

Model Predictive Control

Lecture: Introduction to Constrained Systems

Colin Jones

Laboratoire d'Automatique, EPFL

Recap: Introduction to Convex Optimization

$$\begin{aligned} & \min f(z) \\ \text{s.t. } & g(z) \leq 0 \\ & Cz = b \end{aligned}$$

where $C \in \mathbb{R}^{p \times n}$ and $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$.

Main point: Easy to solve if f , g convex and twice differentiable.

MPC problems with these properties are common:

- Linear dynamics
- Convex cost and constraints (e.g., quadratic, linear, sum of norms, etc)

Interior-point methods

- Solve linear system to compute search direction, take step, repeat:

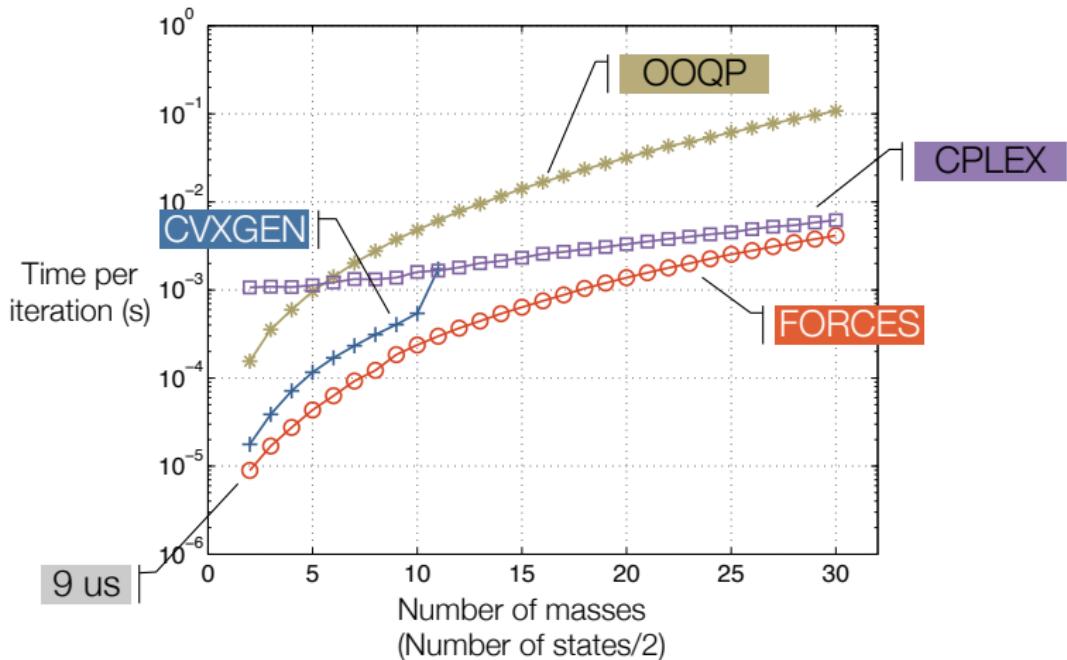
$$(\nabla^2 f(z) + \kappa \nabla^2 \phi(z)) \Delta z_{nt} = -\nabla f(z) - \kappa \nabla \phi(z)$$

- General form of linear system more complex, but same idea
- Generally have to iterate about 10 – 30 times
- Computation time determined by how long it takes to solve linear equation
 - A *lot* faster if structure of problem is exploited to speed solution

MPC can be used for Fast Systems

Time per iteration for MPC problem on desktop PC.

Total time will be $\sim 10\times$ slower.



Outline

1. Objectives of Constrained Control
2. Invariance
3. Controlled Invariance
4. Polytopes and Polytopic Computation
5. Summary
6. Summary of Exercise Session
7. Ellipsoids and Invariance (Not on Exam, but may be useful to you)

Constrained Control

$$x^+ = f(x, u) \quad (x, u) \in \mathbb{X}, \mathbb{U}$$

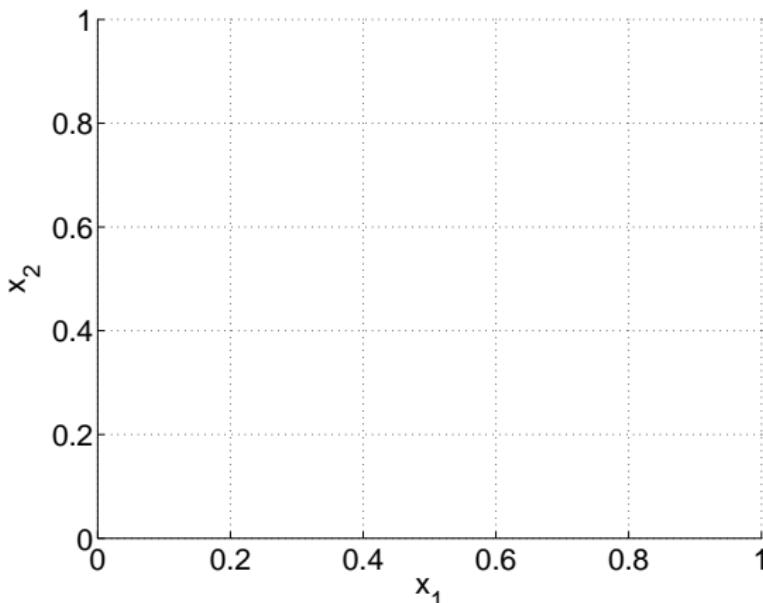
Design control law $u = \kappa(x)$ such that the system:

1. Satisfies constraints : $\{x_i\} \subset \mathbb{X}$, $\{u_i\} \subset \mathbb{U}$
2. Is stable: $\lim_{i \rightarrow \infty} x_i = 0$
3. Optimizes “performance”
4. Maximizes the set $\{x_0 \mid \text{Conditions 1-3 are met}\}$

This lecture is about how to ensure #1

(Remaining lectures cover 2-4)

Limitations of Linear Controllers



Does linear control work?

System:

$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u$$

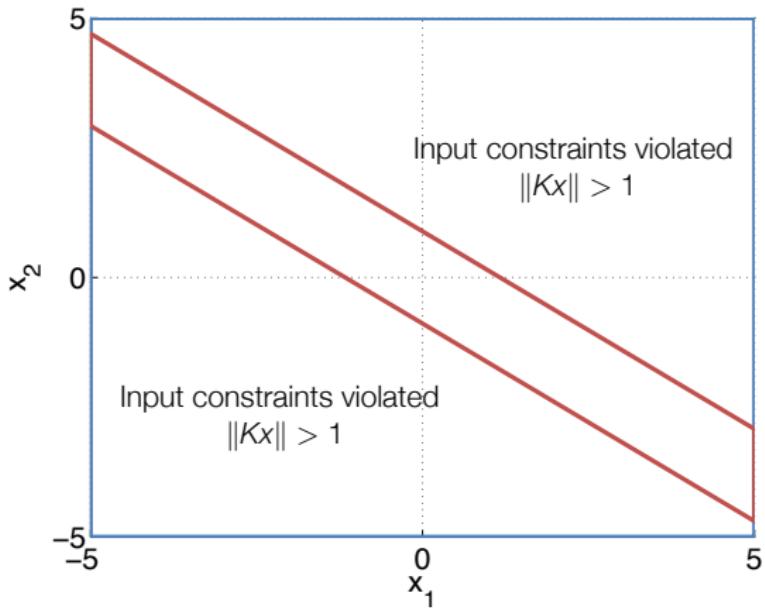
Constraints:

$$\mathbb{X} := \{x \mid \|x\|_\infty \leq 5\}$$

$$\mathbb{U} := \{u \mid \|u\|_\infty \leq 1\}$$

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

Limitations of Linear Controllers



System:

$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u$$

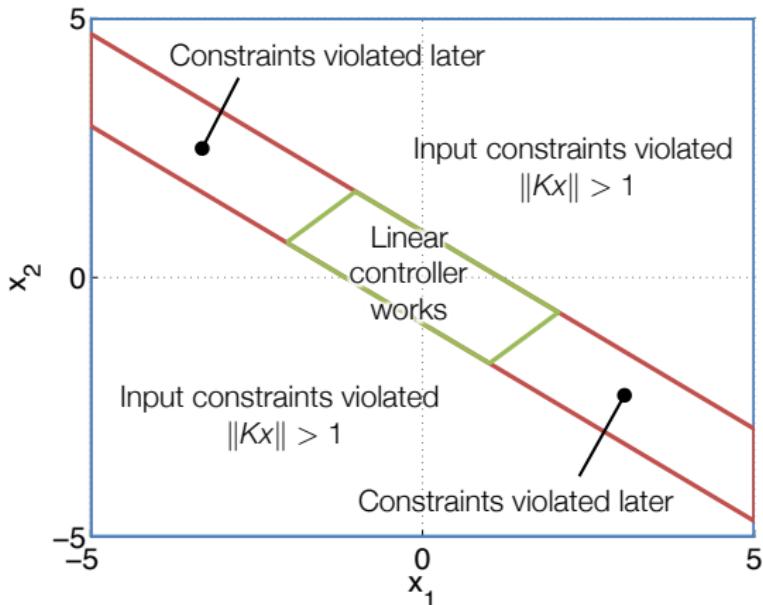
Constraints:

$$\mathbb{X} := \{x \mid \|x\|_\infty \leq 5\}$$

$$\mathbb{U} := \{u \mid \|u\|_\infty \leq 1\}$$

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

Limitations of Linear Controllers



Does linear control work?

Yes, but the region where it works is very small

System:

$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u$$

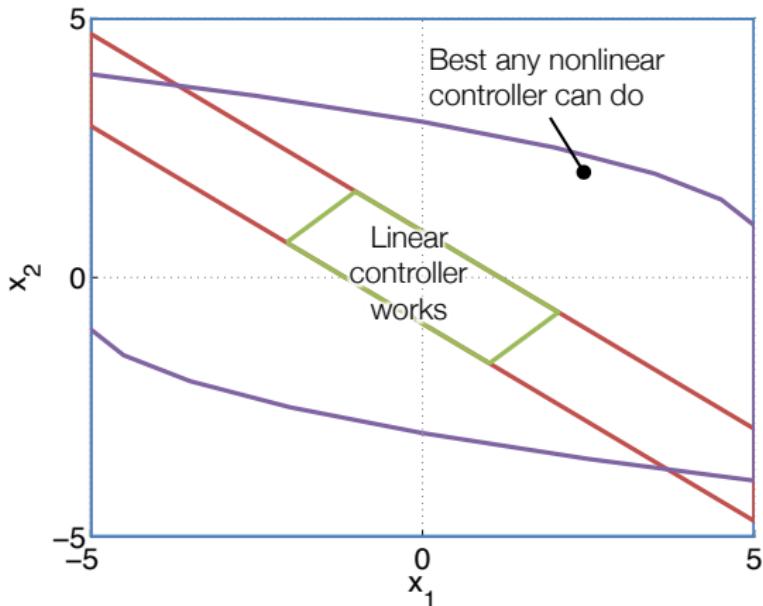
Constraints:

$$\mathbb{X} := \{x \mid \|x\|_\infty \leq 5\}$$

$$\mathbb{U} := \{u \mid \|u\|_\infty \leq 1\}$$

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

Limitations of Linear Controllers



Does linear control work?

