University of Pennsylvania, ESE 6190

# Model Predictive Control

## Chapter 8: Invariance

Prof. Manfred Morari

Spring 2023

Coauthors: Prof. Colin Jones, EPFL
Prof. Francesco Borrelli, UC Berkeley

F. Borrelli, A. Bemporad, and M. Morari, Predictive Control for Linear and Hybrid Systems, Cambridge University Press, 2017. [Ch. 4, 10.1-10.2].

# Outline

# Outline

1. Objectives of Constrained Control

2. Invariance

3. Controlled Invariance

4. Polytopes and Polytopic Computation

5. Summary

6. Ellipsoids and Invariance

# Constrained Control

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

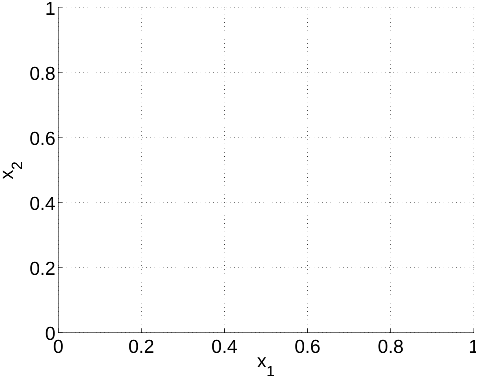
Design control law $u(k) = \kappa(x(k))$ such that the system:

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




Does linear control work?

System:

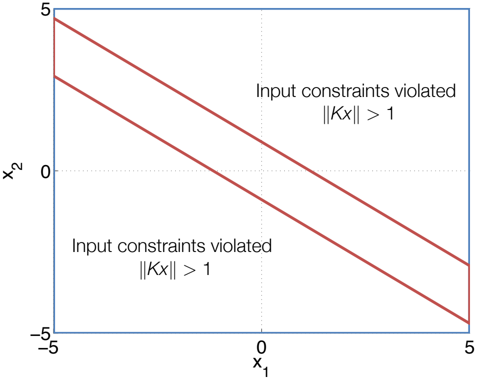
$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k)$$

Constraints:

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

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

# Limitations of Linear Controllers




System:

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

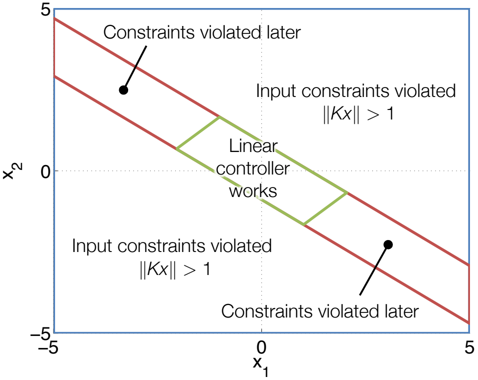
Constraints:

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

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

Does linear control work?

# 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:

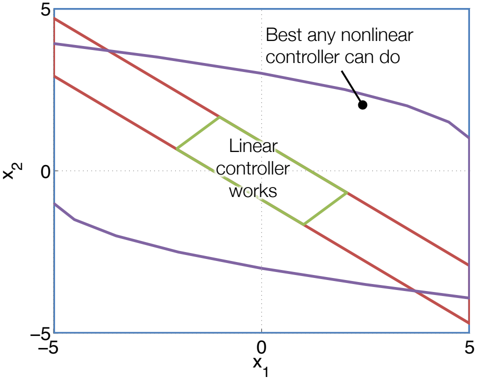
$$\mathcal{X} := \{x \mid \|x\|_\infty \leq 5\}$$
$$\mathcal{U} := \{u \mid \|u\|_\infty \leq 1\}$$

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**

# Lecture Take Homes

We consider the following two types of systems:

- Autonomous system $x(k+1) = f_a(x(k))$,
- Controlled system $x(k+1) = f(x(k), u(k))$.

Both systems are subject to state and input constraints

$$x(k) \in \mathcal{X}, u(k) \in \mathcal{U}, \forall k \leq 0.$$

Two concepts this lecture:

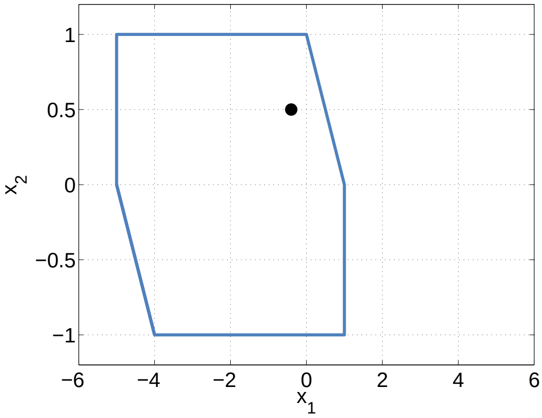
- Invariance
  - Region in which an **autonomous** system will satisfy the constraints **for all time**
- Controlled invariance
  - Region for which there **exists a controller** so that the system satisfies the constraints **for all time**

And some practical computation:

- How to compute these for some important problems

# Invariance: Which states are "good"?

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




System:

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

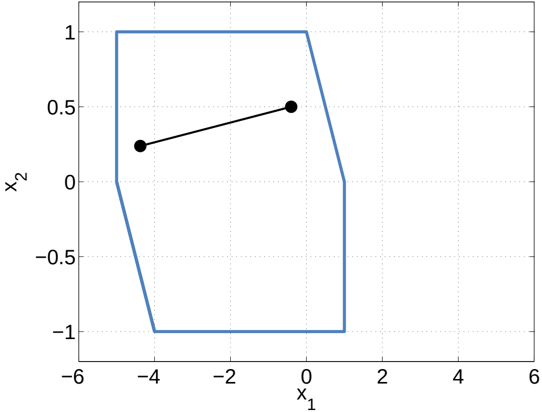
where $\omega = 10$, $\zeta = 0.01$, sampled at 10Hz.

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?




System:

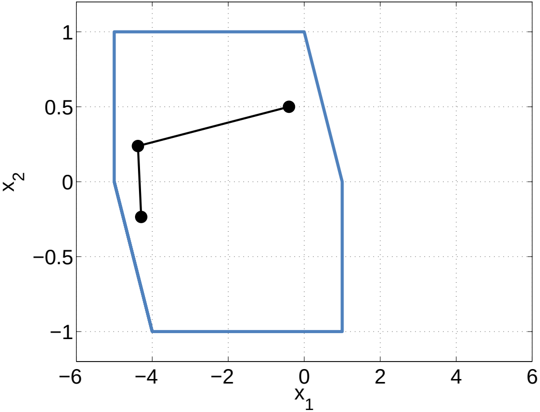
$$\dot{x}(t) = \begin{bmatrix} -2\zeta\omega & -\omega^2 \\ 1 & 0 \end{bmatrix} x(t)$$

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...




