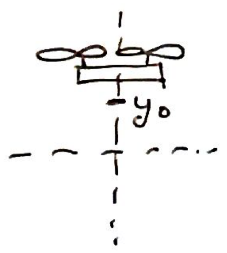
Lecture #5

Last Lecture: Continued discussion of control.

(1) Some important ideas/definitions:
- Fixed point: $\dot{\bar{x}} = f(\bar{x}_0, \bar{u}_o) = \bar{0}$.
- Global/Local asymptotic stability (for linear systems, local $\iff$ global)
- Stabilizability (saw that not all systems are stabilizable)
- Discussed proportional-derivative (PD) control. Saw how we could choose a controller of the form:

$$\bar{u}(\bar{x}) = \bar{u}_0 + \underset{\substack{\uparrow \\ \text{“Gain Matrix”}}}{K}(\bar{x} - \bar{x}_0)$$

to stabilize the planar quadrotor constrained to move in the y-direction.



(2) Ended lecture with a discussion of other considerations beyond stability (e.g.; different performance criteria)

Goal: Explore (2) in more detail and complete discussion of control.

The techniques we will develop today will only be directly applicable to *linear* systems. For nonlinear systems, we can linearize our system about a fixed point (e.g.; hover) and apply the techniques we develop to the linearized system. The hope is then that the controller we obtain for the linearized system is also a “good” controller for the original nonlinear system.

$$\underbrace{\dot{\bar{x}} = f(\bar{x}, \bar{u})} \approx \dot{\bar{x}} = A(\bar{x} - \bar{x}_0) + B(\bar{u} - \bar{u}_0) \rightarrow \underbrace{\text{Controller based on A, B}}$$

$$\begin{tikzcd} \dot{\bar{x}} = f(\bar{x}, \bar{u}) & \text{Controller based on A, B} \arrow[l, dashed, "\text{Apply on nonlinear system.}"] \end{tikzcd}$$

Linear Quadratic Regulator (LQR): We will develop a technique that *simultaneously* achieves two things:

(1) It stabilizes your (linear) system (assuming it is stabilizable)
(2) It optimizes a performance metric.

We start by assuming we have a linear system:

$$\dot{\bar{x}} = A(\underset{\substack{\uparrow \\ \in\mathbb{R}^n}}{\bar{x}} - \bar{x}_0) + B(\underset{\substack{\uparrow \\ \in\mathbb{R}^m}}{\bar{u}} - \bar{u}_0)$$

(Here, $\bar{x}_0$ is assumed to be a fixed point.)

To make things a bit more convenient, we do a change of coordinates: $\widetilde{x} \triangleq \bar{x}-\bar{x}_0$, and $\widetilde{u} \triangleq \bar{u}-\bar{u}_0$. Then $\dot{\widetilde{x}} = \dot{\bar{x}} = A(\bar{x}-\bar{x}_0) + B(\bar{u}-\bar{u}_0) \Rightarrow$

$$\dot{\widetilde{x}} = A\widetilde{x} + B\widetilde{u}$$

What would be a reasonable performance metric?

- Maybe we want to minimize derivations of the state from the desired state $\bar{x}_0$
- Maybe we want to minimize deviations from the nominal (i.e., reference) control input $\bar{u}_0$

Combining these, we could define:

$$J\big(\widetilde{x}(0)\big) \triangleq \int_0^\infty \Big[\underbrace{\widetilde{x}(t)^T\widetilde{x}(t)}_{\substack{\text{scalar}\\ =\|\widetilde{x}(t)\|^2}} + \underbrace{\widetilde{u}(t)^T\widetilde{u}(t)}_{\substack{\text{scalar}\\ =\|\widetilde{u}(t)\|^2}}\Big]dt \tag{1}$$

Here, $\widetilde{x}(t)$ is $\widetilde{x}$ at time t assuming you started from $\widetilde{x}(0)$.

In (1), we are penalizing all states (and control inputs) equally. Maybe some states/control inputs are more important than others.

$$\underbrace{J\big(\widetilde{x}(0)\big)}_{\text{“Cost function”}} \triangleq \int_0^\infty \Big[\widetilde{x}(t)^T \underset{\substack{\uparrow\\ n\times n}}{Q} \widetilde{x}(t) + \widetilde{u}(t)^T \underset{\substack{\uparrow\\ m\times m}}{R} \widetilde{u}(t)\Big]dt$$

Q and R are user defined matrices that specify relative importance of different states and control inputs. We will need to choose them to be *positive definite* and *symmetric*. (Recall: A symmetric matrix Q is positive definite if $\bar{x}^TQ\bar{x} > 0$, for all $\bar{x}$. This is equivalent to the eigenvalues of Q being positive.) In practice, we typically choose *diagonal* Q, R.