Yes, but the region where it works is very small

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

System:

$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u$$

Constraints:

$$\mathbb{X} := \{x \mid \|x\|_\infty \leq 5\}$$

$$\mathbb{U} := \{u \mid \|u\|_\infty \leq 1\}$$

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

Lecture Take Homes

Two concepts this lecture:

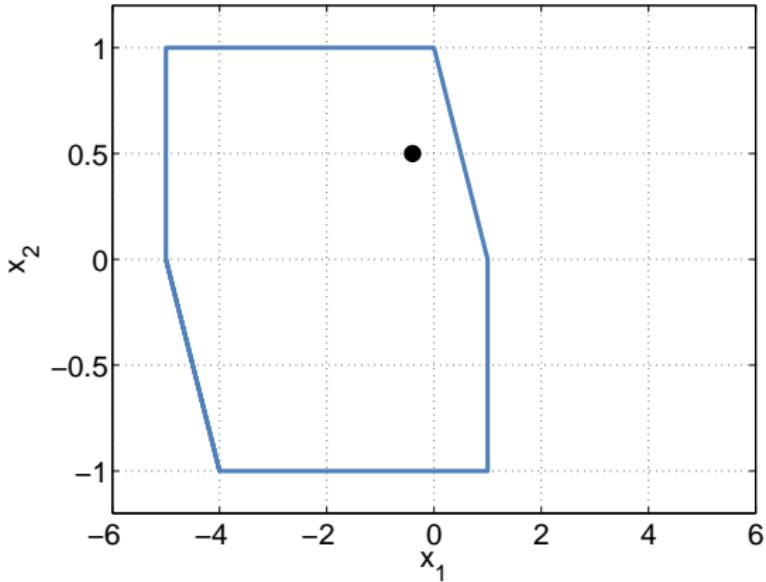
- Invariance
 - Region in which an **autonomous** system will satisfy the constraints **for all time**
- Controlled invariance
 - Region for which there **exists a controller** so that the system satisfies the constraints **for all time**

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:

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

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

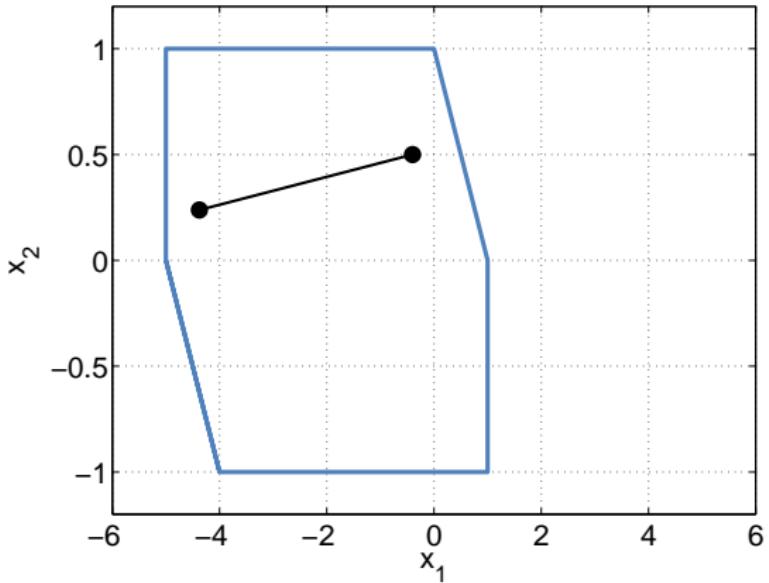
Constraints:

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

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.

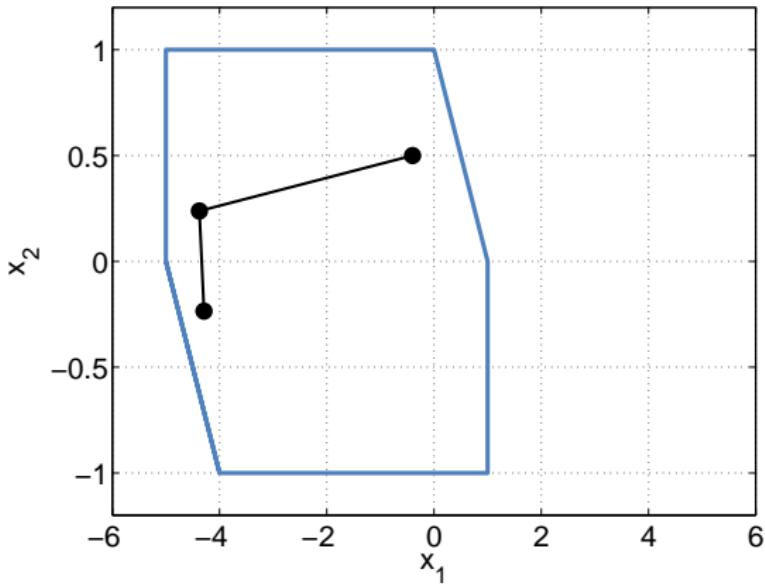
Constraints:

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

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.

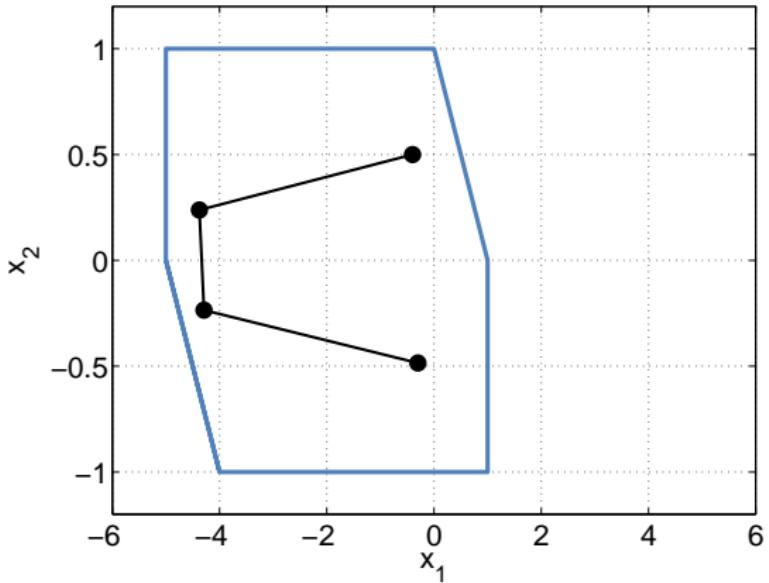
Constraints:

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

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.

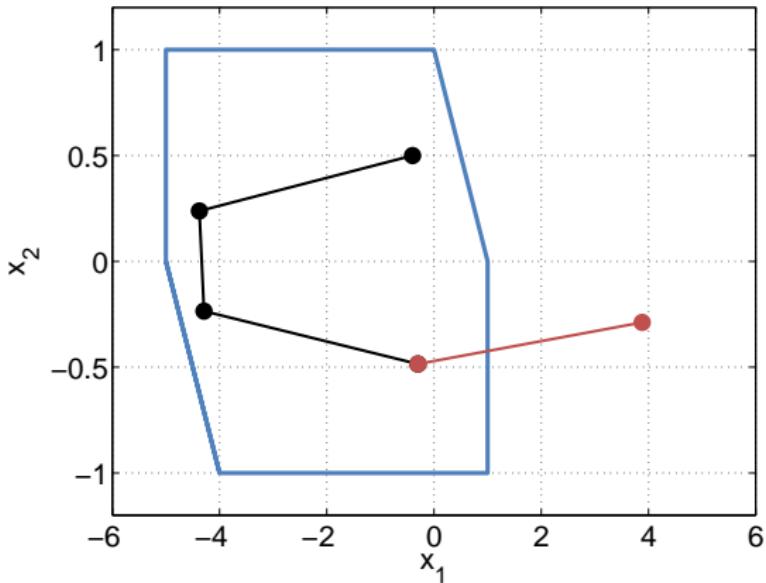
Constraints:

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

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.

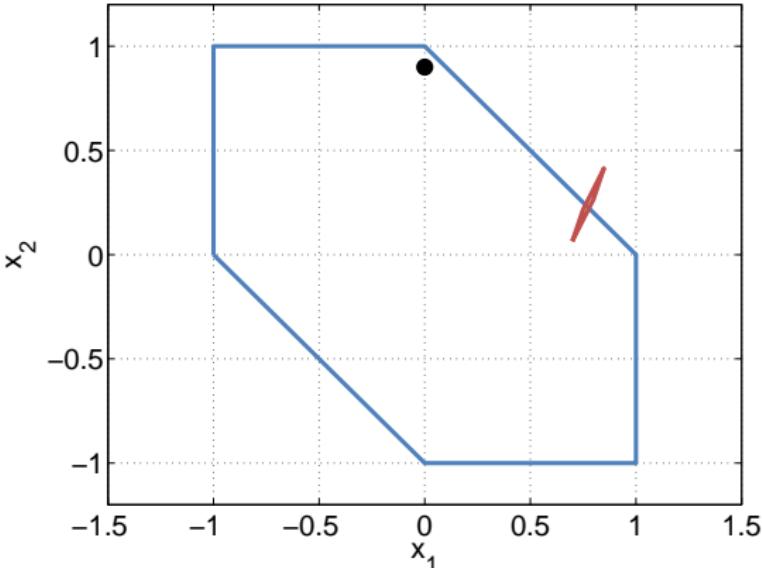
Constraints:

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

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?



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

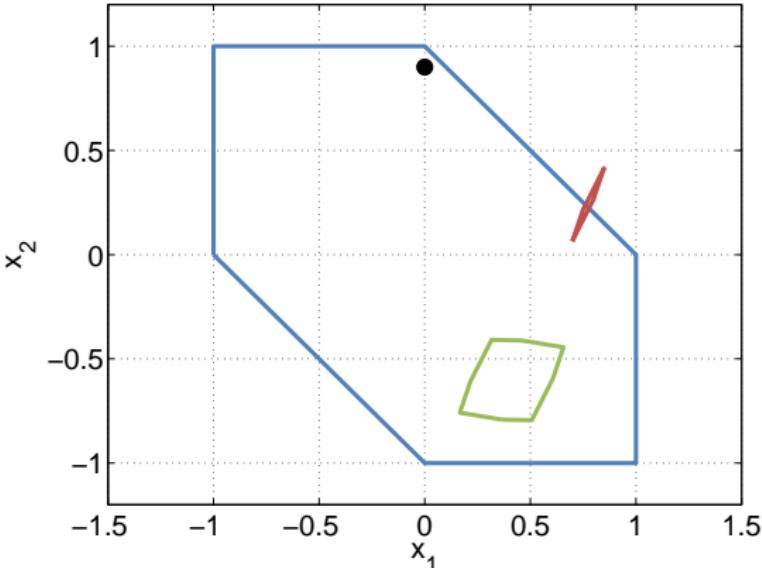
Constraints:

$$\|u\|_\infty \leq 0.1 \\ \|x\|_\infty \leq 1 \\ \|[1 \quad 1] x\|_\infty \leq 1$$

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?