System:

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

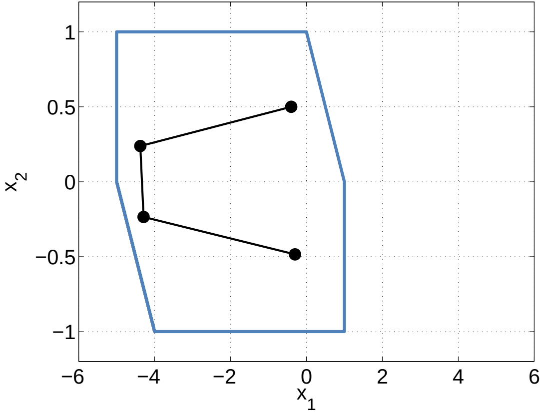
where $\omega = 10$, $\zeta = 0.01$, sampled at 10Hz.

Constraints:

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

# Invariance: Which states are "good"?

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




System:

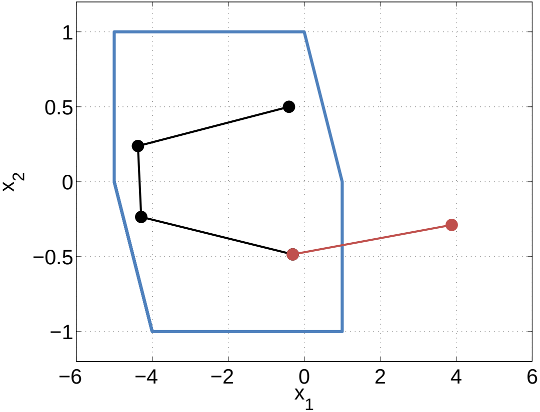
$$\dot{x}(t) = \begin{bmatrix} -2\zeta\omega & -\omega^2 \\ 1 & 0 \end{bmatrix} x(t)$$

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:

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

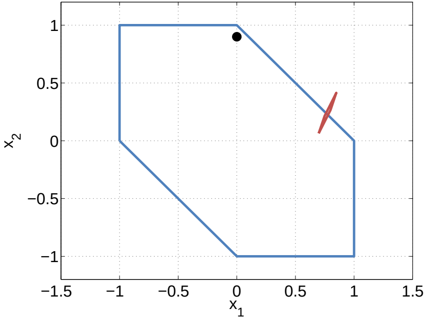
where $\omega = 10$, $\zeta = 0.01$, sampled at 10Hz.
Constraints:

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

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?




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

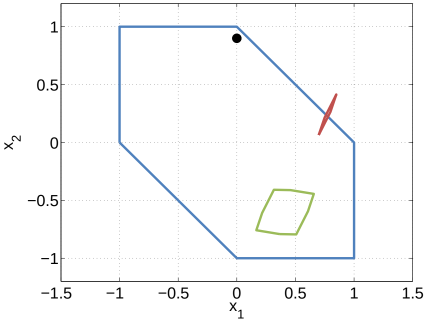
Constraints:

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

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?




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

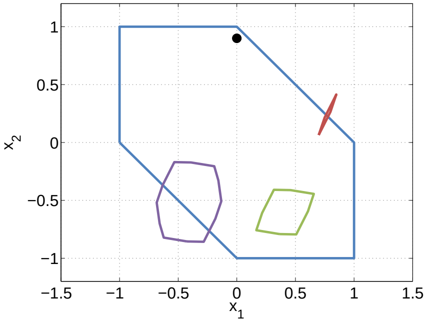
Constraints:

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

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?




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

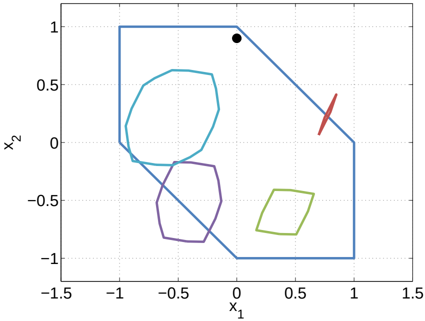
Constraints:

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

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?




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

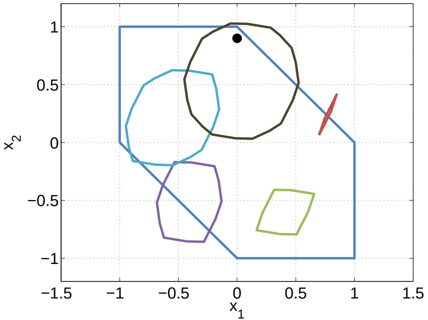
Constraints:

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

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?




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

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 $x(k+1) = f(x(k))$, or **closed-loop** system $x(k+1) = f(x(k), \kappa(x(k)))$ for a **given** controller $\kappa$.

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

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 $\mathcal{O}_\infty$.

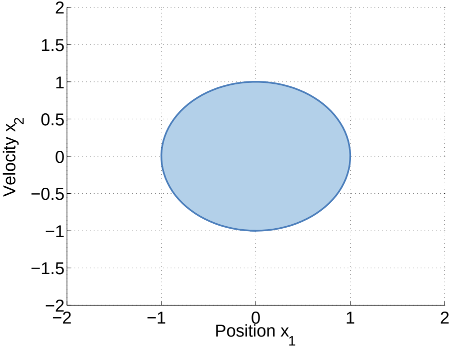
# Pre-Sets

**Pre Set**

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

$$\text{pre}(S) := \{x \,|\, f(x) \in S\}$$

# Pre-Set Example : Pendulum




Pendulum:

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

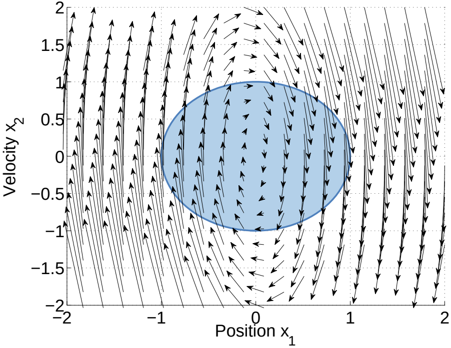
(Discretized with forward Euler at 1Hz)

Target set:

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

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)

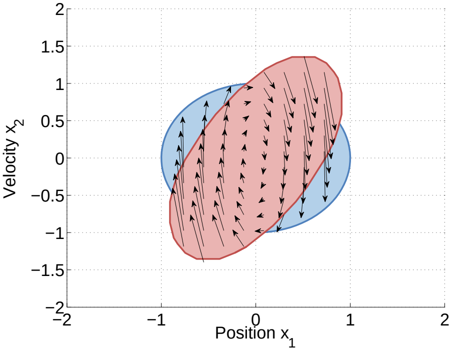
Target set:

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

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.

