# Controllability

If an input to a system can be found that takes every state from an initial state to a desired other state, the system is controllable

Obvious much?

But REALLY important! No amount of gain adjustment can help us if those gains aren't actually influencing the things we want to influence!

A really simple example:

$$\dot{x} = \begin{bmatrix} -1 & 0 \\ 0 & 2 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$\dot{x}_1 = -x_1$

the state variable $x_1$ is not controlled by $u$.

$\dot{x}_2 = 2x_2 + u$

there is no actuator to influence $x_1$

$$A - BK = \begin{bmatrix} -1 & 0 \\ 0 & 2 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \end{bmatrix}$$

$$= \begin{bmatrix} -1 & 0 \\ 0 & 2-k_2 \end{bmatrix} \Leftarrow \text{no } k_1!$$

Or, practical examples:

- An aircraft w/ no rudder can't control yaw
- A satellite w/ no thrusters can't control orientation
- A car w/ no pedals can't control speed
- A quadcopter w/ no propellers can't control anything...

# Observability

A system is observable if all relevant states can be known from the system outputs

The reason for this is straightforward → if we want to get to a desired value we need to be able to measure the current value

Practically, this is achieved by:

1) Directly sensing the current state (gyro, IMU, etc)

2) Estimating the current state from other things you can observe

Which approach is used is driven by practical concerns of cost, weight, accuracy of available sensors and difficulty in state estimations

→ Consider also that accuracy & speed of estimations will hugely affect the result

Fun thought experiment: Think about a self-piloting quadcopter. What is needed to make it controllable & observable?

**Controllable**

- ESCs
- Computer

could you do it from just this? →

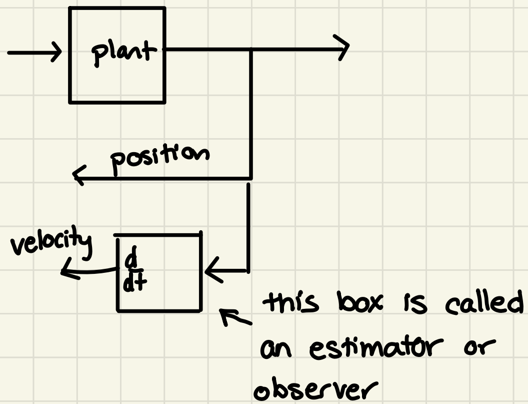
**Observable**

- GPS
- IMUs / accelerometers
- altimeter?

Now it's worth noting that there are two ways we can observe a state

1) Measure it
2) Estimate it from our measurements

Let's think about this a little more




It's worth noting that designing the estimator is an important part of control system design

→ sensitivity to error / noise / measurement bias
→ speed of estimation

But adding more sensors can be \$, heavy, complex

So there is a tradeoff in the system design where you need to figure out which states to measure and which to estimate

Often, experienced engineers can simply infer controllability and observability from deep knowledge of their system. But these can also be calculated.

Controllability: An $n^{th}$ order system with state equation

$$\dot{x} = Ax + Bu$$

is completely controllable if the matrix

controllability matrix → $C_M = [B \quad AB \quad A^2B \cdots A^{n-1}B]$ is of rank $n$

↳ number of linearly independant rows or coloums

We can very quickly find this in Matlab

CM = ctrb(A, B) to get $C_M$

Rank = rank(CM) to get its rank

Quick Ex

$$\dot{x} = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 2 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$

$$CM = [B \quad AB \quad A^2B] = \begin{bmatrix} 0 & 1 & -2 \\ 1 & -1 & 1 \\ 1 & -2 & 4 \end{bmatrix}$$

$\text{rank}(CM) = 3 \Rightarrow$ system is controllable!

Observability: An $n^{th}$ order plant with state eqns

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

is completely observable if the matrix

observability matrix $\rightarrow$ $OM = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$

has rank n

we can quickly find in Matlab via

```
OM = obsv(A,C)
Rank = rank(OM)
```

A quick example

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -4 & -3 & -2 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 5 & 1 \end{bmatrix} x$$

$$OM = \begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix} = \begin{bmatrix} 0 & 5 & 1 \\ -4 & -3 & 3 \\ -12 & -13 & -9 \end{bmatrix}$$

rank(OM) = 3

system is observable

Okay, so as long as I have a SS system that is observable and controllable, I can just do this pole placement thing and BAM! perfect control?