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

Constraints:

$$\|u\|_\infty \leq 0.1$$

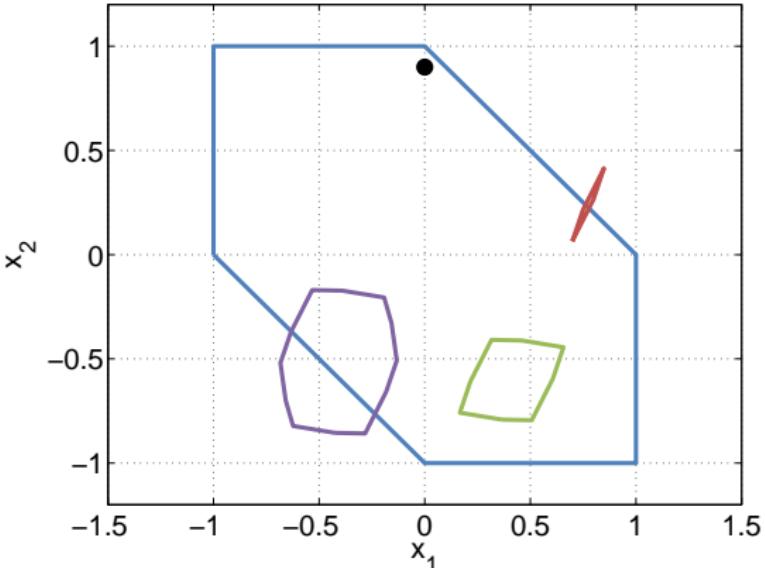
$$\|x\|_\infty \leq 1$$

$$\left\| \begin{bmatrix} 1 & 1 \end{bmatrix} x \right\|_\infty \leq 1$$

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?



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

Constraints:

$$\|u\|_\infty \leq 0.1$$

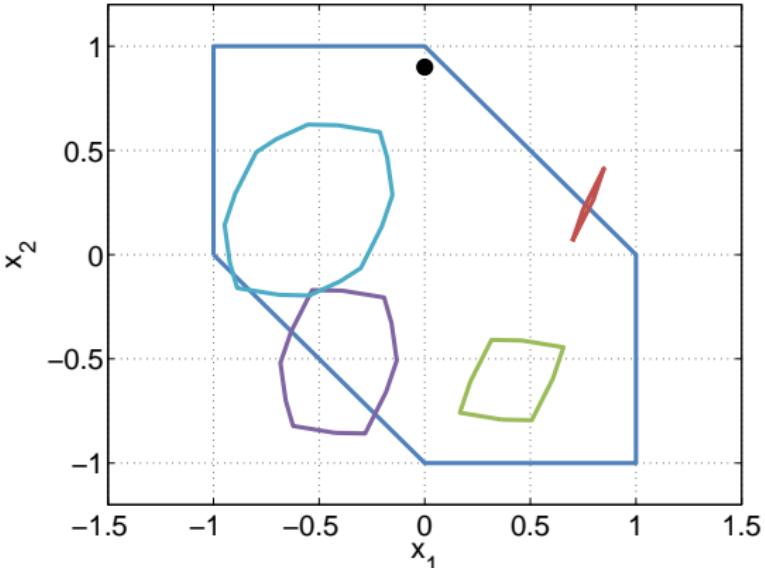
$$\|x\|_\infty \leq 1$$

$$\| [1 \quad 1] x \|_\infty \leq 1$$

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?



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

Constraints:

$$\|u\|_\infty \leq 0.1$$

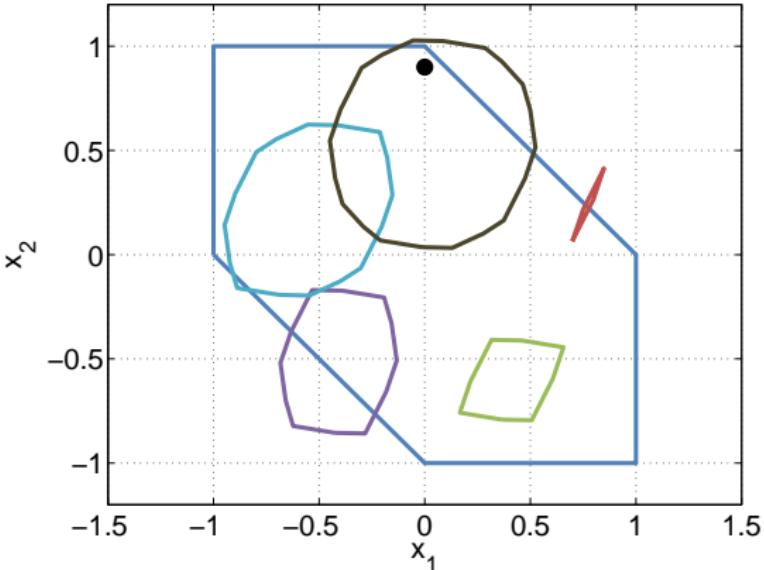
$$\|x\|_\infty \leq 1$$

$$\| [1 \quad 1] x \|_\infty \leq 1$$

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?



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

Constraints:

$$\|u\|_\infty \leq 0.1$$

$$\|x\|_\infty \leq 1$$

$$\| [1 \quad 1] x \|_\infty \leq 1$$

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

1. Objectives of Constrained Control
2. Invariance
3. Controlled Invariance
4. Polytopes and Polytopic Computation
5. Summary
6. Summary of Exercise Session
7. Ellipsoids and Invariance (Not on Exam, but may be useful to you)

Invariance

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

Positive Invariant set

A set \mathcal{O} is said to be a positive invariant set for the autonomous system $x_{i+1} = f(x_i)$ if

$$x_i \in \mathcal{O} \Rightarrow x_i \in \mathcal{O}, \quad \forall i \in \{0, 1, \dots\}$$

The invariant set provides a set of initial states from which the trajectory will never violate the system constraints.

Maximal Positive Invariant Set \mathcal{O}_∞

The set $\mathcal{O}_\infty \subset \mathbb{X}$ is the maximal invariant set with respect to \mathbb{X} if $0 \in \mathcal{O}_\infty$, \mathcal{O}_∞ is invariant and \mathcal{O}_∞ 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}_∞ .

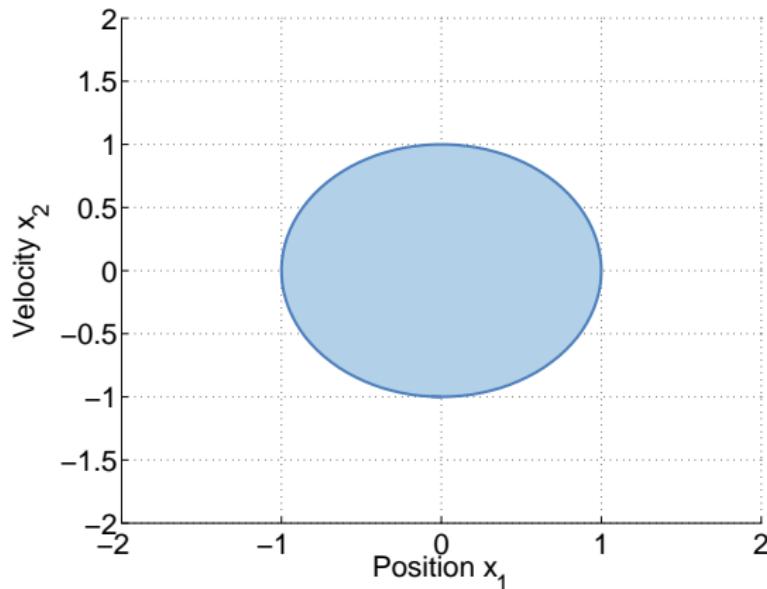
Pre-Sets

Pre Set

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

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

Pre-Set Example : Pendulum



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

Pendulum:

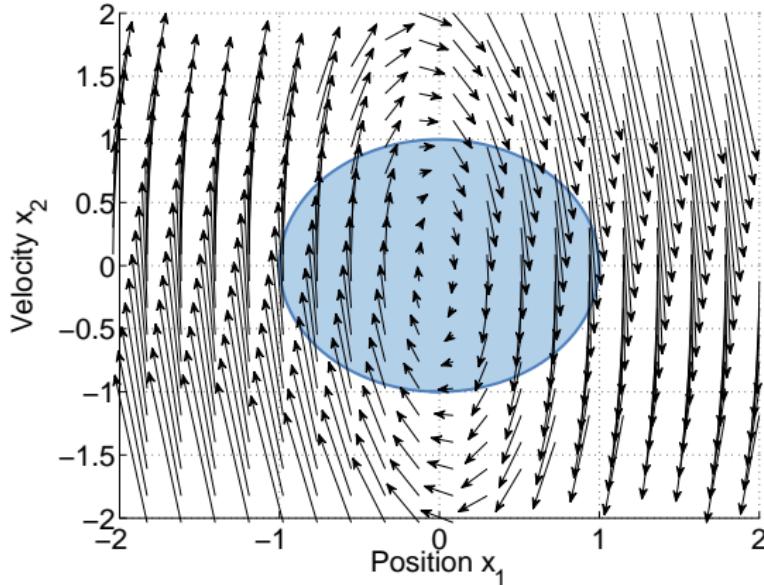
$$x^+ = x + \begin{bmatrix} x_2 \\ -9.8 \sin x_1 - x_2 \end{bmatrix}$$

(Discretized with forward Euler
at 1Hz)

Target set:

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

Pre-Set Example : Pendulum



Pendulum:

$$x^+ = x + \begin{bmatrix} x_2 \\ -9.8 \sin x_1 - x_2 \end{bmatrix}$$

(Discretized with forward Euler
at 1Hz)

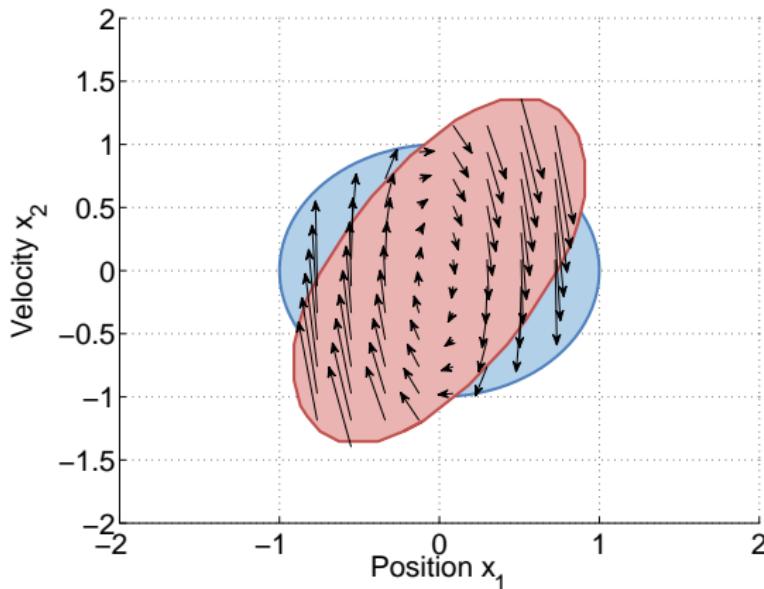
Target set:

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

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

Consider the phase diagram.