# 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 $\mathsf{pre}(\mathcal{O})$, $f(\bar{x}) \notin \mathcal{O}$ 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 $f(\bar{x}) \notin \mathcal{O}$. 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

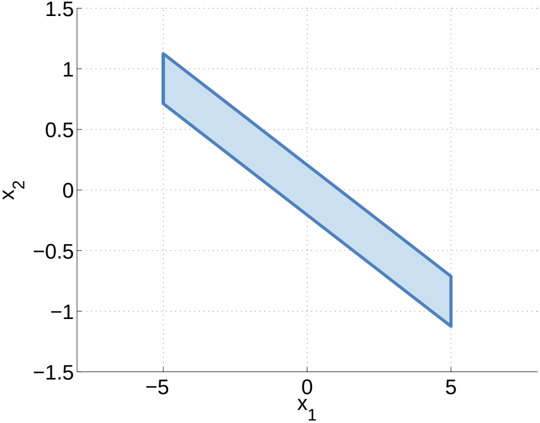
**Conceptual Algorithm to Compute Invariant Set**

**Input:** $f$, $\mathcal{X}$
**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 $\Omega_i$ is the maximal positive invariant set $\mathcal{O}_\infty$ for $x(k+1) = f(x(k))$.

# Computing Invariant Sets




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

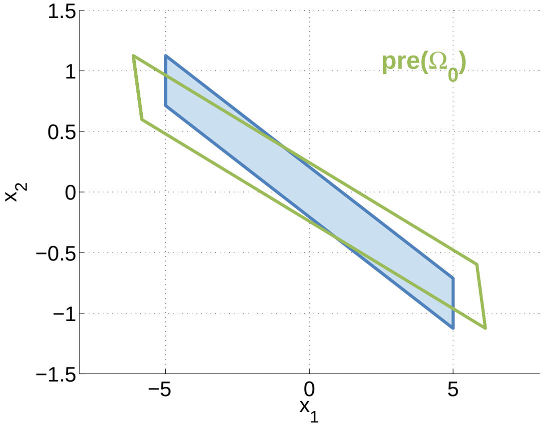
$\Omega_0 \leftarrow \mathcal{X}$
**loop**
  $\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$
  **if** $\Omega_{i+1} = \Omega_i$ **then**
    **return** $\mathcal{O}_\infty = \Omega_i$
  **end if**
**end loop**

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:** $f$, $\mathcal{X}$
**Output:** $\mathcal{O}_\infty$

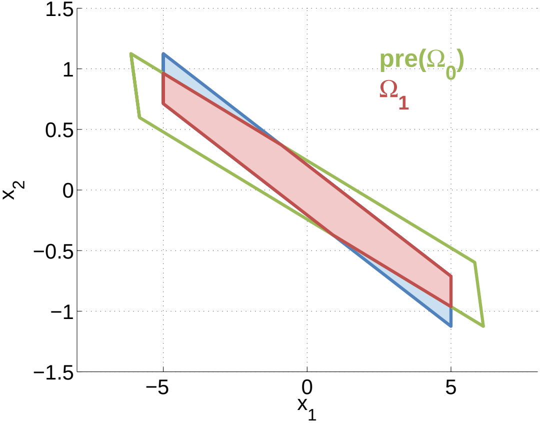
$\Omega_0 \leftarrow \mathcal{X}$
**loop**
  $\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$
  **if** $\Omega_{i+1} = \Omega_i$ **then**
    **return** $\mathcal{O}_\infty = \Omega_i$
  **end if**
**end loop**

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:** $f$, $\mathcal{X}$
**Output:** $\mathcal{O}_\infty$

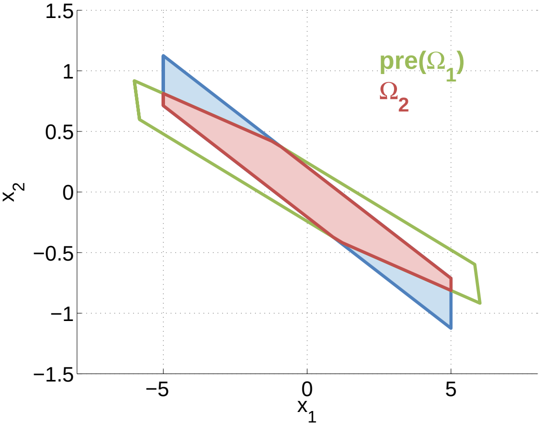
$\Omega_0 \leftarrow \mathcal{X}$
**loop**
  $\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$
  **if** $\Omega_{i+1} = \Omega_i$ **then**
    **return** $\mathcal{O}_\infty = \Omega_i$
  **end if**
**end loop**

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:** $f$, $\mathcal{X}$
**Output:** $\mathcal{O}_\infty$

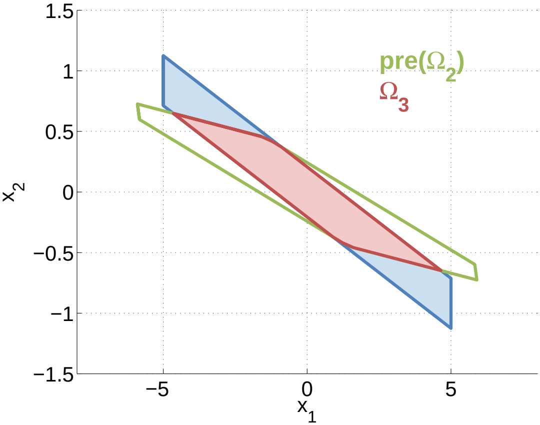
$\Omega_0 \leftarrow \mathcal{X}$
**loop**
  $\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$
  **if** $\Omega_{i+1} = \Omega_i$ **then**
    **return** $\mathcal{O}_\infty = \Omega_i$
  **end if**
**end loop**

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:** $f$, $\mathcal{X}$
**Output:** $\mathcal{O}_\infty$

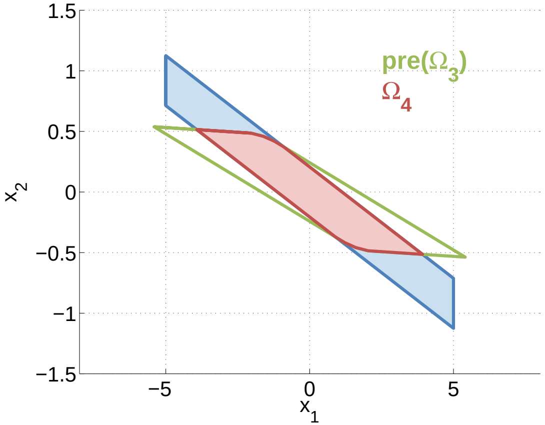
$\Omega_0 \leftarrow \mathcal{X}$
**loop**
  $\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$
  **if** $\Omega_{i+1} = \Omega_i$ **then**
    **return** $\mathcal{O}_\infty = \Omega_i$
  **end if**
**end loop**

