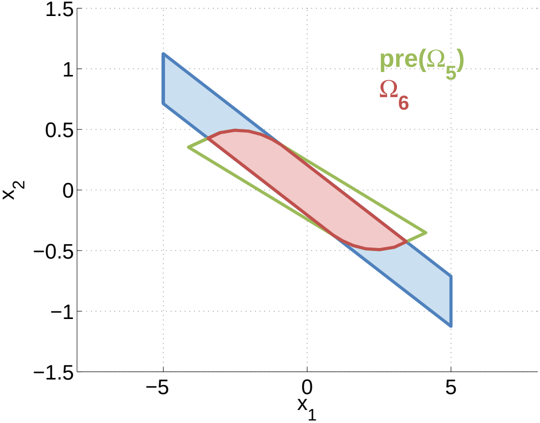
$\Omega_0 \leftarrow \mathcal{X}$
loop
  $\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$
  if $\Omega_{i+1} = \Omega_i$ **then**
    return $\mathcal{O}_\infty = \Omega_i$
  end if
end loop

System: $x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k)$, $\begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix}$

$$-0.1 \leq u(k) \leq 0.1$$

Where $u(k) = Kx(k)$, with K the optimal LQR controller for $Q = I$, $R = 90$.

Computing Invariant Sets



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

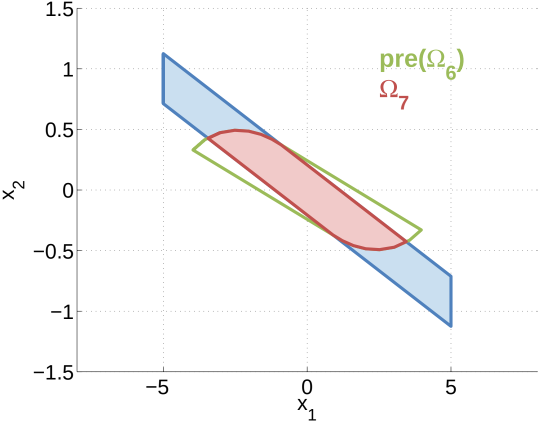
$\Omega_0 \leftarrow \mathcal{X}$
loop
  $\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$
  if $\Omega_{i+1} = \Omega_i$ **then**
    return $\mathcal{O}_\infty = \Omega_i$
  end if
end loop

System: $x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k)$, $\begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix}$

$-0.1 \leq u(k) \leq 0.1$

Where $u(k) = Kx(k)$, with K the optimal LQR controller for $Q = I$, $R = 90$.

Computing Invariant Sets



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

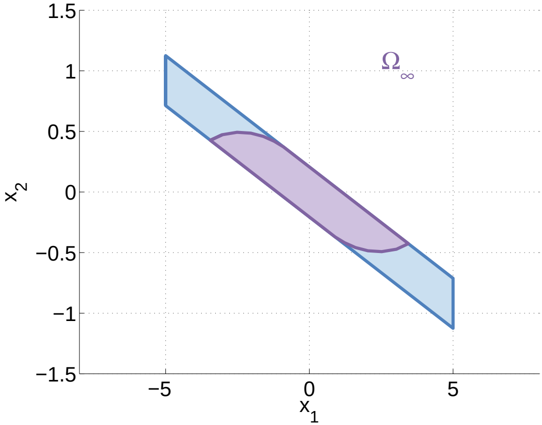
$\Omega_0 \leftarrow \mathcal{X}$
loop
  $\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$
  if $\Omega_{i+1} = \Omega_i$ **then**
    return $\mathcal{O}_\infty = \Omega_i$
  end if
end loop

System: $x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k)$, $\begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix}$

$$-0.1 \leq u(k) \leq 0.1$$

Where $u(k) = Kx(k)$, with K the optimal LQR controller for $Q = I$, $R = 90$.

Computing Invariant Sets



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

System: $x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k)$, $\begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix}$

$-0.1 \leq u(k) \leq 0.1$

Where $u(k) = Kx(k)$, with K the optimal LQR controller for $Q = I$, $R = 90$.

Outline

Controlled Invariance

Control Invariant Set

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

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

Defines the states for which there exists a **controller** that will satisfy constraints for **all time**.