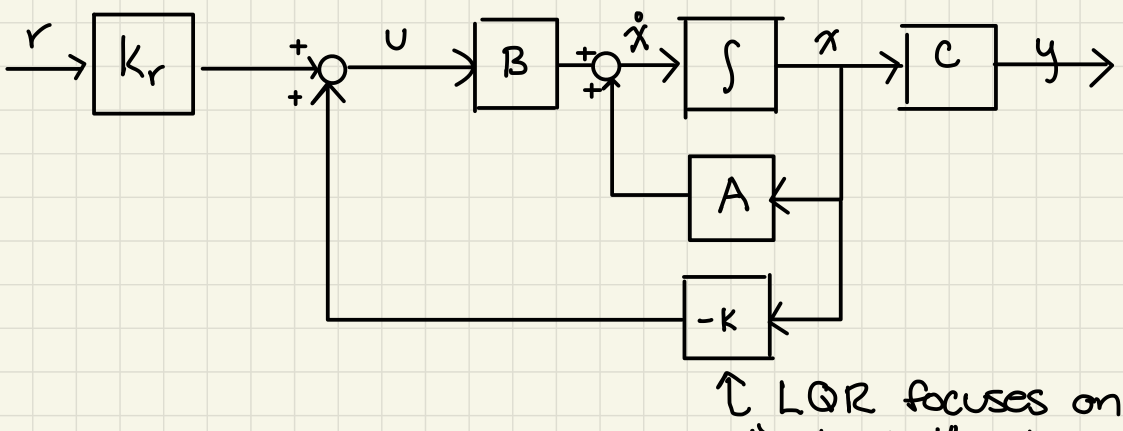
Well... yes... but also no...

The question is - where do you want to put the poles? We really only know how to select pole locations for systems of 2nd order or that can be approximately 2nd order. But we have seen that not all systems work like this.

Wouldn't it be EVEN BETTER if we could just decide how important our performance criteria are and get the controller to pick the Ks for us to get the poles in a good location?

Linear Quadratic Regular "LQR" is one optimal control method that is widely used

So let's go back to our block diagram we used in pole placement. We'll use the same one for LQR, but the method to find the K values will change.




↑ LQR focuses on "optimal K" values

First-what does optimal mean?

Let's imagine that COVID is over and spring break is back! We're going to plan a trip! Where should we go?

| alternatives | "performance" | "cost" |
|---|---|---|
| ↓ | ↓ | ↓ |
| **Destination** | **Awesomeness** (1-10) | **100s of $ for trip** |
| GA mountains | 2 | $2 |
| FL beach | 5 | $5 |
| Bahamas | 8 | $15 |
| European Getaway | 10 | $30 |

Where will you go? How will you decide?

$$J = Q \cdot \text{Awesomeness} + R \cdot \$ \quad \Leftarrow \text{minimize}$$

What if $Q \gg R$? $R \gg Q$? $R = Q$?

Now what does this have to do with controls? Well, let's think about what this looks like for a control system.

The $x$ state vector tells performance

The $u$ vector tells actuation effort (cost)

The objective function is the weighted sum of performance and cost

0 for us

$$J = \underbrace{\int_0^\infty (\underbrace{x^T Q x}_{\text{performance}} + \underbrace{u^T R u}_{\text{cost}})\,dt}_{} + \cancel{\int_0^\infty 2x^T N u\,dt}$$

deals w/ crossterms b/w $u$ and $x$. often $\emptyset$

So let's talk about why our objective function looks this way and how it works.

First, some observations:

1) This is quadratic ($x$ and $u$ are squared)
2) For the math to work, $Q$ and $R$ are square matricies with dimensions equal to the length of $x$ and $u$ respectively.
3) $Q$ & $R$ must be positive definate (+ eigenvales)
→ convex → can use gradient methods

$\underset{1\times n}{x^T}\ \underset{n\times n}{Q}\ \underset{n\times 1}{x}$ becomes a positive scalar (or zero)

same for $\underset{1\times m}{u^T}\ \underset{m\times m}{R}\ \underset{m\times 1}{u}$

So the cost function will always give a positive number, and the goal is to minimize this

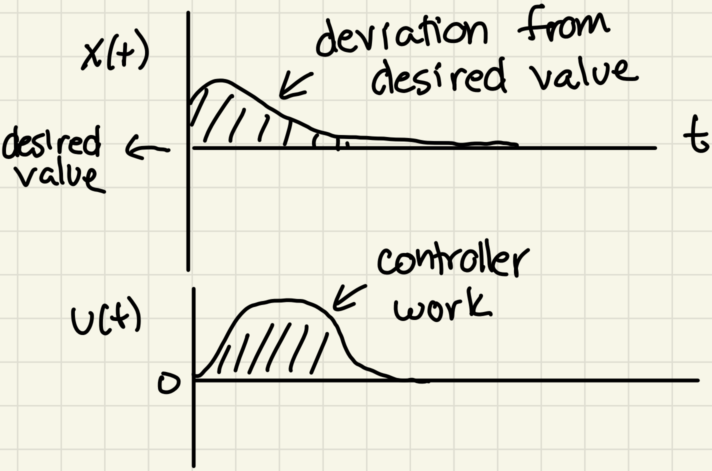
We have an optimization problem!

$$\boxed{\begin{array}{ll} \text{minimize } J & u \in \mathbb{R}^m \\ \text{such that} & \dot{x} = Ax + Bu \end{array}}$$

Basically, I want to find the control setup that gets the best performance for the least cost, where "best" is determined by the values in $Q$ & $R$

$\int x^T Q x$ measures deviation from desired value

$\int u^T R u$ measures how much control effort is needed to get there




* as a side note the "squared" part ensures all deviation counts as ⊕ to the $\int$