# IEOR 290A – Lecture 19
# Reachability

## 1 Polytope Constraints

So far, we have not considered constraints on our input $u_n$ or states $x_n$, when designing feedback controllers for LTI systems in discrete-time; however, there are many applications in which we would like to impose constraints on both the inputs and states. In some sense, though the constraints often arise from physical or economic limitations on the system being modeled, we get to design the constraints that we use for engineering. Specifically, we can choose any representation of constraints that respects the physical or economic limitations of the system, even if our constraints are more conservative than needed. Such choices can be necessary for the purpose of efficient computation.

In fact, a fairly broad class of constraints with useful computational and mathematical properties can be defined by *bounded convex polytopes*, which can be defined as the convex hull of the set of points. Note that when we refer to polytopes in the future, we will specifically mean bounded convex polytopes. The reason that polytopes are an attractive approach to defining constraints is that they can be represented as the intersection of half-spaces. Recall that a half-space can be represented by $f_i'x \leq h_i$, and so the intersection of half-spaces can be represented by multiple linear inequalities: $Fx \leq h$.

So if we have a polytope $\mathcal{X} = \{x : Fx \leq h\}$, then the constraint that $x_n \in \mathcal{X}$ means that we would like $x_n$ such that $Fx_n \leq h$. We will often refer to constraints on the states and inputs by referring to the polytopes in which they lie; that is, we will ask that $x_n \in \mathcal{X}$ and $u_n \in \mathcal{U}$, where $\mathcal{X}, \mathcal{U}$ are polytopes.

### 1.1 Box Constraints

A common type of constraint are *box constraints*. For a vector $x_n \in \mathbb{R}^p$, a box constraint is that there exists $a_i, b_i$ for $i = 1, \ldots, p$ such that $a_i \leq x_n^i \leq b_i$ for all $i$. It turns out that we can express these constraints as a polytope:

$$x_n \in \mathcal{X} = \{x : x_i \leq b_i, -x_i \leq -a_i, \forall i\}.$$

### 1.2 Linear Transform of Polytope

We define the linear transform $T$ of a polytope $\mathcal{P} = \{u : Fu \leq h\}$ as the polytope

$$T\mathcal{P} = \{Tu : u \in \mathcal{P}\}.$$

## 2 Maximal Output Invariant Sets

Consider an LTI system in discrete time:

$$x_{n+1} = Ax_n + Bu_n,$$

where $(A, B)$ is stabilizable. And assume that we have chosen a $K$ such that using the state-feedback controller $u_n = Kx_n$ leads to a stable system $x_{n+1} = (A + BK)x_n$. Now consider this same system, and suppose that we have polytopic constraints: In particular, we require that $x_n \in \mathcal{X}$ and $u_n \in \mathcal{U}$ for all $n \geq 0$. A natural question to ask is: Does there exist a set $\Omega$ such that if $x_0 \in \Omega$, then the controller $u_n = Kx_n$ ensures that both constraints are satisfied. In mathematical terms, we would like this set $\Omega$ to achieve (a) constraint satisfaction

$$\Omega \subseteq \{x : x \in \mathcal{X}; Kx \in \mathcal{U}\},$$

and (b) control invariance

$$(A + BK)\Omega \subseteq \Omega.$$

It can be shown that if $0 \in \mathcal{X}$ and $0 \in \mathcal{U}$, then the set $\Omega$ can be represented by a polytope with a finite number of constraints. There is also an algorithm to compute this set:

**input** : $X = \{x : F_x x \leq h_x\}$ and $U = \{u : F_u u \leq h_u\}$
**input** : $A, B, K$

**output**: $\Omega$

set $t \leftarrow 0$;
set $k_1 \leftarrow \text{rows}(h_x)$;
set $k_u \leftarrow \text{rows}(h_u)$;
**repeat**
  **for** $j \leftarrow 1$ **to** $k_1$ **do**
    set $L_j^* \leftarrow \max\{(F_x)_j (A + BK)^{t+1} x - (h_x)_j : (A + BK)^k x \in \mathcal{X}, K(A + BK)^k x \in \mathcal{U}, \forall k = 0, \ldots, t\}$;
  **end**
  **for** $j \leftarrow 1$ **to** $k_2$ **do**
    set $M_j^* \leftarrow \max\{(F_u)_j K(A + BK)^{t+1} x - (h_u)_j : (A + BK)^k x \in \mathcal{X}, K(A + BK)^k x \in \mathcal{U}, \forall k = 0, \ldots, t\}$;
  **end**
  set $t \leftarrow t + 1$;
**until** $L_i^* \leq 0, \forall i = 1, \ldots, k_1$ *and* $M_i^* \leq 0, \forall i = 1, \ldots, k_2$;
set $t^* \leftarrow t - 1$;
set $\Omega = \{x : (A + BK)^k x \in \mathcal{X}, K(A + BK)^k x \in \mathcal{U}, \forall k = 0, \ldots, t^*\}$;

Note that the $\Omega$ returned by this algorithm is a polytope, and so we can rearrange terms to express this set as $\Omega = \{x : F_\omega x \leq h_\omega\}$.