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

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

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

Conceptual Calculation of Control Invariant Sets

Concept of a pre-set extends to systems with exogenous inputs

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

The same geometric condition holds for control invariant sets

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

As a result, the same conceptual algorithm can be used:

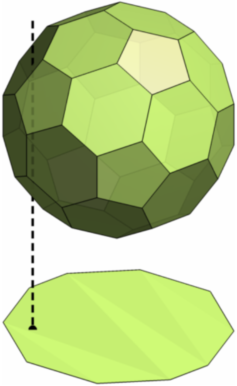
$\Omega_0 \leftarrow \mathcal{X}$
loop
  $\Omega_{i+1} \leftarrow \mathrm{pre}(\Omega_i) \cap \Omega_i$
  if $\Omega_{i+1} = \Omega_i$ **then**
    return $\mathcal{C}_\infty = \Omega_i$
  end if
end loop

However, it is now much harder to compute the pre-set!

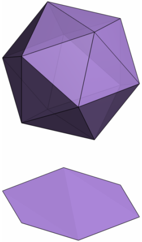
Pre-Set Computation: Controlled System

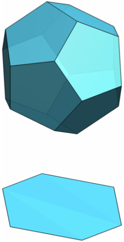
Consider the system $x(k+1) = Ax(k) + Bu(k)$ under the constraints $u(k) \in \mathcal{U} := \{u \mid Gu \leq g\}$ and the set $S := \{x \mid Fx \leq f\}$.

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

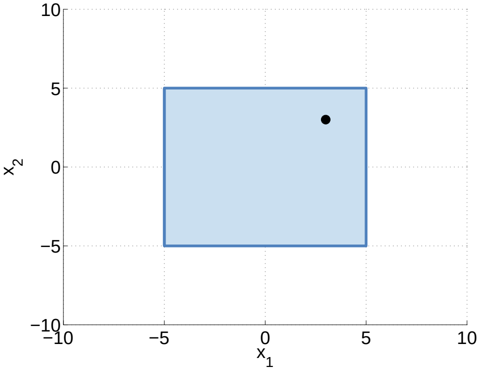


This is a **projection** operation.





Computing Control Invariant Sets



System:

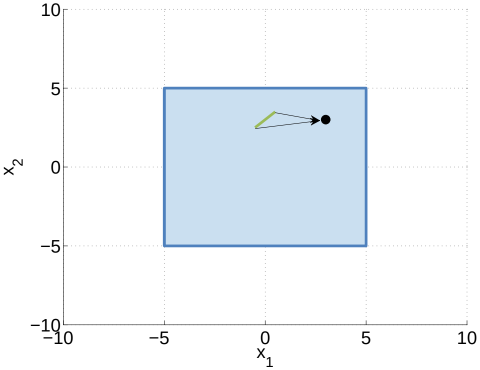
$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k)$$

Constraints:

$$\|x(k)\|_\infty \leq 5$$
$$\|u(k)\|_\infty \leq 1$$

An entire set of states can map into each point

Computing Control Invariant Sets

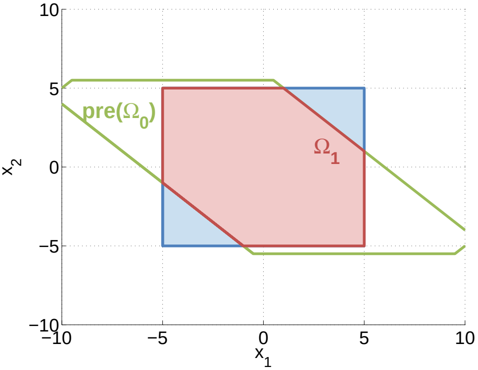


Algorithm:

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

An entire set of states can map into each point

Computing Control Invariant Sets



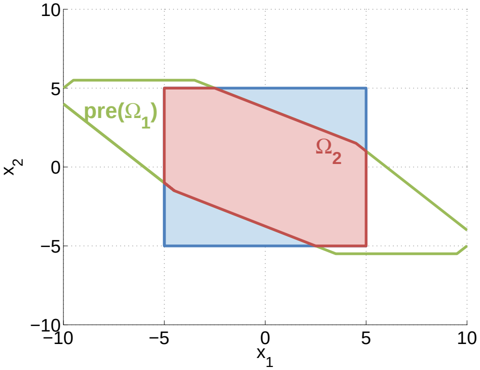
Algorithm:

