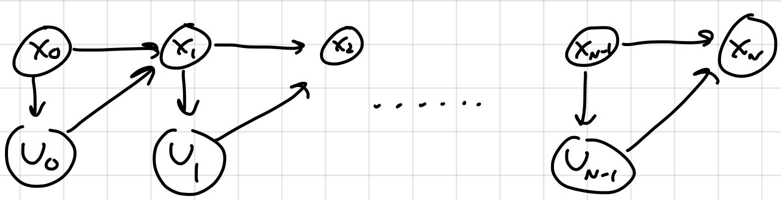
Recitation 4 (2/10/23)

- LQR
  - terminology
  - coordinate change
  - extension to nonlinear

$$x_{k+1} = f(x_k, u_k)$$



Our discrete time dynamics is a Markov Decision Process (MDP). Basically we have states and actions, and we need a state and action to get to the next state. Because of this, our finite control problems deal with N x's, and (N-1) u's.

LQR road map

discrete only

"finite horizon LQR"

"TVLQR"

$$\min_{\substack{x_{1:N} \\ u_{1:N-1}}} \sum_{i=1}^{N-1} \underbrace{\left(\tfrac{1}{2} x_i^T Q x_i + \tfrac{1}{2} u_i^T R u_i\right)}_{\text{stage cost}} + \underbrace{\tfrac{1}{2} x_N^T Q_f x_N}_{\text{terminal cost}}$$

$$\text{s.t.} \quad x_0 = x_0$$

$$x_{k+1} = A_k x_k + B_k u_k$$

- equality constrained QP ✓

Do not confuse these with iLQR, which we haven't gotten to yet in this class.

~~iLQR~~

$\text{eig}(Q) \geq 0, \; Q \succeq 0$

$\text{eig}(R) > 0, \; R \succ 0$

Q must be PSD, R must be PD

Finite horizon LQR is an equality constrained quadratic program, which is easy to solve (we just have to solve one big linear system). We can also solve it with Ricatti for a time varying feedback policy.

"infinite horizon LQR"

$$\min_{\substack{x_{1:\infty} \\ u_{1:\infty-1}}} \sum_{i=1}^{\infty-1} \underbrace{\left(\tfrac{1}{2} x_i^T Q x_i + \tfrac{1}{2} u_i^T R u_i\right)}_{\text{stage cost}}$$

$$\text{s.t.} \quad x_0 = x_0$$

$$x_{k+1} = A x_k + B u_k$$

infinite horizon LQR has no terminal cost, only stage cost (since it goes forever). We also can't deal with time varying A and B directly (since it goes forever).

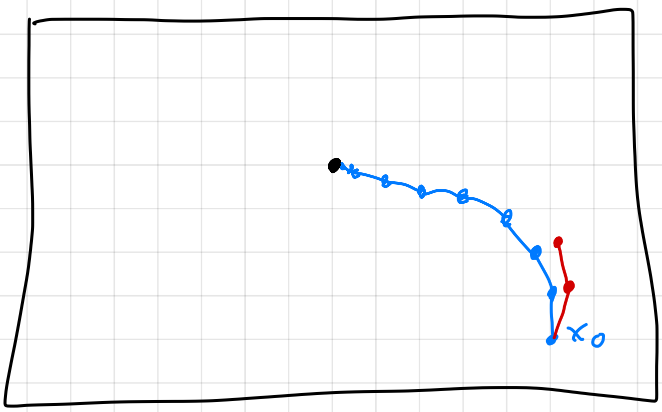
infinite horizon LQR we can't form a QP anymore (since it goes forever), but we can use Ricatti to get a feedback policy.

Policies: $\pi(x) = u$

TVLQR: ① $\pi(x) = \text{solve\_QP}(A_{1:N-1}, B_{1:N-1}, Q, R, Q_f)$

② $\pi(x) = -K_k x$ from riccati

In the case of TVLQR, we can either re-solve the QP every time the our state deviates from the planned trajectory, or we can simply use the feedback policy for the EXACT same controls. Obviously we'd rather use the feedback policy.