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

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

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

# Computing Invariant Sets




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

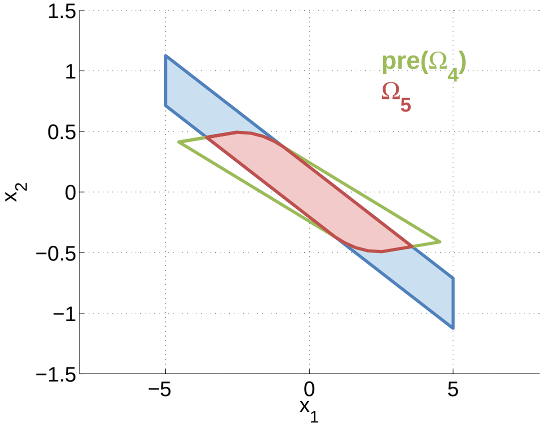
$\Omega_0 \leftarrow \mathcal{X}$
**loop**
  $\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$
  **if** $\Omega_{i+1} = \Omega_i$ **then**
    **return** $\mathcal{O}_\infty = \Omega_i$
  **end if**
**end loop**

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:** $f$, $\mathcal{X}$
**Output:** $\mathcal{O}_\infty$

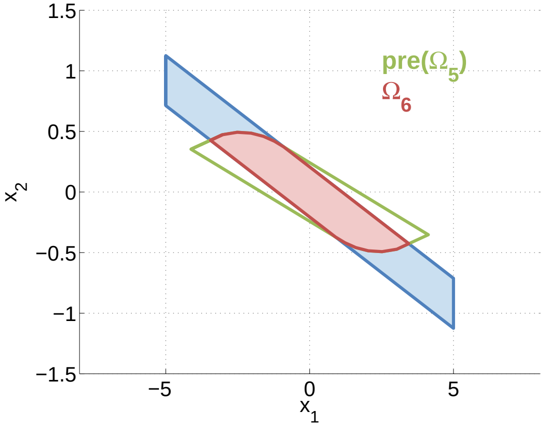
$\Omega_0 \leftarrow \mathcal{X}$
**loop**
  $\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$
  **if** $\Omega_{i+1} = \Omega_i$ **then**
    **return** $\mathcal{O}_\infty = \Omega_i$
  **end if**
**end loop**

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:** $f$, $\mathcal{X}$
**Output:** $\mathcal{O}_\infty$

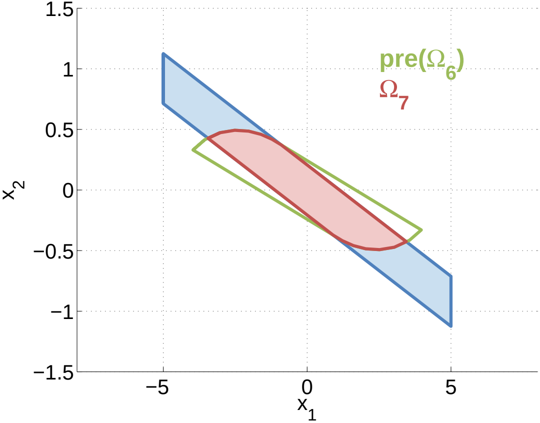
$\Omega_0 \leftarrow \mathcal{X}$
**loop**
  $\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$
  **if** $\Omega_{i+1} = \Omega_i$ **then**
    **return** $\mathcal{O}_\infty = \Omega_i$
  **end if**
**end loop**

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:** $f$, $\mathcal{X}$
**Output:** $\mathcal{O}_\infty$

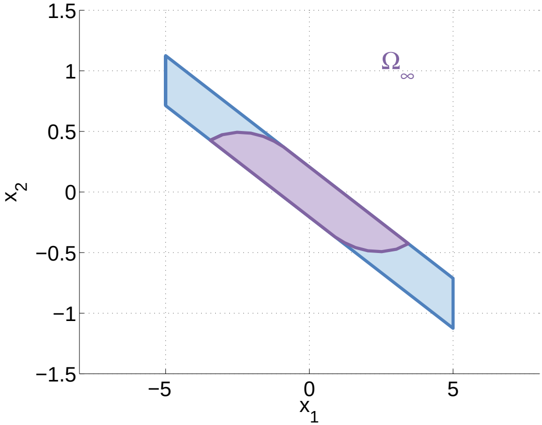
$\Omega_0 \leftarrow \mathcal{X}$
**loop**
  $\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$
  **if** $\Omega_{i+1} = \Omega_i$ **then**
    **return** $\mathcal{O}_\infty = \Omega_i$
  **end if**
**end loop**

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:** $f$, $\mathcal{X}$
**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

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

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 $x(k+1) = f(x(k), u(k))$ subject to the constraints $(x(k), u(k)) \in \mathcal{X} \times \mathcal{U}$ 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

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

The same geometric condition holds for control invariant sets

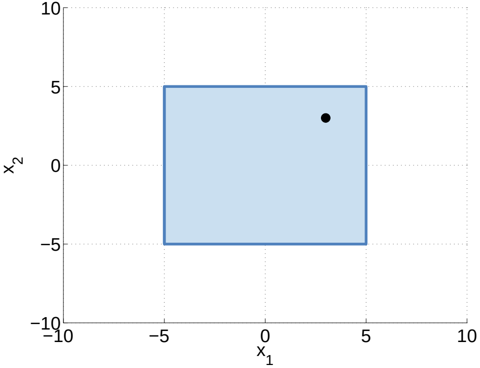
A set $\mathcal{C}$ is a control invariant set if and only if $\mathcal{C} \subseteq \text{pre}(\mathcal{C})$

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

$\Omega_0 \leftarrow \mathcal{X}$
**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**

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

# Computing Control Invariant Sets




System:

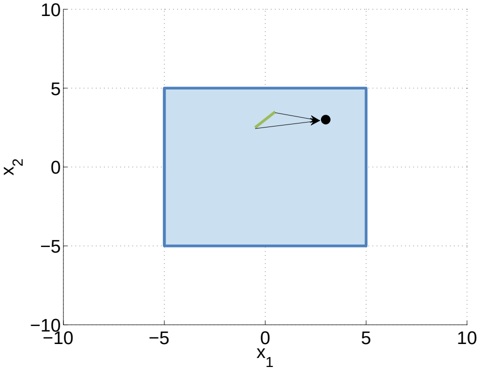
$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k)$$

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

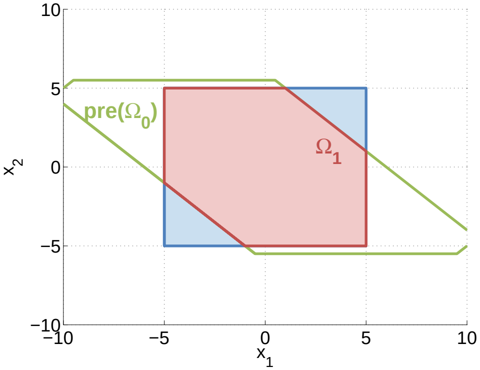



Algorithm:

$\Omega_0 \leftarrow \mathcal{X}$
**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