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

An entire set of states can map into each point

The pre-set is a lot larger, but much more difficult to compute

Computing Control Invariant Sets



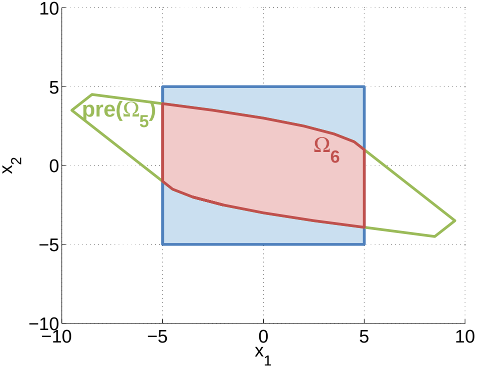
Algorithm:

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

An entire set of states can map into each point

The pre-set is a lot larger, but much more difficult to compute

Computing Control Invariant Sets



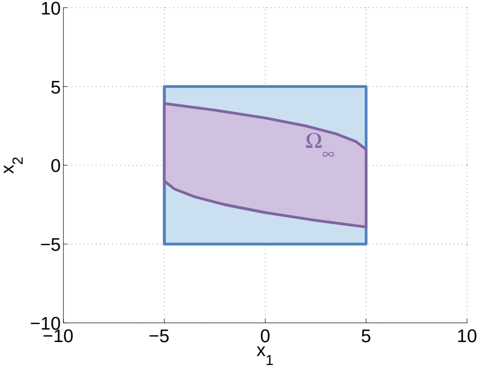
Algorithm:

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

An entire set of states can map into each point

The pre-set is a lot larger, but much more difficult to compute

Computing Control Invariant Sets



Algorithm:

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

An entire set of states can map into each point

The pre-set is a lot larger, but much more difficult to compute

The maximum control invariant set is the best any controller can do

Control Invariant Set $\Rightarrow$ Control Law

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

A control law $\kappa(x(k))$ will guarantee that the system $x(k+1) = g(x(k), \kappa(x(k)))$ will satisfy the constraints **for all time** if:

$$g(x, \kappa(x)) \in C \qquad \text{for all } x \in C$$

We can use this fact to **synthesize** a control law from a control invariant set by solving an optimization problem:

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

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

This doesn't ensure that the system will converge, but it will satisfy constraints.

Relation to MPC

- A control invariant set is a powerful object
- If one can compute one, it provides a direct method for synthesizing a control law that will satisfy constraints
- The maximal control invariant set is the best any controller can do!!!

So why don't we always compute them?

Relation to MPC

- A control invariant set is a powerful object
- If one can compute one, it provides a direct method for synthesizing a control law that will satisfy constraints
- The maximal control invariant set is the best any controller can do!!!

So why don't we always compute them?

We can't...

- Constrained linear systems : Often too complex
- (Constrained) nonlinear system : (Almost) always too complex

$\Rightarrow$ MPC **implicitly** describes a control invariant set such that it's easy to represent and compute.

Outline

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 $\sim 5 - 10$ states
    - Resulting MPC optimization will be a quadratic program

  - Ellipsoidal invariant set
    - Smaller than polyhedral (not the maximal invariant set)
    - Easy to compute for large dimensions
    - Fixed complexity
    - Resulting MPC optimization will be a quadratically constrained quadratic program

Summary: Control Invariant Sets

Special case: Linear system, polyhedral constraints.