Pre-Set Example : Pendulum



Pendulum:

$$x^+ = x + \begin{bmatrix} x_2 \\ -9.8 \sin x_1 - x_2 \end{bmatrix}$$

(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 (next lecture).

Invariant Set Conditions

Theorem: Geometric condition for invariance

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

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

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

Necessary If $\mathcal{O} \not\subseteq \text{pre}(\mathcal{O})$, then $\exists \bar{x} \in \mathcal{O}$ such that $\bar{x} \notin \text{pre}(\mathcal{O})$. From the definition of $\text{pre}(\mathcal{O})$, $f(\bar{x}) \notin \mathcal{O}$ 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 $\bar{x} \notin \text{pre}(\mathcal{O})$ and thus $\mathcal{O} \not\subseteq \text{pre}(\mathcal{O})$.



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

Computing Invariant Sets

Conceptual Algorithm to Compute Invariant Set

Input: f, \mathbb{X}

Output: \mathcal{O}_∞

$\Omega_0 \leftarrow \mathbb{X}$

loop

$\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$

if $\Omega_{i+1} = \Omega_i$ **then**

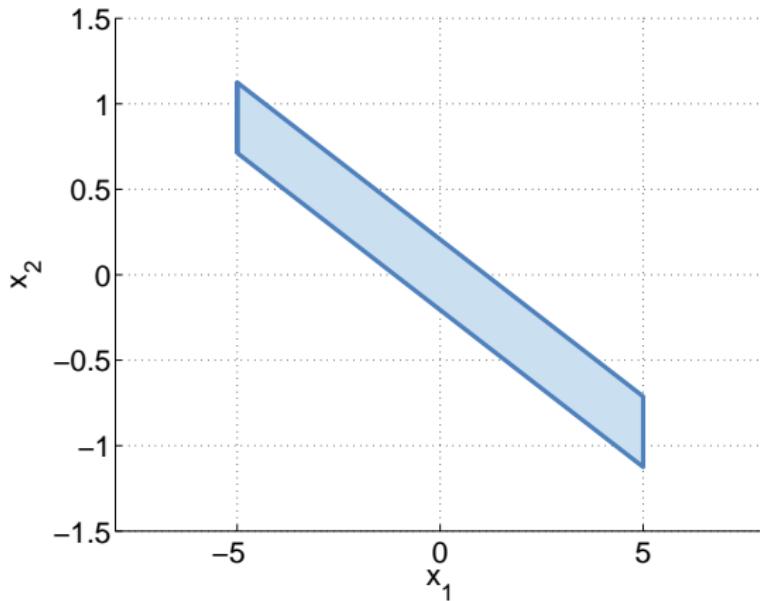
return $\mathcal{O}_\infty = \Omega_i$

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}_∞ for $x^+ = f(x)$.

Computing Invariant Sets



System:

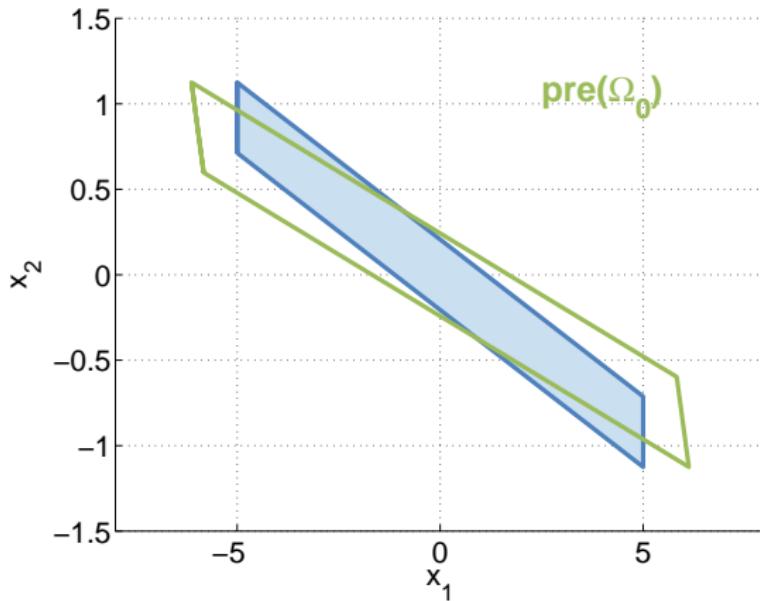
$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u \quad \begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix} \quad -0.1 \leq u \leq 0.1$$

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

Input: f, \mathbb{X}
Output: \mathcal{O}_∞

```
 $\Omega_0 \leftarrow \mathbb{X}$ 
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
```

Computing Invariant Sets



System:

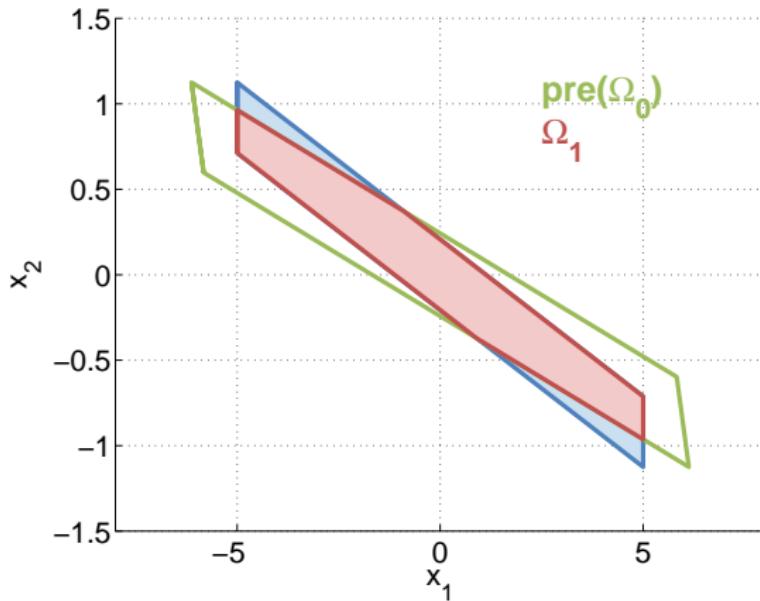
$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u \quad \begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix} \quad -0.1 \leq u \leq 0.1$$

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

Input: f, \mathbb{X}
Output: \mathcal{O}_∞

```
 $\Omega_0 \leftarrow \mathbb{X}$ 
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
```

Computing Invariant Sets



System:

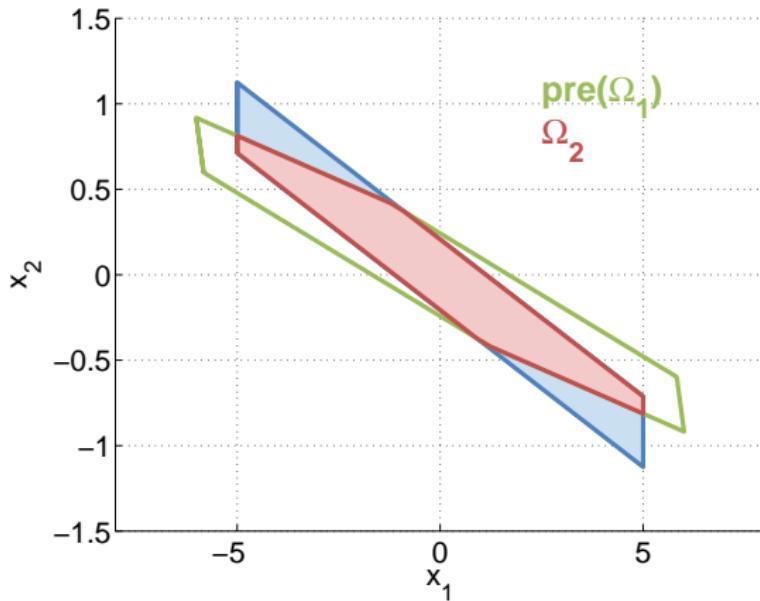
$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u \quad \begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix} \quad -0.1 \leq u \leq 0.1$$

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

Input: f, \mathbb{X}
Output: \mathcal{O}_∞

```
 $\Omega_0 \leftarrow \mathbb{X}$ 
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
```

Computing Invariant Sets



System:

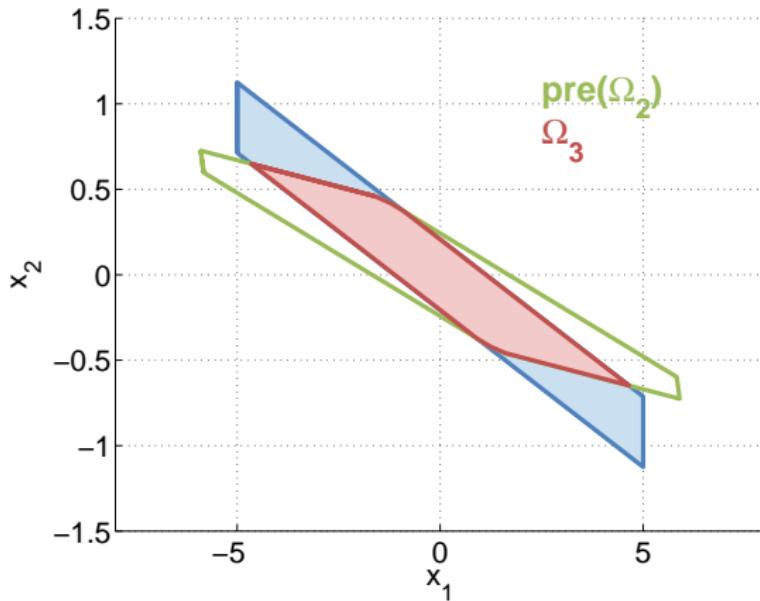
$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u \quad \begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix} \quad -0.1 \leq u \leq 0.1$$

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

Input: f, \mathbb{X}
Output: \mathcal{O}_∞

```
 $\Omega_0 \leftarrow \mathbb{X}$ 
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
```

Computing Invariant Sets



System:

$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u \quad \begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix} \quad -0.1 \leq u \leq 0.1$$