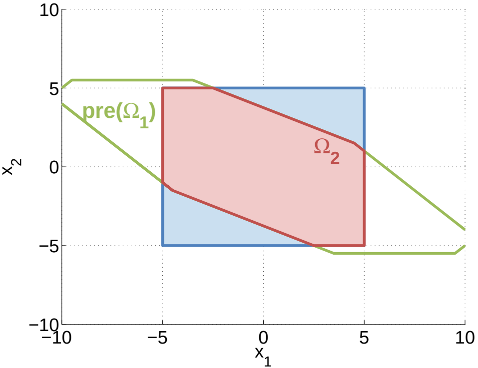

Algorithm:

$\Omega_0 \leftarrow \mathcal{X}$
**loop**
  $\Omega_{i+1} \leftarrow \mathrm{pre}(\Omega_i) \cap \Omega_i$
  **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

# Computing Control Invariant Sets



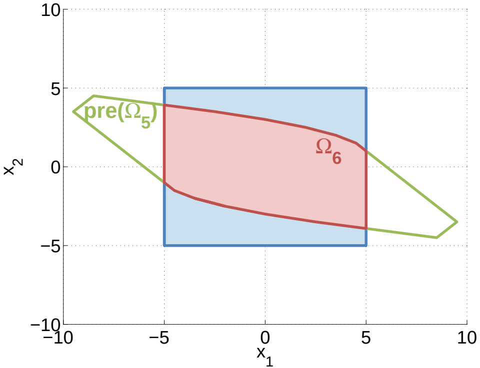

Algorithm:

$\Omega_0 \leftarrow \mathcal{X}$
**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



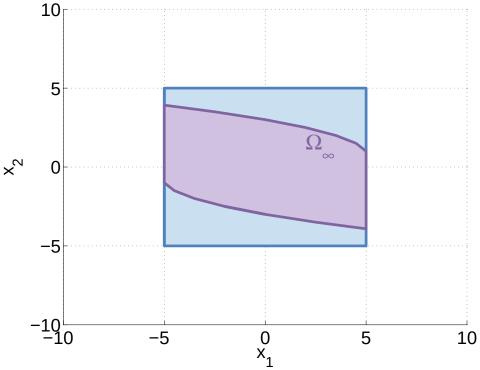

Algorithm:

$\Omega_0 \leftarrow \mathcal{X}$
**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

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 $x(k+1) = f(x(k), u(k))$.

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

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

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

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

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

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

What is MPC?

- A method of **implicitly** describing a control invariant set such that it's easy to represent and compute!

# Outline

# 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 $\Omega_i$ (Polytopes)
- Intersection
- Pre-set computation
- Equality test (bi-directional subset)

This part of the lecture will go through these operations for **polytopes**.

# Polyhedra

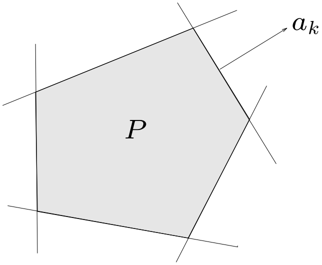
**Polyhedron**

A **polyhedron** is the intersection of a finite number of halfspaces.

$$P := \left\{x \,\middle|\, a_i^T x \leq b_i,\ i = 1, \ldots, n\right\}$$

A **polytope** is a bounded polyhedron.

Often written as $P := \{x \,|\, Ax \leq b\}$, for matrix $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$, where the inequality is understood row-wise.




# Convex hull

**Convex hull**

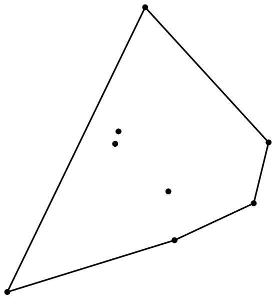
For any subset $S$ of $\mathbb{R}^d$, the convex hull $\mathbf{co}(S)$ of $S$ is the intersection of all convex sets containing $S$. Since the intersection of two convex sets is convex, it is the smallest convex set containing $S$.

**Proposition: Convex hull**

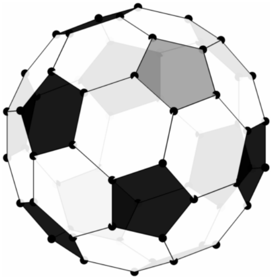
The convex hull of a set $S \subseteq \mathbb{R}^d$ is
Given a set of points $\{v_1, \ldots, v_k\}$ in $\mathbb{R}^d$, their **convex hull** is

$$\mathbf{co}(\{v_1, \ldots, v_k\}) := \left\{ x \,\middle|\, x = \sum_i \lambda_i v_i,\ \lambda_i \geq 0,\ \sum_i \lambda_i = 1\ \forall i = 1, \ldots, k \right\}$$

# Examples of convex hulls



2D convex hull



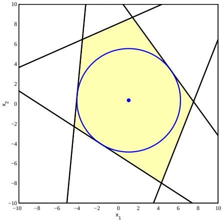
3D convex hull

# Basic Operations on Polytopes

- **Polytope reduction** is the computation of the minimal representation of a polytope. A polytope $\mathcal{P} \subset \mathbb{R}^n$, $\mathcal{P} = \{x \in \mathbb{R}^n \ : \ Ax \leq b\}$ is in a **minimal representation** if the removal of any row in $Ax \leq b$ would change it (i.e., if there are no redundant constraints).

  In Matlab: P = Polytope(A,b,normal,minrep), minrep=1
- The **Chebychev Ball** of a polytope $\mathcal{P}$ corresponds to the largest radius ball $\mathcal{B}(x_c, R)$ with center $x_c$, such that $\mathcal{B}(x_c, R) \subset \mathcal{P}$.

  In Matlab: P.chebyCenter.x, P.chebyCenter.r

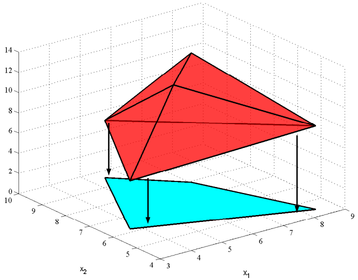


# Basic Operations on Polytopes

- **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

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

In Matlab: Q = projection(P,dim)

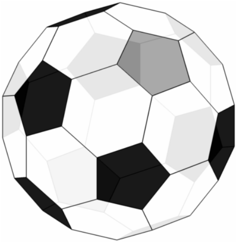


# Polytopes: Representations

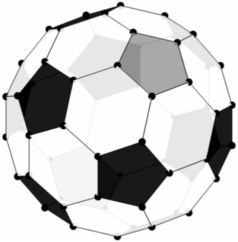
**Theorem: Minkowski-Weyl Theorem**

For $P \subseteq \mathbb{R}^d$, the following statements are equivalent:

- $P$ is a polytope, i.e., $P$ is bounded and there exist $A \in \mathbb{R}^{m \times d}$ and $b \in \mathbb{R}^m$ such that $P = \{x \mid Ax \leq b\}$
- $P$ is finitely generated, i.e., there exist a finite set of vectors $\{v_i\}$ such that $P = \text{co}\left(\{v_1, \ldots, v_s\}\right)$



$=$



$P = \{x \mid Ax \leq b\}$ $\qquad$ $P = \text{co}\left(\{v_1, \ldots, v_s\}\right)$