Our goal is to find some way to map states to control inputs in order to minimize the cost function. We won't go through the derivation (which is complicated), but will just talk about the solution.

It turns out that a *linear* controller:

$$\widetilde{u}(\widetilde{x}) = K^*\widetilde{x}$$

gives us the solution. This is pretty magical! To obtain this controller, there are two steps:

Step 1 Solve this *matrix equation* to find an $n \times n$ matrix S:

$$0 = Q - SBR^{-1}B^TS + SA + A^TS$$

Note that S is the only unknown in this equation. This equation is called the *Algebraic Riccatti Equation.* Denote the solution by S^*.

Step 2 Compute:

$$K^* = -R^{-1}B^TS^*$$

Our feedback controller is then:

$$\widetilde{u}(\widetilde{x}) = K^*\widetilde{x}$$
$$\Longrightarrow \bar{u}(\bar{x}) = \bar{u}_0 + K^*(\bar{x} - \bar{x}_0)$$

Now, you might be wondering how to go about doing Step 1... That equation seems nasty. Thankfully, people figured out how to solve the Riccati equation a while ago (using numerical techniques).

We can just use Python/Matlab:

$$[K^*, S^*] = lqr(A, B, Q, R).$$

This is super convenient! *Note*: Python/Matlab will also check if your system is stabilizable (and will complain if not). This is also convenient!

Warning: Matlab and Python use a slightly different convention than we do. In particular, they use controllers of the form $\widetilde{u} = -K^*\widetilde{x}$, instead of $\widetilde{u} = K^*\widetilde{x}$. So, the K^* they return is the negation of what we want to make things work.

Interpretation of $\boldsymbol{S^*}$: It turns out that the solution S^* has a nice interpretation. It gives us the optimal cost:

$$J^*\big(\widetilde{x}(0)\big) = \widetilde{x}(0)^T S^* \widetilde{x}(0)$$

where $J^*(\widetilde{x}(0))$ is the cost incurred by applying $K^*\widetilde{x}$ assuming we start from $\widetilde{x}(0)$.

Stability: So far, we haven't said anything about the stability (we just talked about minimizing the cost function). It turns out that the feedback controller stabilizes (global asymptotic stability) the system to the fixed point $\widetilde{x} = 0$ (i.e. $\bar{x} = \bar{x}_0$).

Thus, the system $\dot{\bar{x}} = A(\bar{x} - \bar{x}_0) + B(\bar{u} - \bar{u}_0)$ is stabilized to the fixed point $\bar{x}_0$, as desired. So, LQR does two things for us:

- It minimizes a (reasonable) performance metric (that tries to minimize deviations of $\bar{x}$ from $\bar{x}_0$ and $\bar{u}$ from $\bar{u}_0$).
- It results in a globally asymptotically stable closed-loop system:

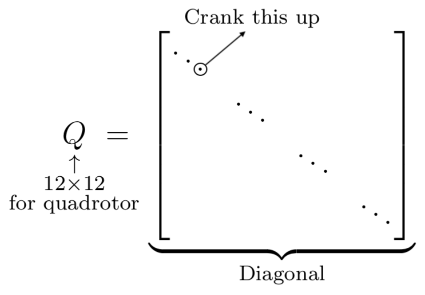
$$\dot{\widetilde{x}} = A\widetilde{x} + B\widetilde{u} = A\widetilde{x} + BK\widetilde{x} = (A + BK)\widetilde{x}.$$

How to choose $\boldsymbol{Q}$ and $\boldsymbol{R}$? In theory, any (positive definite) choices of Q and R will make the system stable. In reality, things are more complicated:

(1) Our system is actually nonlinear
(2) There are imperfections in our model
(3) We need to sense the state perfectly in order to apply the controller.

So, we can choose/tune Q and R to get the behavior we want. This is a bit of an art... you will get to do a bit of this tuning in the assignment.

Example: Suppose you see that the system is not stabilizing the z (height) well. What should you do?



Suppose you see that roll (ϕ) is not being stabilized well. Then you can increase the 4th element (on the diagonal) of Q. This places a higher penalty on roll.
(Recall: $\bar{x} = [x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, p, q, r]$.)

It takes some practice/tuning to get things right! But, usually, with some effort things work well! (See videos in slides.)