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

Input: f, \mathbb{X}
Output: \mathcal{O}_∞

$$\Omega_0 \leftarrow \mathbb{X}$$

loop

$$\Omega_{i+1} \leftarrow \text{pre}(\Omega_i) \cap \Omega_i$$

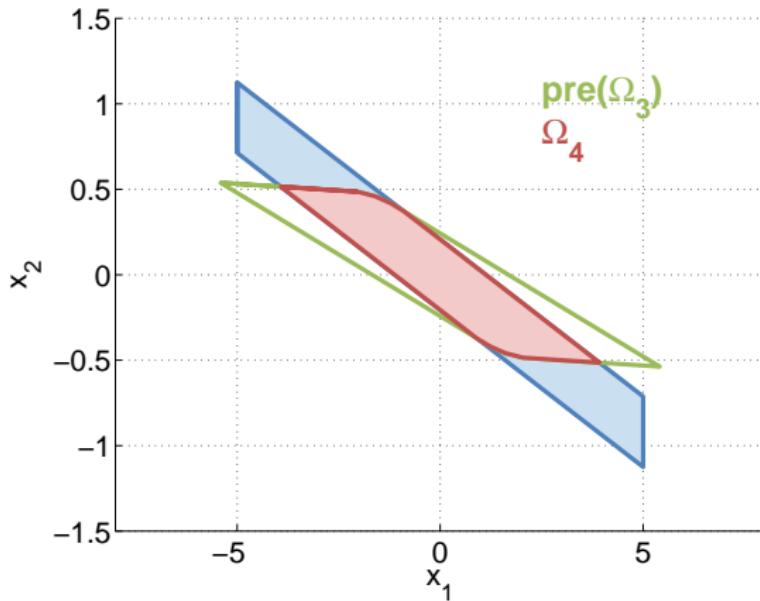
if $\Omega_{i+1} = \Omega_i$ **then**

return $\mathcal{O}_\infty = \Omega_i$

end if

end loop

Computing Invariant Sets



System:

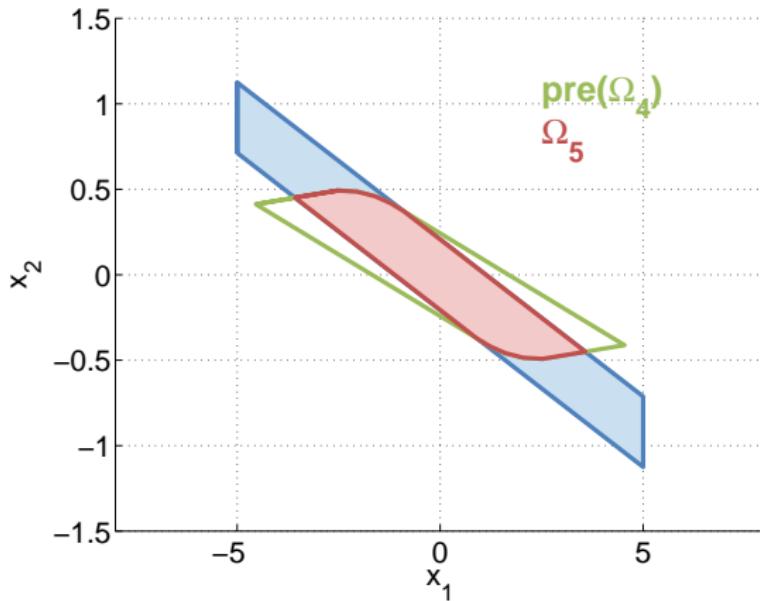
$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u \quad \begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix} \quad -0.1 \leq u \leq 0.1$$

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

Input: f, \mathbb{X}
Output: \mathcal{O}_∞

```
 $\Omega_0 \leftarrow \mathbb{X}$ 
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
```

Computing Invariant Sets



System:

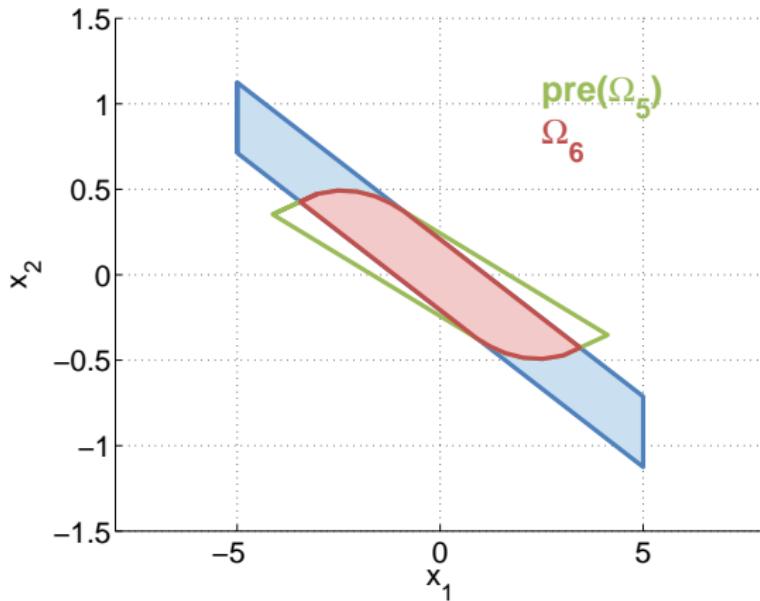
$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u \quad \begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix} \quad -0.1 \leq u \leq 0.1$$

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

Input: f, \mathbb{X}
Output: \mathcal{O}_∞

```
 $\Omega_0 \leftarrow \mathbb{X}$ 
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
```

Computing Invariant Sets



System:

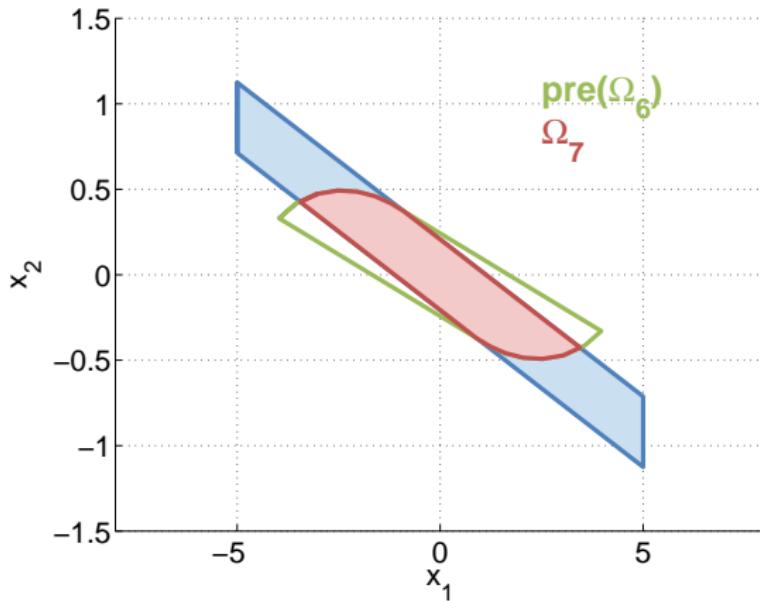
$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u \quad \begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix} \quad -0.1 \leq u \leq 0.1$$

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

Input: f, \mathbb{X}
Output: \mathcal{O}_∞

```
 $\Omega_0 \leftarrow \mathbb{X}$ 
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
```

Computing Invariant Sets



System:

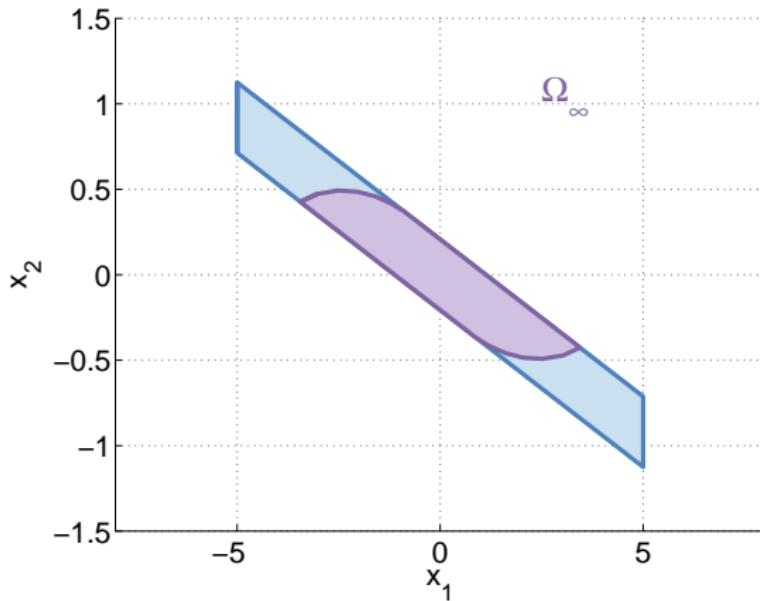
$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u \quad \begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix} \quad -0.1 \leq u \leq 0.1$$

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

Input: f, \mathbb{X}
Output: \mathcal{O}_∞

```
 $\Omega_0 \leftarrow \mathbb{X}$ 
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
```

Computing Invariant Sets



System:

$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u \quad \begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix} \quad -0.1 \leq u \leq 0.1$$

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

Input: f, \mathbb{X}
Output: \mathcal{O}_∞

```
 $\Omega_0 \leftarrow \mathbb{X}$ 
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
```

Outline

1. Objectives of Constrained Control
2. Invariance
3. Controlled Invariance
4. Polytopes and Polytopic Computation
5. Summary
6. Summary of Exercise Session
7. Ellipsoids and Invariance (Not on Exam, but may be useful to you)

Controlled Invariance

Control Invariant Set

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

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

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

Maximal Control Invariant Set \mathcal{C}_∞

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

Conceptual Calculation of Control Invariant Sets

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

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

The same iterative condition holds for control invariant sets

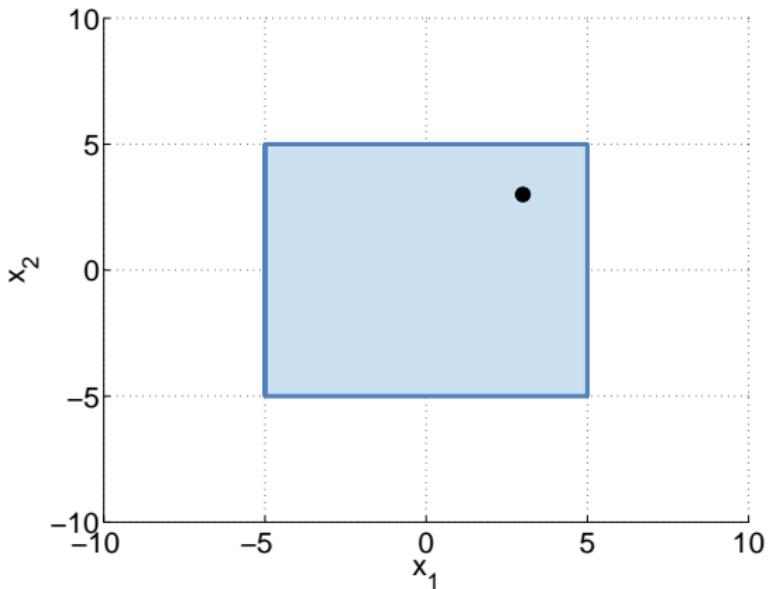
A set \mathcal{C} is a control invariant set if and only if $\mathcal{C} \subseteq \text{pre}(\mathcal{C})$