- Very difficult to compute
- Very complex
- Very useful

Next week:
Turn an invariant set into a control invariant set with tractable computation
(This is what MPC does)

Outline

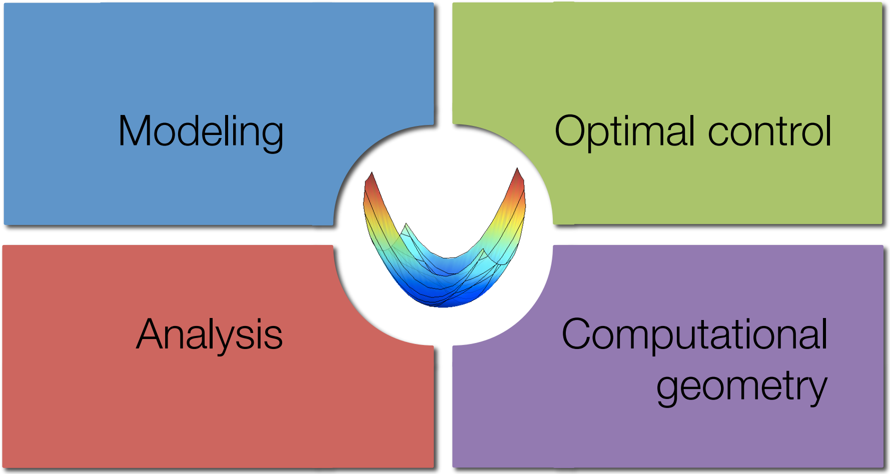
Outline

5. Practical Computation of Invariant Sets

    Polytopes

    Ellipsoids

Polytopes: MultiParametric Toolbox (MPT) (see supplementary material)



http://control.ee.ethz.ch/~mpt/

Outline

5. Practical Computation of Invariant Sets

    Polytopes

    Ellipsoids

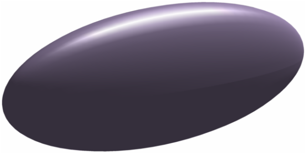
Ellipsoids

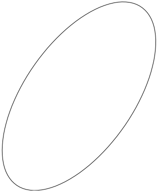
Ellipse

Let $P \succeq 0$ by a symmetric and positive-definite matrix in $\mathbb{R}^{n\times n}$ and $x_c \in \mathbb{R}^n$. The set

$$E := \left\{x \,\middle|\, (x - x_c)^T P(x - x_c) \leq 1\right\}$$

is an **ellipse**.





Ellipsoids are useful because the complexity of evaluating containment is always quadratic in the dimension, whereas polyhedra can be arbitrarily complex.

Invariant Sets from Lyapunov Functions

Lemma: Invariant set from Lyapunov function

If $V : \mathbb{R}^n \to \mathbb{R}$ 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 $\alpha \geq 0$.

"Proof": We have the basic properties:

- $V(x) \geq 0$ for all x
- $V(g(x)) - V(x) < 0$

The second property implies that once $V(x(k)) \leq \alpha$, $V(x(j))$ will be less than α for all $j \geq k \to$ Invariance □

We often want the largest invariant set contained in our constraints.

If V is a Lyapunov function for the system $x(k+1) = g(x(k))$, and our constraints are given by the set $\mathcal{X}$, then we maximize α such that

$$Y_\alpha := \{x \,|\, V(x) \leq \alpha\} \subseteq \mathcal{X}$$

Invariant Sets from Lyapunov Functions

Consider the system $x(k+1) = Ax(k)$, and assume $P \succ 0$ satisfies the condition

$$A^T PA - P \prec 0$$

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

Our goal is to find the largest α such that the invariant set Y_α is contained in the system constraints $\mathcal{X}$:

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

Equivalently, we want to solve the problem:

$$\begin{aligned} &\max_{\alpha}\ \alpha \\ &\text{subj. to } h_{Y_\alpha}(F_i) \leq f_i \text{ for all } i \in \{1, \ldots, n\} \end{aligned} \tag{1}$$

Maximum Ellipsoidal Invariant Sets

Support of an ellipse:

$$\begin{aligned} h_{Y_\alpha}(\gamma) = &\max_x \gamma^T x \\ &\text{subj. to } x^T P x \le \alpha \end{aligned} \tag{2}$$

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

$$\begin{aligned} h_{Y_\alpha}(\gamma) = &\max_y \gamma^T P^{-1/2} y \\ &\text{subj. to } y^T y \le \sqrt{\alpha}^2 \end{aligned} \tag{3}$$

which can be solved by inspection:

