Lecture 5: Acrobats, Cart-poles and Quadrotors I

Recap

Dynamic Programming view of optimal Control

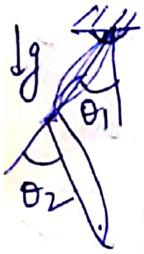
- Tabular setting (everything discrete $\Rightarrow$ optimization on a discrete graph)
- LQR (dynamics are linear) $\longrightarrow$ restricted to linear dynamics and a lot of things we care about is not linear dynamics.

· Value function / Cost-to-go function

$$J^*(s) = \min_a \left[l(s,a) + J^*(f(s,a)) \right]$$

(Continuous setting) $\quad 0 = \min_u \left[l(x,u) + \frac{\partial J^*}{\partial x} f(x,u) \right]$

Acrobot

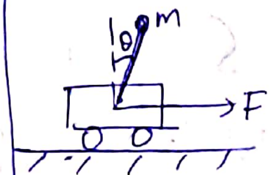


$q = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$

$u = \left[\tau_{elbow} \right] \in [u_{min}, u_{max}]$

torque at the elbow

Cart-Pole



$X = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \Rightarrow$ Full state 4×1 vector

$2\times1 \quad q$

$2\times1 \quad \dot{q}$

$q = \begin{bmatrix} X_{cart} \\ \theta_{pendulum} \end{bmatrix}$

$u = \left[F_{cart} \right] \in [u_{min}, u_{max}]$

State Constraints

$X \in [X_{min}, X_{max}]$

for cart-pole

$B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

Manipulator eqn

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} = \tau_g(q) + Bu$$

for acrobot,

$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Scanned with OKEN Scanner

LQR for non-linear Systems

(linear approx. of a non-linear dynamics)

$$\dot{x} = f(x, u) \Rightarrow \dot{x} = Ax + Bu$$

Linearize the dynamics about some nominal (x_0, u_0)

(Approximate $\dot{x}$ with Taylor approximation)

$$\dot{x} \approx \underbrace{f(x_0, u_0)}_{\dot{x}_0} + \frac{\partial f}{\partial x}\Big|_{\substack{x=x_0 \\ u=u_0}}(x - x_0) + \frac{\partial f}{\partial u}\Big|_{\substack{x=x_0 \\ u=u_0}}(u - u_0) + \ldots$$

higher order terms

Change of variables,

$$\bar{x} = x - x_0 \quad , \bar{u} = u - u_0$$

$$\dot{\bar{x}} = \dot{x} - \dot{x}_0$$ (Simple version is when $\dot{x}_0 = 0$)

if $\dot{x}_0 = 0 \Rightarrow f(x_0, u_0) = 0$

(this means we linearize around a fixed point)

$$\dot{\bar{x}} = \underbrace{\frac{\partial f}{\partial x}}_{\text{matrix } A} \bar{x} + \underbrace{\frac{\partial f}{\partial u}}_{\text{matrix } B} \bar{u}$$

Linearization for the Pendulum

$$m l^2 \ddot{\theta} + b\dot{\theta} + mgl\sin\theta = \tau$$

$$x = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \quad \dot{x} = \begin{bmatrix} \dot{\theta} \\ \frac{1}{ml^2}(u - b\dot{\theta} - mgl\sin\theta) \end{bmatrix}$$

linearize about upright fixed point $\Rightarrow x_0 = \begin{bmatrix} \pi \\ 0 \end{bmatrix}, u_0 = 0$

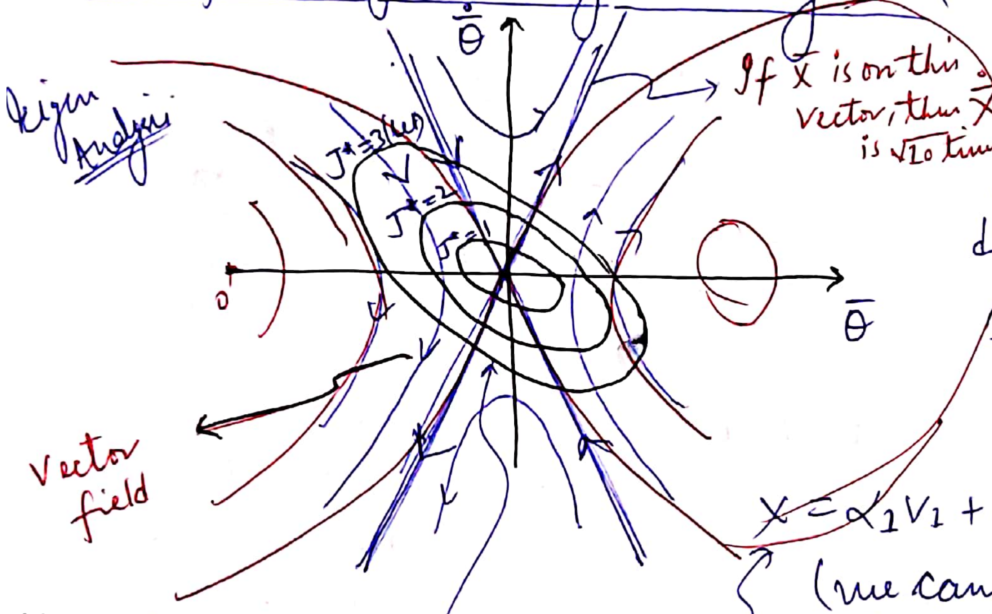
$$\frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 \\ -g/l\cos\theta & -b/ml^2 \end{bmatrix}\Bigg|_{\theta=\pi} = \begin{bmatrix} 0 & 1 \\ g/l & -b/ml^2 \end{bmatrix}$$

$$\frac{\partial f}{\partial u} = \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix}_{2\times 1}$$