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

```
 $\Omega_0 \leftarrow \mathbb{X}$ 
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!
(Will go through details for linear systems / constraints in next lecture)

Computing Control Invariant Sets



System:

$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u$$

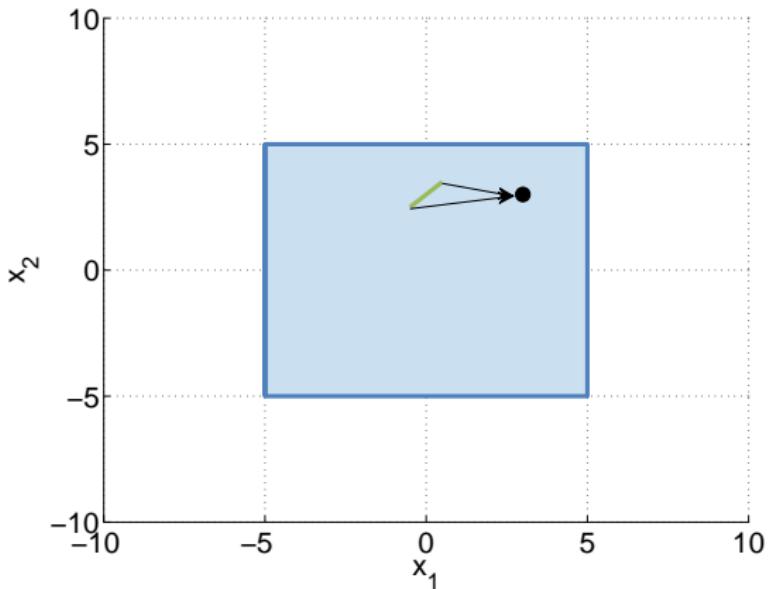
Constraints:

$$\|x\|_\infty \leq 5$$

$$\|u\|_\infty \leq 1$$

An entire set of states can map into each point

Computing Control Invariant Sets

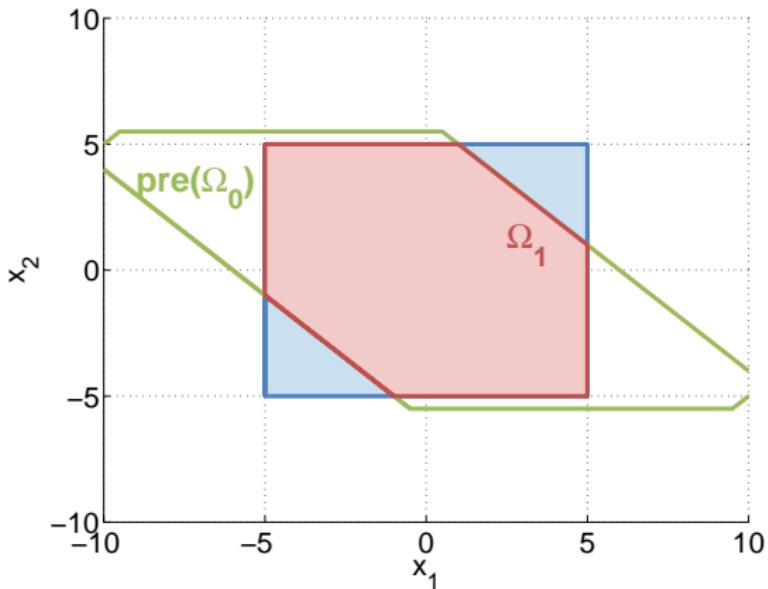


An entire set of states can map into each point

Algorithm:

```
 $\Omega_0 \leftarrow \mathbb{X}$ 
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
```

Computing Control Invariant Sets



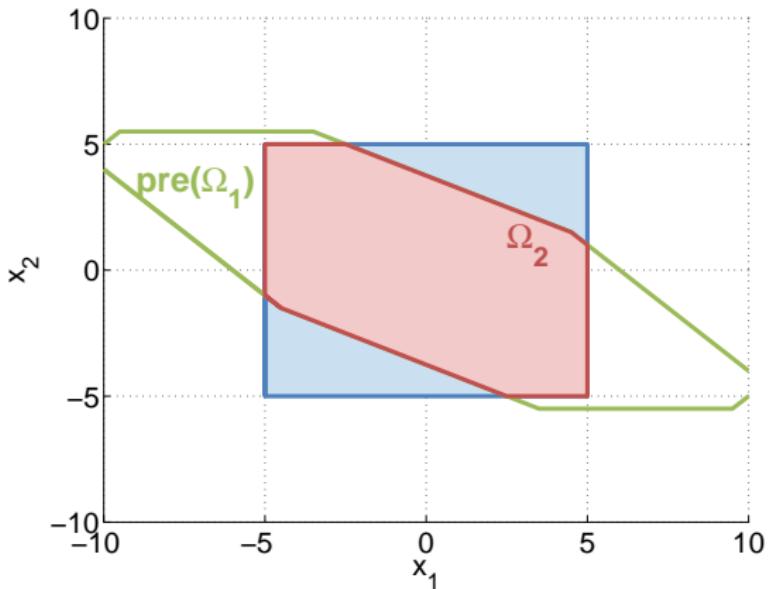
Algorithm:

```
 $\Omega_0 \leftarrow \mathbb{X}$ 
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



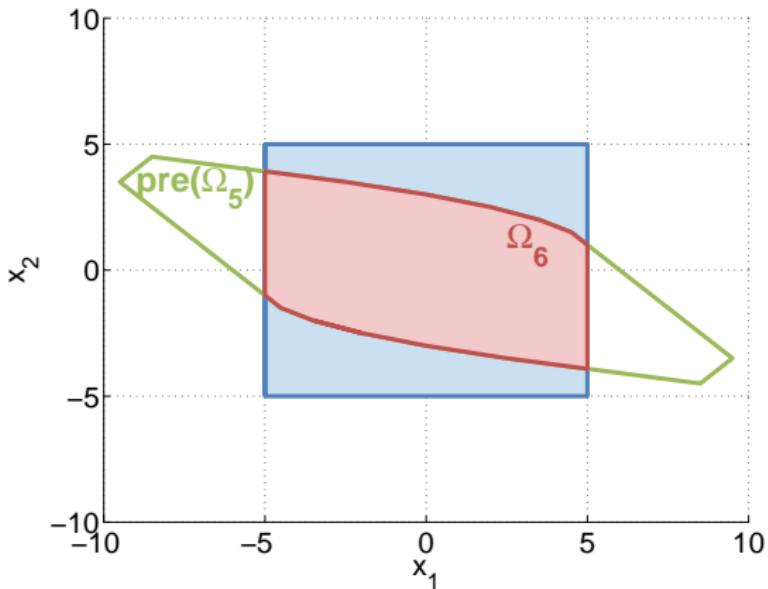
Algorithm:

```
 $\Omega_0 \leftarrow \mathbb{X}$ 
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



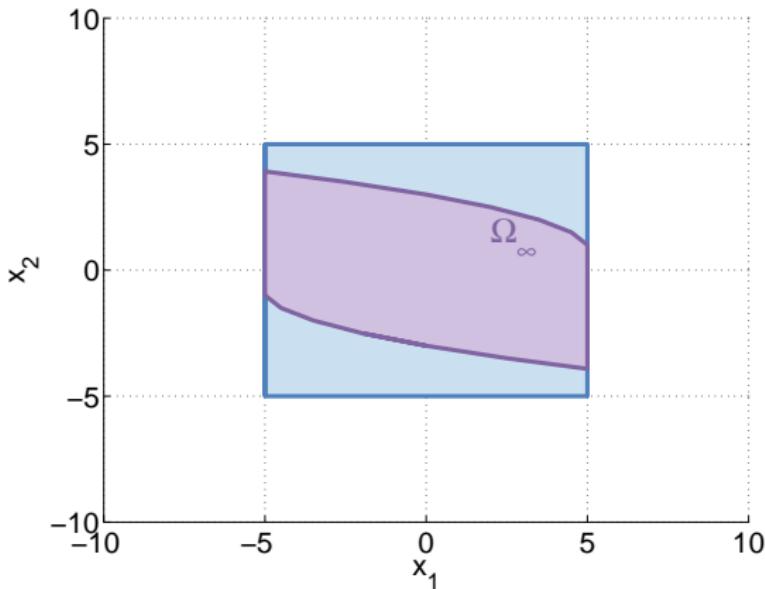
An entire set of states can map into each point

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

Algorithm:

```
 $\Omega_0 \leftarrow \mathbb{X}$ 
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
```

Computing Control Invariant Sets



Algorithm:

```
 $\Omega_0 \leftarrow \mathbb{X}$ 
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 $x^+ = f(x, u)$.

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

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

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

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

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

1. Objectives of Constrained Control
2. Invariance
3. Controlled Invariance
4. Polytopes and Polytopic Computation
5. Summary
6. Summary of Exercise Session
7. Ellipsoids and Invariance (Not on Exam, but may be useful to you)