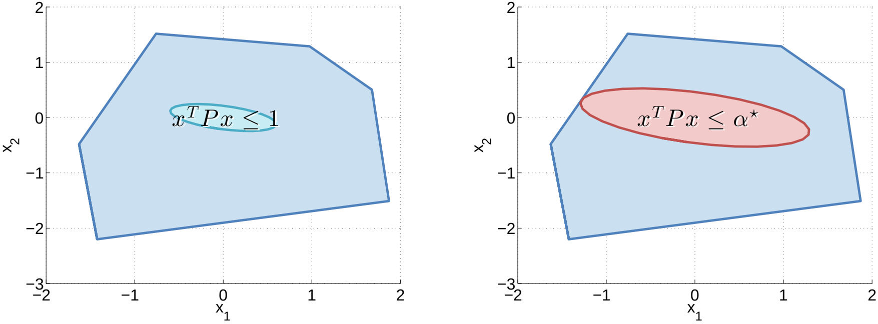
$$h_{Y_\alpha}(\gamma) = \gamma^T P^{-1/2} \frac{P^{-1/2}\gamma}{\|P^{-1/2}\gamma\|}\sqrt{\alpha} = \|P^{-1/2}\gamma\|\sqrt{\alpha}$$

Maximum Ellipsoidal Invariant Sets

Largest ellipse in a polytope is now a one-dimensional optimization problem:

$$\alpha^\star = \max_{\alpha} \ \alpha \quad \text{s.t.} \ \ \|P^{-1/2}F_i^T\|^2\alpha \le f_i^2 \text{ for all } i \in \{1,\ldots,n\}$$

$$= \min_{i\in\{1,\ldots,n\}} \frac{f_i^2}{F_i P^{-1} F_i^T}$$



It is possible to optimize over P, maximizing the volume of the ellipse, subject to stability and containment constraints (convex semi-definite program)

Largest Invariant Ellipsoid within Constraints (supplementary)

Convex optimization problem, which returns the largest invariant ellipoid within a polytope $\mathbb{X} = \{x \,|\, Fx \leq f\}$ (where we define $\tilde{P} := P^{-1}$)

$$\begin{aligned} \min_{\tilde{P}} \;& -\log\det \tilde{P} \\ \text{subj. to } & \begin{bmatrix} \tilde{P} & \tilde{P}A^T \\ A\tilde{P} & \tilde{P} \end{bmatrix} \succeq 0 \\ & F_i \tilde{P} F_i^T \leq f_i^2, \quad \text{for all } i = 1 \dots n \end{aligned}$$

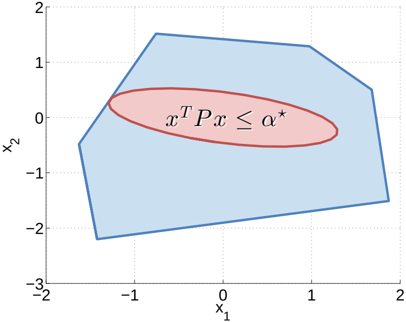
Notes:

- The volume of an ellipse is proportional to $\log\det P^{-1}$
- $\|P^{-1/2} F_i^T\|^2 = F_i P^{-1} F_i^T$
- $A^T P A - P \preceq 0 \Rightarrow \tilde{P} A^T P A \tilde{P} - \tilde{P} \preceq 0$ (pre- and postmultiply by $\tilde{P}$)
  $\Rightarrow$ Apply Schur complement

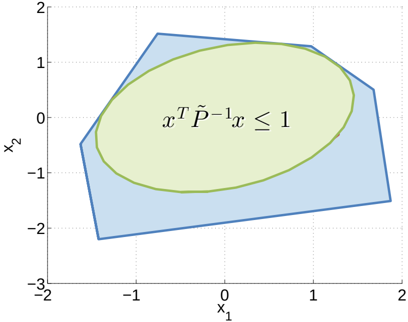
The largest volume ellipse centered at zero within the polytope $\mathbb{X}$ is then

$$\mathcal{E} = \left\{ x \,\middle|\, x^T \tilde{P}^{-1} x \leq 1 \right\} \subset \mathbb{X}$$

Example Revisited



Maximum volume ellipse using some matrix P.



Maximum volume ellipse resulting from any quadratic Lyapunov function.