# Most Common Polytopic Constraints

$$x(k+1) = Ax(k) + Bu(k) \qquad\qquad y(k) = Cx(k)$$

Suppose we have the following input and output constraints:

$$u_{low} \leq u(k) \leq u_{high}$$
$$y_{low} \leq y(k) \leq y_{high}$$

Recalling that $y(k) = Cx(k)$, this is equivalent to:

$$\begin{bmatrix} 0 & -I \\ 0 & I \\ -C & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \leq \begin{bmatrix} -u_{low} \\ u_{high} \\ -y_{low} \\ y_{high} \end{bmatrix}$$

# Polytopes in MPC

## Input saturation

$$u_{lb} \leq u(k) \leq u^{ub}$$
$$\Downarrow$$
$$\begin{bmatrix} 1 \\ -1 \end{bmatrix} u(k) \leq \begin{bmatrix} u^{ub} \\ -u_{lb} \end{bmatrix}$$

## Magnitude constraints

$$\|Cx(k)\|_\infty \leq \alpha$$
$$\Downarrow$$
$$\begin{bmatrix} C \\ -C \end{bmatrix} x(k) \leq \mathbf{1}\alpha$$

## Rate constraints

$$\|x(k) - x(k+1)\|_\infty \leq \alpha$$
$$\Downarrow$$
$$\begin{bmatrix} I & -I \\ -I & I \end{bmatrix} \begin{bmatrix} x(k) \\ x(k+1) \end{bmatrix} \leq \mathbf{1}\alpha$$

## Integral constraints

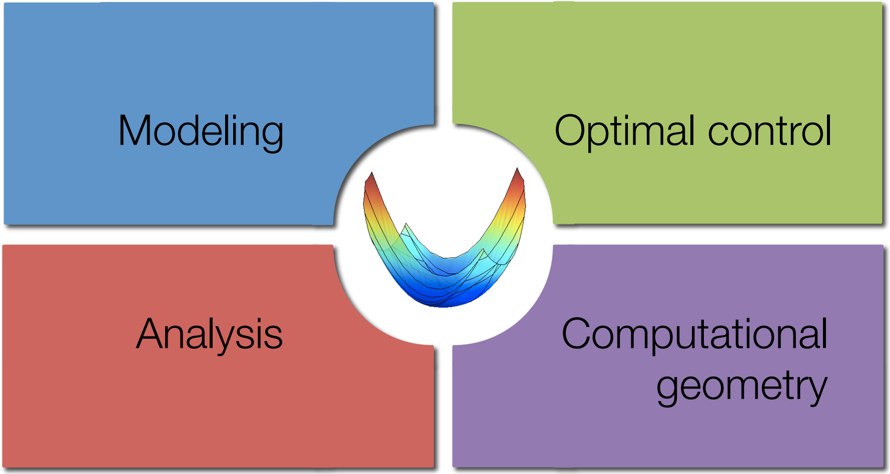
$$\|x(k)\|_1 \leq \alpha$$
$$\Downarrow$$
$$x(k) \in \mathrm{co}\left(e_i\alpha, -e_i\alpha\right)$$

Polytopes in MPC are commonly described as a set of **inequalities**.
This is a standing assumption in the following.

---

$\mathbf{1}$ is a vector of all ones
$e_i$ is the $i^{th}$ elementary vector $(0, \ldots, 0, 1, 0, \ldots, 0)$, with the 1 in the $i^{th}$ position

# MultiParametric Toolbox (MPT)



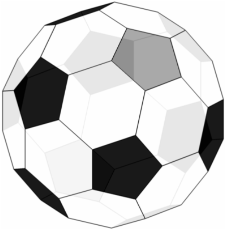

http://control.ee.ethz.ch/~mpt/

# Creating polytopes in MPT

## Polytope in inequality form

Define $P = \{x \mid Fx \leq f\}$:

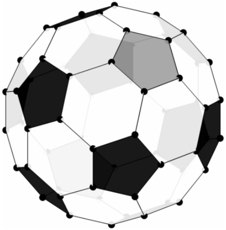
```
P = Polyhedron(F, f);
P.plot
```



## Polytope in vertex form

Define $P = \mathrm{co}\,(v_i)$

```
P = Polyhedron([v0 v1 .. vn]');
P.plot
```



Obtaining the vertices / inequalities:

```
F = P.A; f = P.b; % Inequalities
V = P.V;     % Vertices
```

# Computing Invariant Sets

**Conceptual Algorithm to Compute Invariant Set**

$\Omega_0 \leftarrow \mathcal{X}$
**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**

Requirements:

- Represent set $\Omega_i$ (Polytopes)
- **Intersection**
- Pre-set computation
- Equality test (bi-directional subset)

This part of the lecture will go through these operations for **polytopes**.

# Intersection of Polytopes

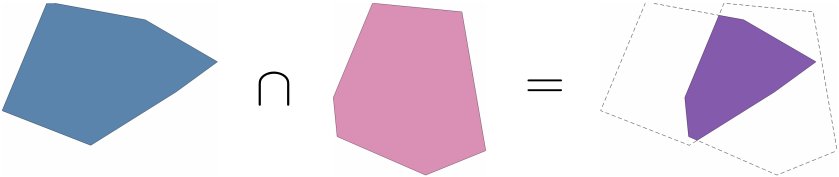
**Intersection**

The intersection $I \subseteq \mathbb{R}^n$ of sets $S \subseteq \mathbb{R}^n$ and $T \subseteq \mathbb{R}^n$ is

$$I = S \cap T := \{x \,|\, x \in S \text{ and } x \in T\}$$

Intersection of polytopes in **inequality form** is easy:

$$S := \{x \,|\, Cx \le c\} \qquad\qquad S \cap T = \left\{ x \,\middle|\, \begin{bmatrix} C \\ D \end{bmatrix} x \le \begin{bmatrix} c \\ d \end{bmatrix} \right\}$$
$$T := \{x \,|\, Dx \le d\}$$



Intersection of polytopes in vertex form is difficult (exponential complexity)

# 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 $\Omega_i$ (Polytopes)
- Intersection
- **Pre-set computation**
- Equality test (bi-directional subset)

This part of the lecture will go through these operations for **polytopes**.

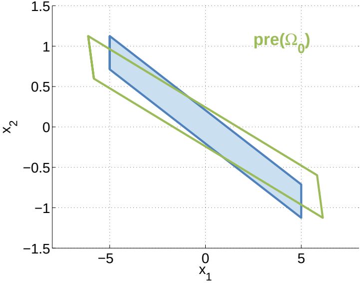
# Pre-Set Computation: Autonomous System

**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:

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

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




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

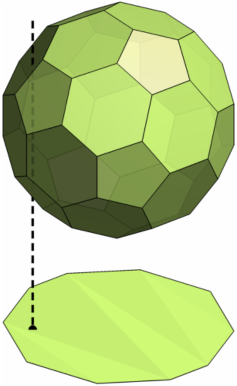
$$\begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x(k) \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix}, \quad \|u(k)\|_\infty \leq 0.1$$

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

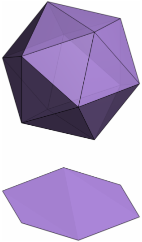
# Pre-Set Computation: Controlled System

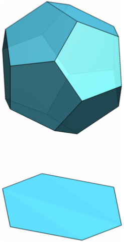
Consider the system $x(k+1) = Ax(k) + Bu(k)$ under the constraints $u(k) \in \mathcal{U} := \{u \,|\, Gu \le g\}$ and the set $S := \{x \,|\, Fx \le f\}$.