Computing Invariant Sets

Conceptual Algorithm to Compute Invariant Set

```
 $\Omega_0 \leftarrow \mathbb{X}$ 
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 Ω_i (Polytopes)
- Intersection
- Pre-set computation
- Equality test (bi-directional subset)

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

Polyhedra

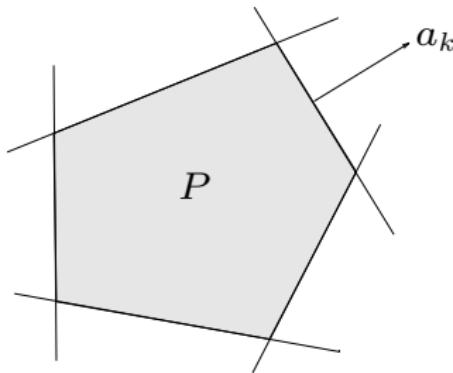
Polyhedron

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

$$P := \{x \mid a_i^T x \leq b_i, i = 1, \dots, n\}$$

A **polytope** is a bounded polyhedron.

Often written as $P := \{x \mid Ax \leq b\}$, 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 $\text{conv}(S)$ 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 .

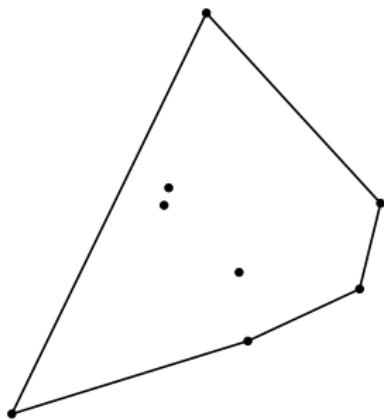
Proposition: Convex hull

The convex hull of a set $S \subseteq \mathbb{R}^d$ is

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

$$\text{conv}(\{v_1, \dots, v_k\}) := \left\{ x \mid x = \sum_i \lambda_i v_i, \lambda_i \geq 0, \sum_i \lambda_i = 1 \forall i = 1, \dots, k \right\}$$

Examples of convex hulls



2D convex hull



3D convex hull

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 $P = \text{conv}(\{v_1, \dots, v_s\})$



=



$$P = \{x \mid Ax \leq b\}$$

$$P = \text{conv}(\{v_1, \dots, v_s\})$$

Most Common Polytopic Constraints

$$x^+ = Ax + Bu$$

$$y = Cx$$

Suppose we have the following input and output constraints:

$$u_{low} \leq u \leq u_{high}$$

$$y_{low} \leq y \leq y_{high}$$

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

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

Polytopes in MPC

Input saturation

$$u_{lb} \leq u \leq u^{ub}$$



$$\begin{bmatrix} 1 \\ -1 \end{bmatrix} u \leq \begin{bmatrix} u^{ub} \\ -u_{lb} \end{bmatrix}$$

Magnitude constraints

$$\|Cx\|_\infty \leq \alpha$$



$$\begin{bmatrix} C \\ -C \end{bmatrix} x \leq \mathbf{1}\alpha$$

Rate constraints

$$\|x_i - x_{i+1}\|_\infty \leq \alpha$$



$$\begin{bmatrix} I & -I \\ -I & I \end{bmatrix} \begin{pmatrix} x_i \\ x_{i+1} \end{pmatrix} \leq \mathbf{1}\alpha$$

Integral constraints

$$\|x\|_1 \leq \alpha$$



$$x \in \text{conv}(e_i\alpha)$$

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 $(0, \dots, 0, 1, 0, \dots, 0)$, with the 1 in the i^{th} position

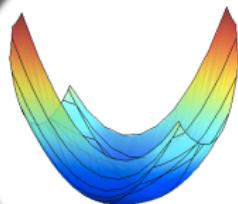
MultiParametric Toolbox (MPT)

Modeling

Optimal control

Analysis

Computational
geometry



<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 $P = \text{conv}(v_i)$

```
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

```
 $\Omega_0 \leftarrow X$ 
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 Ω_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

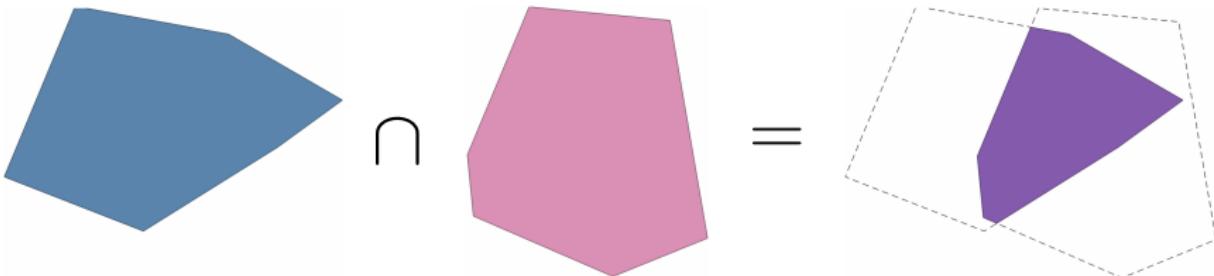
$$I = S \cap T := \{x \mid x \in S \text{ and } x \in T\}$$

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

$$S := \{x \mid Cx \leq c\}$$

$$T := \{x \mid Dx \leq d\}$$

$$S \cap T = \left\{ x \mid \begin{bmatrix} C \\ D \end{bmatrix} x \leq \begin{bmatrix} c \\ d \end{bmatrix} \right\}$$



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

Computing Invariant Sets

Conceptual Algorithm to Compute Invariant Set

```
 $\Omega_0 \leftarrow \mathbb{X}$ 
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 Ω_i (Polytopes)
- Intersection
- **Pre-set computation**
- Equality test (bi-directional subset)

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

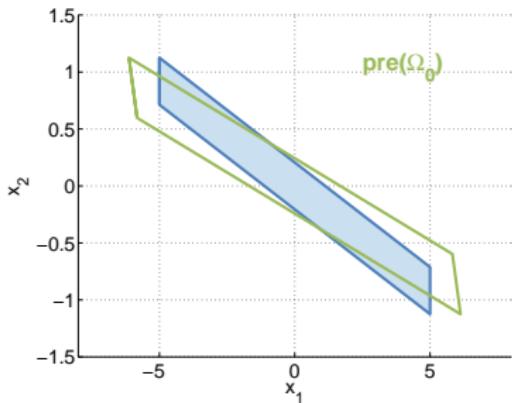
Pre-Set Computation: Autonomous System

Pre Set

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

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

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



$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u$$
$$\begin{bmatrix} -5 \\ -10 \end{bmatrix} \leq x \leq \begin{bmatrix} 5 \\ 10 \end{bmatrix}, \quad \|u\|_\infty \leq 0.1$$

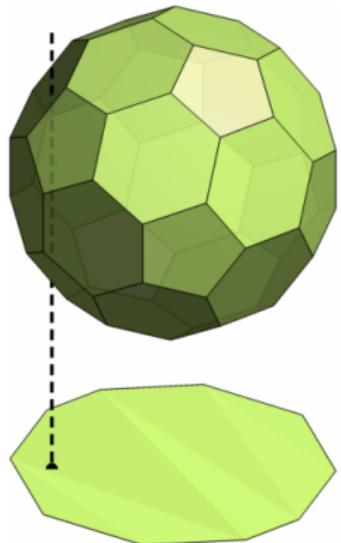
Where $u = Kx$, with K the optimal LQR controller for $Q = I$, $R = 90$.

Pre-Set Computation: Controlled System

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

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

This is a **projection** operation.



Polytopic Projection

Polytopic Projection

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

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

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 (My PhD)

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

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

Polytopic Projections in MPT

Several projection algorithms are implemented in MPT.

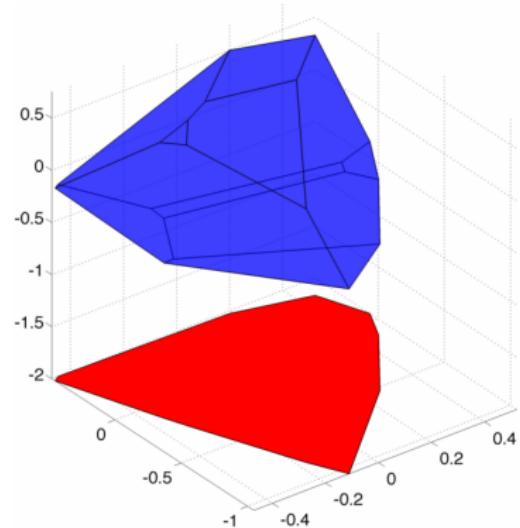
The best is a function of the dimension, complexity and numerical sensitivity of the polytope being projected. For the most part, the defaults work well.

```
% 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

```
 $\Omega_0 \leftarrow \mathbb{X}$ 
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:

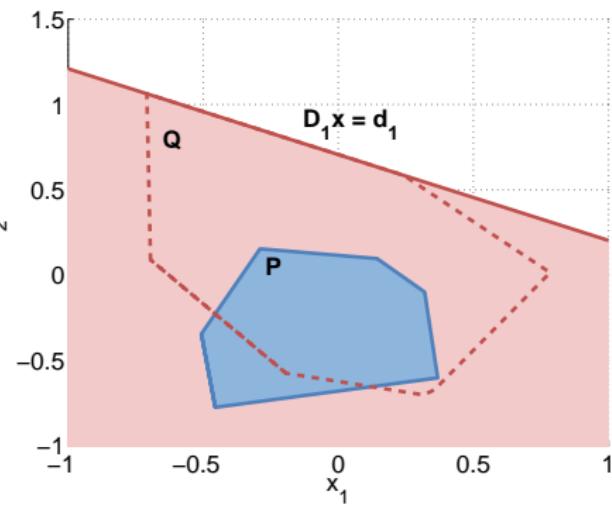
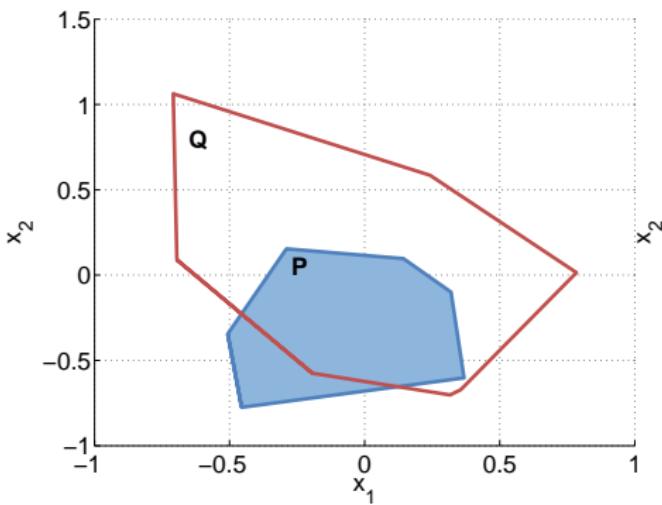
- Represent set Ω_i (Polytopes)
- Intersection
- Pre-set computation
- **Equality test (bi-directional subset)**

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

Subset Test

Problem: Is $P := \{x \mid Cx \leq c\}$ contained in $Q := \{x \mid Dx \leq d\}$?