$m = l = 1$, $b = 0$, $g = 10$ (Let's assume)

$$\dot{X} = A\bar{X} + B\bar{u} \qquad A = \begin{bmatrix} 0 & 1 \\ 10 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Phase portrait for above dynamical system (we can use eigen value to do this)



$Av = \lambda v$

$\det(A - \lambda I) = 0$

$\lambda_1 = \sqrt{10}$ $\quad V_1 = \begin{bmatrix} 1 \\ \sqrt{10} \end{bmatrix}$

$\lambda_2 = -\sqrt{10}$ $\quad V_2 = \begin{bmatrix} -1 \\ \sqrt{10} \end{bmatrix}$

$X = \alpha_1 V_1 + \alpha_2 V_2$

(we can write any x as linear combination of eigen vectors as they span the entire space)

$\dot{X} = \lambda_1 \alpha_1 V_1 + \lambda \alpha_2 V_2$

Non-linear overlay (in red)

cost-to-go

★ when the cost-to-go (J^*) is less than some number then the linear controller is going to work for some non-linear system.

LQR Appreciation

$\bar{u} = -K\bar{x}$

$A, B, \underline{Q}, \underline{R} \Rightarrow \underline{K}$] If it's not hard to find Q & R. But what's magical here is any choice of Q & R gives us a stable 'K' for the linear system.

$\dot{X} = Ax + Bu$, $u = -Kx$

$\dot{X} = (A - BK)x \Rightarrow$ if we choose K arbitrarily (here $\cancel{\text{muck}}$ mucky around in K) it would be result in instability. So, it's better to muck around in Q & R to get a stabilizing K

★ Searching in K directly is a hard ety

Instead of searching in the control parameters, we will search in the cost function.

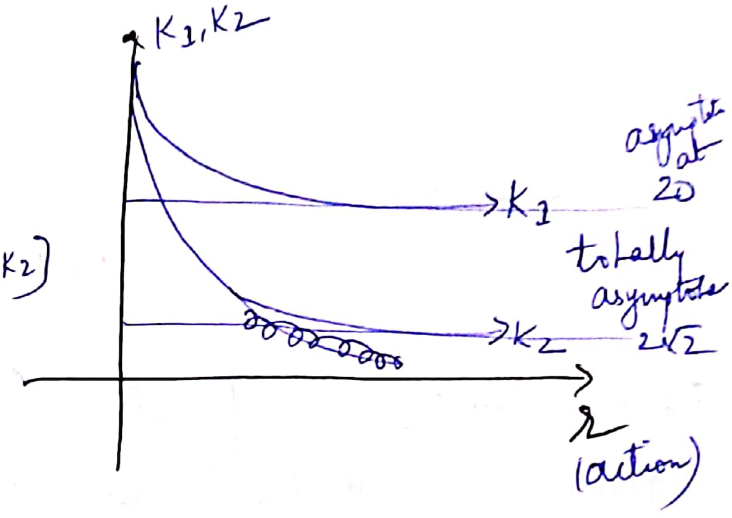
Any choice of Q & R gives us a stable 'K' for the linear system)

→ Better to write cost functions instead of control functions.

LQR for pendulum

$A = \begin{bmatrix} 0 & 1 \\ 10 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$Q = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix} \quad R = \underbrace{[r]}_{\text{scalar}}, \; K = [K_1, K_2]$



If we double Q, then more penalty of trying to get them fast.

→ If we leave Q & double R, then it says our actions are expensive. So, get them more slowly.

[K1 K2]

for what value of K, the system is stable?

Ans – To be stable, we need

$K_1 > 10$

$K_2 > 0$

(that what we get by looking at the eigen values of closed loop matrix.)

Note

The drawback of LQR is that there are stabilizing controllers that we cannot find by just changing Q & R.

$\dot{x} = (A - BK)x$

$(A, B, Q, R \Rightarrow K)$ This parametrizes a class of stabilizing controllers but it does not stabilize the entire family of stabilizing controllers.

* If we search K directly $[\dot{x} = (A-BK)x]$ we might blow up sometimes but there are more controllers available for us to search for.

Search in Q & R (in $A, B, Q, R \Rightarrow K$) is safe but ~~might not~~ there might be a controller that we cannot find that way)

Scanned with OKEN Scanner

Linearizing the manipulator eqn

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} = \tau_g(q) + B_M u \quad (M = \text{manipulator})$$

$$x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \qquad \dot{x} = \begin{bmatrix} \dot{q} \\ M^{-1}(q)\left[Bu + \tau_g(q) - C(q,\dot{q})\dot{q}\right] \end{bmatrix}$$

If we run the manipulator equations through the linearization around a fixed point, then

Automatic differentiation Pytorch

$$A = \begin{bmatrix} 0 & I \\ \left(-M^{-1}\frac{\partial \tau_g}{\partial q}\right) & \left(-M^{-1}C\right) \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ M^{-1}B_M \end{bmatrix}$$

Does it still simply nicely in the time varying case?
→ It doesn't simplify as much as this.

Controllability

Defn: A non-linear system $\dot{x} = f(x,u)$ is controllable if $\forall\ x_0, x_f$ we can find $u(t)\ \forall\ t \in [t_0, t_f]$ such that $x(t_0) = x_0,\ x(t_f) = x_f$.

Achieving in a finite time. Time can be arbitrarily large but it has to be finite

So, A system is controllable if for any initial condition and any final condition in state we can find a finite set of trajectory of u that will get us from one state to the other.

It's tempting to say that under actuation limits your controllability. That's not the case.

~~Unactuated~~ $\boxed{\text{Underactuated} \not\Rightarrow \text{Uncontrollable}}$
(doesnot imply)

Stabilizability is a related concept.