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



This is a **projection** operation.





# Polytopic Projection

**Polytopic Projection**

Given a polytope $P = \left\{(x, y) \in \mathbb{R}^n \times \mathbb{R}^d \,\middle|\, Cx + Dy \le b\right\}$, find a matrix $E$ and vector $e$, such that the polytope

$$P_\pi = \{x \,|\, Ex \le e\} = \{x \,|\, \exists y,\ (x, y) \in P\}$$

Computing projections in inequality form is computationally complex.

If $C \in \mathbb{R}^{m\times n}$, and $E \in \mathbb{R}^{q\times n}$, then:

- $q$ can be an exponential function of $m$ (worst case)
- Standard algorithms take time and space doubly exponential in $m$ and $q$
- Best algorithm to date is polynomial time in $m$ and linear in $q$[1]
  (Colin Jones' PhD)

We won't go through this algorithm here, but the lecture on explicit MPC will give you an idea of how it works.

[1] Requires that $P$ has a special structure, which is a form of general position

# Polytopic Projections in MPT

Several projection algorithms are implemented in MPT.

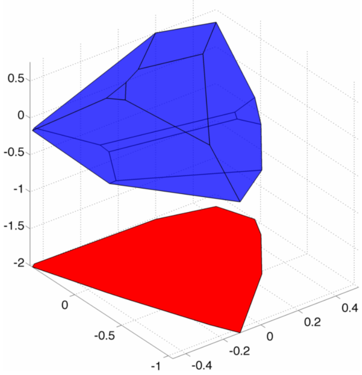
The best is a function of the dimension, complexity and numerical sensitivity of the polytope being projected. For the most part, the defaults work well.

```
% Random polytope in R3
P = Polyhedron(randn(20,3), ones(20,1))

% Dimensions to project onto
dims = 1:2;

% Compute the projection
p = P.projection(dims);

% Plot the result
plot(P+[0;0;1],'color','b')
hold on;
p.plot('color', r');
```



# Computing Invariant Sets

| Conceptual Algorithm to Compute Invariant Set |
| --- |
| $\Omega_0 \leftarrow \mathcal{X}$<br>**loop**<br>    $\Omega_{i+1} \leftarrow \mathrm{pre}(\Omega_i) \cap \Omega_i$<br>    **if** $\Omega_{i+1} = \Omega_i$ **then**<br>        **return** $\mathcal{O}_\infty = \Omega_i$<br>    **end if**<br>**end loop** |

Requirements:

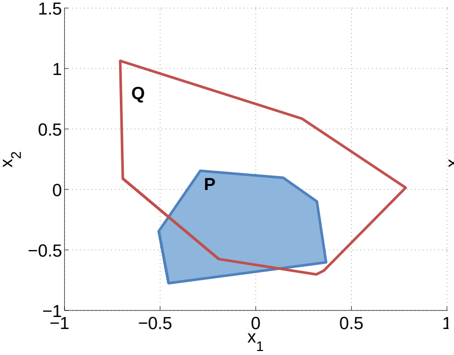
- Represent set $\Omega_i$ (Polytopes)
- Intersection
- Pre-set computation
- **Equality test (bi-directional subset)**

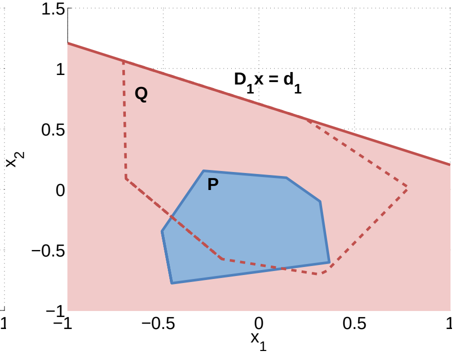
This part of the lecture will go through these operations for **polytopes**.

# Subset Test

**Problem**: Is $P := \{x \,|\, Cx \le c\}$ contained in $Q := \{x \,|\, Dx \le d\}$?

The statement is true if $P \subset \{x \,|\, D_i x \le d_i\}$ for each row $D_i$ of $D$.

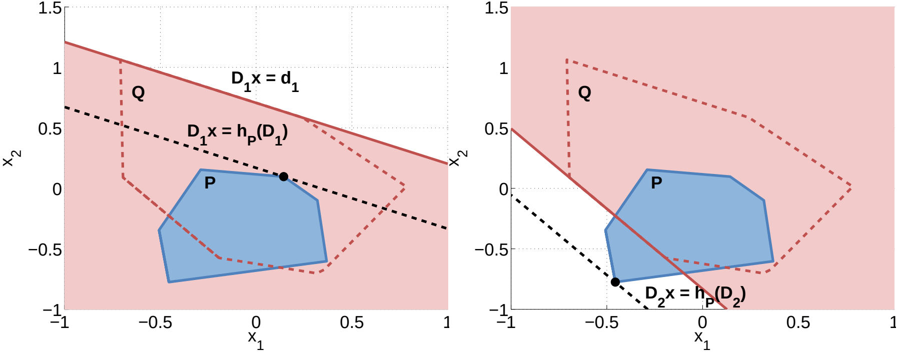


# Subset Test

Define the **support function** of the set $P$:

$$h_P(D_i) := \max_x \; D_i x \qquad \text{subj. to } Cx \le c \tag{1}$$



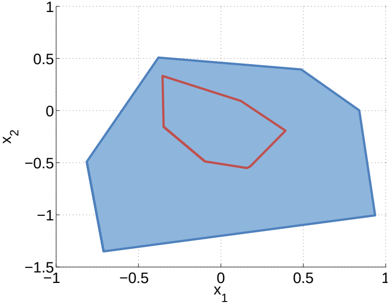

$h_P(D_1) \le d_1$ ✔ $\qquad h_P(D_2) > d_2$ ✖

# Subset Test in MPT

```
P = Polyhedron(randn(10,2), ones(10,1)); % Define two polytopes
Q = Polyhedron(randn(10,2), 0.5*ones(10,1));

if      P <= Q, fprintf('P is a subset of Q\n');
elseif Q <= P, fprintf('Q is a subset of P\n');
end

if P == Q, fprintf('P is equal to Q\n'); end
```




```
Q is a subset of P
```

# Computing Invariant Sets

| Conceptual Algorithm to Compute Invariant Set |
| --- |
| $\Omega_0 \leftarrow \mathcal{X}$<br>**loop**<br>    $\Omega_{i+1} \leftarrow \mathrm{pre}(\Omega_i) \cap \Omega_i$<br>    **if** $\Omega_{i+1} = \Omega_i$ **then**<br>        **return** $\mathcal{O}_\infty = \Omega_i$<br>    **end if**<br>**end loop** |

Requirements:

- Represent set $\Omega_i$ (Polytopes)
- Intersection
- Pre-set computation
- Equality test (bi-directional subset)

This part of the lecture will go through these operations for **polytopes**.