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

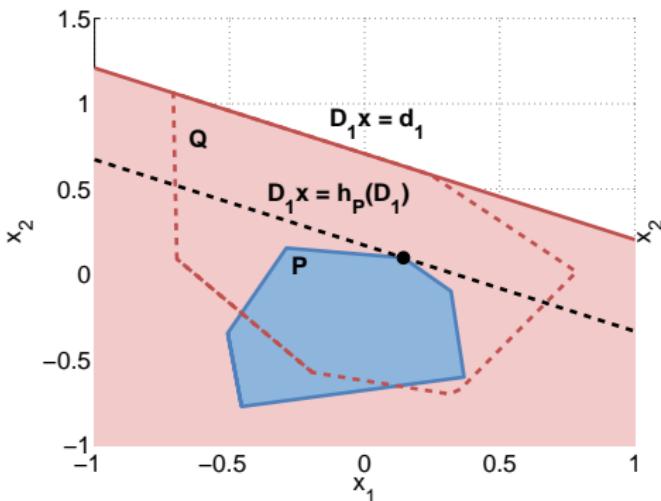


Subset Test

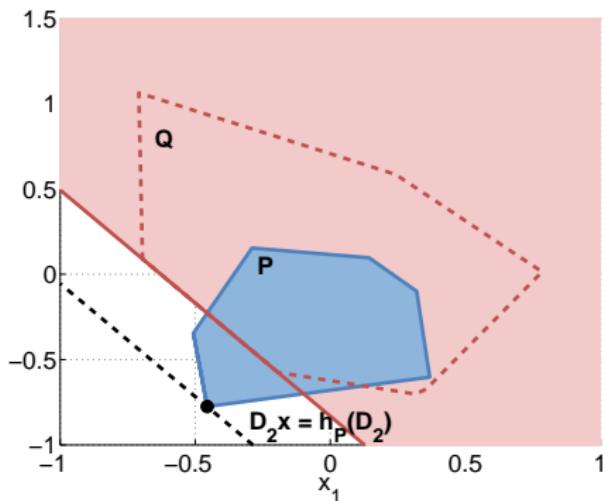
Define the **support function** of the set P :

$$h_P(D_i) := \max_x D_i x$$

s.t. $Cx \leq c$



$$h_P(D_1) \leq d_1 \quad \checkmark$$



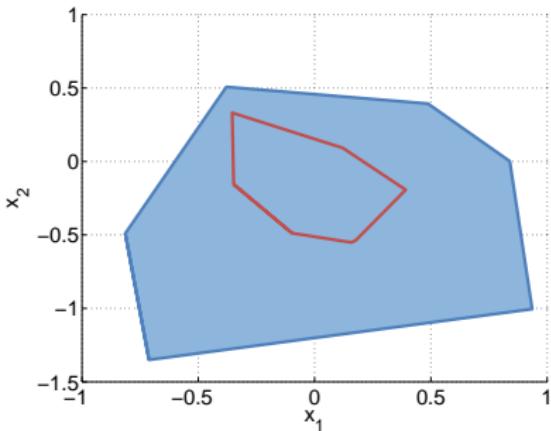
$$h_P(D_2) > d_2 \quad \times$$

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

```
 $\Omega_0 \leftarrow \mathbb{X}$ 
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 Ω_i (Polytopes)
- Intersection
- Pre-set computation
- Equality test (bi-directional subset)

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

Outline

1. Objectives of Constrained Control
2. Invariance
3. Controlled Invariance
4. Polytopes and Polytopic Computation
5. Summary
6. Summary of Exercise Session
7. Ellipsoids and Invariance (Not on Exam, but may be useful to you)

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

Next week:

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

Outline

1. Objectives of Constrained Control
2. Invariance
3. Controlled Invariance
4. Polytopes and Polytopic Computation
5. Summary
6. Summary of Exercise Session
7. Ellipsoids and Invariance (Not on Exam, but may be useful to you)

Exercise Session #3

Tasks:

1. Compute maximal invariant sets
2. Compute maximal control invariant sets

(You may find the next two slides useful!)

Installing MPT3

1. Goto <http://control.ee.ethz.ch/~mpt/3/Main/Installation>
2. Download and run `install_mpt3.m`

Summary: Operations in MPT

Define the polytopes:

$$P := \{x \mid Cx \leq c\}$$

$$Q := \{x \mid Dx \leq d\}$$

Intersection

```
S = Polyhedron([C;D], [c;d]); % S is the intersection of P and Q
```

Subset / equality test

```
P <= Q % True if P is a subset of Q  
Q <= P % True if Q is a subset of P  
P == Q % True if P equals Q
```

Pre-Sets

Autonomous system

$$x^+ = Ax \quad S := \{x \mid Fx \leq f\} \quad \text{pre}(S) = \{x \mid FAx \leq f\}$$

```
pres = Polyhedron(F*A, f); % pres is the pre-set of S
```

Controlled system

$$x^+ = Ax + Bu \quad S := \{x \mid Fx \leq f\} \quad \mathbb{U} := \{u \mid Gu \leq g\}$$

$$\text{pre}(S) = \left\{ x \mid \exists u, \begin{bmatrix} FA & FB \\ 0 & G \end{bmatrix} \begin{pmatrix} x \\ u \end{pmatrix} \leq \begin{bmatrix} f \\ g \end{bmatrix} \right\}$$

```
% Matrix A is n x n and the matrix G is q x m
```

```
pres = projection(Polyhedron([F*A F*B; zeros(q,n) G], [f;g]), [1:n]);
```

Outline

1. Objectives of Constrained Control
2. Invariance
3. Controlled Invariance
4. Polytopes and Polytopic Computation
5. Summary
6. Summary of Exercise Session
7. Ellipsoids and Invariance (Not on Exam, but may be useful to you)

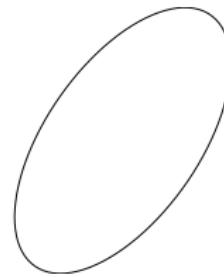
Ellipsoids

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

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

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 $V : \mathbb{R}^n \rightarrow \mathbb{R}$ is a Lyapunov function for the system $x^+ = f(x)$, then

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

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 α 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 $x^+ = f(x)$, and our constraints are given by the set \mathbb{X} , then we maximize α such that

$$Y_\alpha := \{x \mid V(x) \leq \alpha\} \subseteq \mathbb{X}$$

Invariant Sets from Lyapunov Functions

Consider the system $x^+ = Ax$, and assume $P \succ 0$ satisfies the condition

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

Then the function $V(x) = x^T Px$ 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 \mathbb{X} :

$$Y_\alpha := \{x \mid x^T Px \leq \alpha^2\} \subset \mathbb{X} := \{x \mid Fx \leq f\}$$

Equivalently, we want to solve the problem:

$$\max_{\alpha} \alpha$$

$$\text{s.t. } h_{Y_\alpha}(F_i) \leq f_i \text{ for all } i \in \{1, \dots, n\}$$

Maximum Ellipsoidal Invariant Sets

Support of an ellipse:

$$\begin{aligned} h_{Y_\alpha}(\gamma) &= \max_x \gamma^T x \\ \text{s.t. } &x^T P x \leq \alpha^2 \end{aligned}$$

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

$$\begin{aligned} h_{Y_\alpha}(\gamma) &= \max_x \gamma^T P^{-1/2} y \\ \text{s.t. } &y^T y \leq \alpha^2 \end{aligned}$$

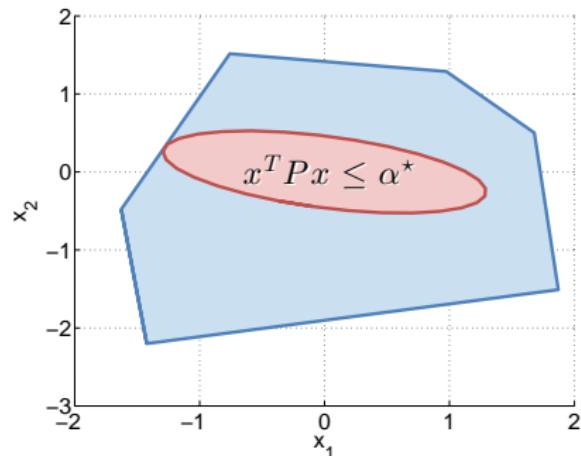
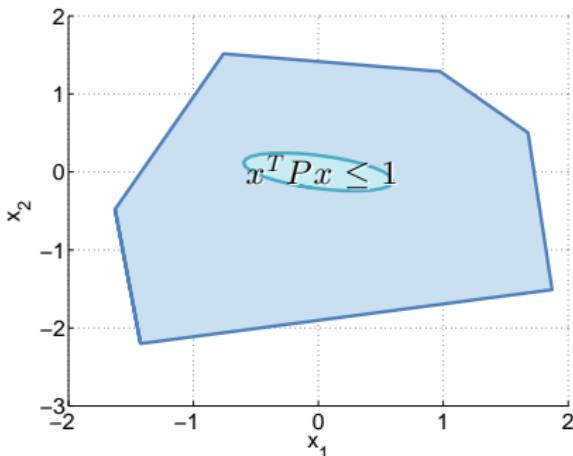
which can be solved by inspection:

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

Maximum Ellipsoidal Invariant Sets

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

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



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

The function $V(x) = x^T Px$ is a Lyapunov function for the system $x^+ = (A + BK)x$ 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

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

(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 $\mathbb{X} = \{x \mid Fx \leq f\}$ (where we define $\tilde{P} := P^{-1}$)

$$\begin{aligned} & \max_{\tilde{P}} -\log \det \tilde{P} \\ \text{s.t. } & A^T \tilde{P} A - \tilde{P} \preceq 0 \\ & F_i \tilde{P} F_i^T \leq f_i^2 \quad \text{for all } i = 1 \dots n \end{aligned}$$

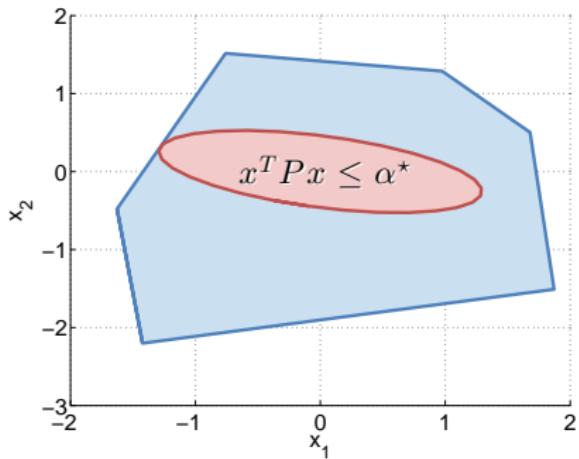
Notes:

- The volume of an ellipse is $\log \det P^{-1}$
- $\|P^{-1/2}F_i^T\|^2 = F_i P^{-1} F_i^T$

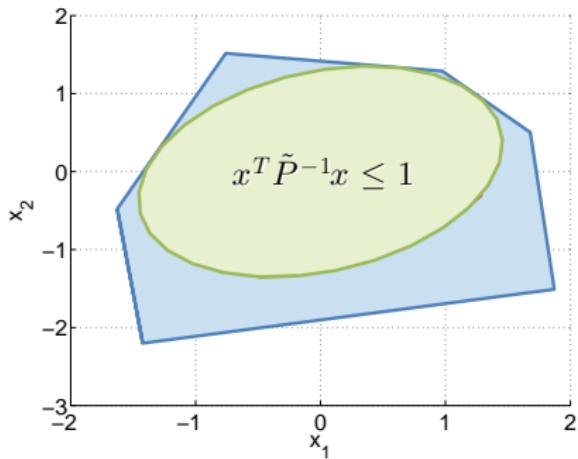
The largest volume ellipse centered at zero within the polytope \mathbb{X} is then

$$\mathcal{E} = \{x \mid x^T \tilde{P}^{-1} x \leq 1\} \subset \mathbb{X}$$

Example Revisited



Maximum volume ellipse using the matrix P from LQR.



Maximum volume ellipse resulting from any quadratic Lyapunov function.