# Outline

# Summary: Invariant Sets

Linear Systems / 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

(See extra notes at end of lecture to learn more about ellipsoidal invariant sets)

# Summary: Control Invariant Sets

Linear system, polyhedral constraints.

- Very difficult to compute
- Very complex
- Very useful

# Summary: Control Invariant Sets

Linear system, polyhedral constraints.

- Very difficult to compute
- Very complex
- Very useful

**MPC:**
Turn an invariant set into a control invariant set with tractable computation

# Outline

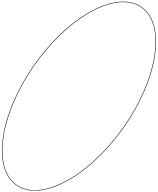
# Ellipsoids

**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 $x(k+1) = f(x(k))$, then $$Y := \{x \,\vert\, V(x) \leq \alpha\}$$ is an invariant set for all $\alpha \geq 0$. |

We have the basic properties:

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

The second property implies that once $V(x(i)) \leq \alpha$, $V(x(j))$ will be less than $\alpha$ for all $j \geq i$. $\square$

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

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

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

# Invariant Sets from Lyapunov Functions

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

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

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

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

$$Y_\alpha := \left\{ x \,\middle|\, x^T P x \le \alpha^2 \right\} \subset \mathcal{X} := \{ x \mid Fx \le f \}$$

Equivalently, we want to solve the problem:

$$\begin{aligned} &\max_{\alpha} \ \alpha \\ &\text{subj. to } h_{Y_\alpha}(F_i) \le f_i \text{ for all } i \in \{1, \dots, n\} \end{aligned} \tag{2}$$

# Maximum Ellipsoidal Invariant Sets

Support of an ellipse:

$$\begin{aligned} h_{Y_\alpha}(\gamma) = \max_x \ & \gamma^T x \\ \text{subj. to } & x^T P x \le \alpha^2 \end{aligned} \tag{3}$$

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

$$\begin{aligned} h_{Y_\alpha}(\gamma) = \max_y \ & \gamma^T P^{-1/2} y \\ \text{subj. to } & y^T y \le \alpha^2 \end{aligned} \tag{4}$$

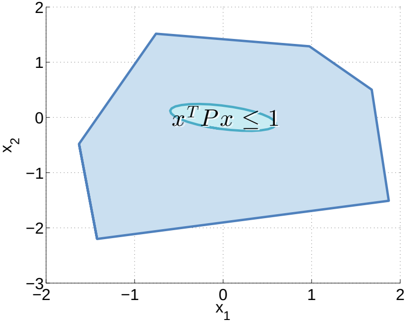
which can be solved by inspection:

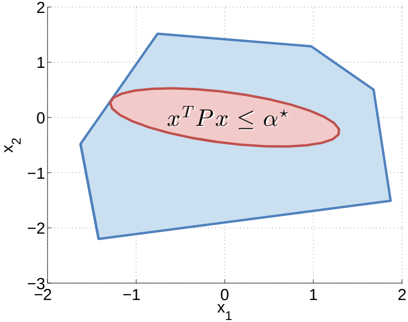
$$h_{Y_\alpha}(\gamma) = \gamma^T P^{-1/2} \frac{P^{-1/2}\gamma}{\|P^{-1/2}\gamma\|}\alpha = \|P^{-1/2}\gamma\|\alpha$$

# Maximum Ellipsoidal Invariant Sets

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

$$\begin{aligned}\alpha^\star &= \max_{\alpha} \ \alpha \quad \text{s.t.} \ \ \|P^{-1/2}F_i^T\|\alpha \le f_i \text{ for all } i \in \{1,\dots,n\} \\ &= \min_{i\in\{1,\dots,n\}} \frac{f_i}{\|P^{-1/2}F_i^T\|}\end{aligned}$$




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

# Doing Better than the LQR Lyapunov Function

The function $V(x(k)) = x(k)^T P x(k)$ is only one of many possible Lyapunov functions for the system $x(k+1) = (A+BK)x(k)$. Can we find one that will give a larger ellipse?

The function $V(x(k)) = x(k)^T P x(k)$ is a Lyapunov function for the system $x(k+1) = (A+BK)x(k)$ if it satisfies the Lyapunov equation

$$A^T PA - P = -Q$$

for some $Q \succeq 0$. This condition is equivalent to the **convex constraint** on $P$:

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

where $\preceq$ means 'negative definite'.

Note that this is equivalent to the condition

$$P^{-1} A^T PAP^{-1} - P^{-1} \preceq 0$$

(multiply left and right by $P^{-1}$)

# Doing Better than the LQR Lyapunov Function

We can now pose a convex optimization problem, which returns the largest invariant ellipse within a polytope $\mathbb{X} = \{x \mid Fx \le f\}$ (where we define $\tilde{P} := P^{-1}$)

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

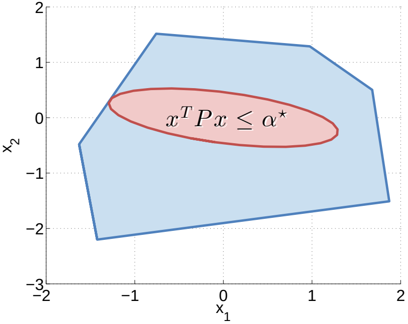
Notes:

- The volume of an ellipse is proportional to $\left(\det P^{-1}\right)^{\frac{1}{2}}$
- $\|P^{-1/2}F_i^T\|^2 = F_i P^{-1} F_i^T$
- $P^{-1}A^T P A P^{-1} - P^{-1} \preceq 0 \Leftrightarrow \left[\begin{smallmatrix} P^{-1} & P^{-1}A^T \\ AP^{-1} & P^{-1} \end{smallmatrix}\right] \succeq 0$ (Schur complement)

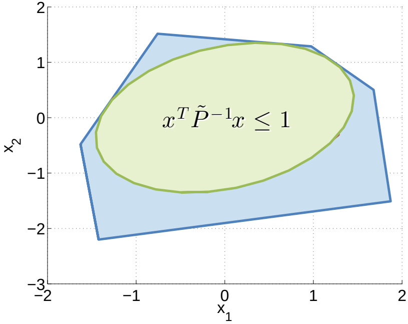
The largest volume ellipse centered at zero within the polytope $\mathbb{X}$ is then

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

# Example Revisited




Maximum volume ellipse using the matrix $P$ from LQR.




Maximum volume ellipse resulting from any quadratic Lyapunov function.