The QP will solve for the optimal trajectory (shown in blue). Unfortunately, due to model uncertainty, noise in the environment, etc, we are going to deviate from this planned trajectory. This would cause us to have to resolve the QP, but the feedback policy is globally optimal, so it's fine wherever we go.

IHLQR: $u = -Kx$

$K = \text{dLQR}(A, B, Q, R)$

$$u = -K(x - x_{goal})$$

We can also drive the system to a x_goal instead of just 0, more on this on the next page.

$$x_{k+1} = A x_k + B u_k$$

$$x_{goal} = A x_{goal} + B u_{goal}$$

Imagine we have a x_goal that we want to drive the system to, and there is a x_goal and u_goal that keeps us at x_goal

$$\tilde{x} = x - x_g \quad , \quad \tilde{u} = u - u_g$$

Let's define some new coordinates, xtilde and utilde, and write out the dynamics.

$$\tilde{x}_{k+1} = x_{k+1} - x_g = A x_k + B u_k - (A x_g + B u_g)$$

$$\tilde{x}_{k+1} = A(x_k - x_g) + B(u - u_g)$$

$$\boxed{\tilde{x}_{k+1} = A \tilde{x}_k + B \tilde{u}_k}$$

We just get the same linear system but in our new coordinates.

$$K = dLQR(A, B, Q, R)$$

We just solve for our infinite horizon LQR gain as usual (no changes based on what x_goal and u_goal were).

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

Which gives us this policy, driving us to x_goal

$$u - u_g = -K(x - x_g)$$

$$\boxed{u = u_g - K(x - x_g)}$$

$$K_{1:N-1} = TVLQR(A_{1:N-1}, B_{1:N-1}, Q, R, Q_f)$$

TVLQR gives us a list of K's, where we have a new feedback gain for each timestep.

$$u_i = -K_i x_i$$

$$x_{k+1} = A(t) x_k + B(t) u_k$$

This was just a restatement that A and B must only vary with time.

$$x_g = A x_g + B u_g$$

$$x = x_g + \Delta x \quad , \quad u = u_g + \Delta u$$

We can do equivalently define our coordinate system with these delta's. Following the same process as before, we get a standard linear system, and feedback policy.

$$x_{k+1} = A x_k + B u_k$$

$$x_{k+1} = x_g + \Delta x_{k+1} = A(x_g + \Delta x) + B(u_g + \Delta u)$$

$$\cancel{x_g} + \Delta x_{k+1} = A \cancel{x_g + B u_g} + A \Delta x + B \Delta u$$

$$\boxed{\Delta x_{k+1} = A \Delta x_k + B \Delta u_k}$$

Same linear system, same dLQR call, same feedback gain.

MAIN TAKEAWAY:
Whether we are driving the system to zero, or driving it to an arbitrary xg ug, we have the same feedback gain K. The incorporation of xg and ug shows up in the u = -Kx statement.

$$K = dLQR(A, B, Q, R)$$

$$\Delta u = -K \Delta x$$

Feedback policy is in terms of deltas. Substitute in our du = u - ug and dx = x - xg to get u.

extension to nonlinear:

$$x_{k+1} = f(x_k, u_k)$$

$$\bar{x}, \bar{u} \text{ is eq} , \quad \bar{x} = f(\bar{x}, \bar{u})$$

Let's take a linear system, and linearize it about x bar u bar.

$$x_{k+1} = f(x_k, u_k)$$

$$\boxed{① \quad x_{k+1} \approx f(\bar{x}, \bar{u}) + \left[\frac{\partial f}{\partial x}\Big|_{\bar{x},\bar{u}} \right](x - \bar{x}) + \left[\frac{\partial f}{\partial u}\Big|_{\bar{x},\bar{u}} \right](u - \bar{u})}$$

This is the first order Taylor series of the discrete nonlinear dynamics.

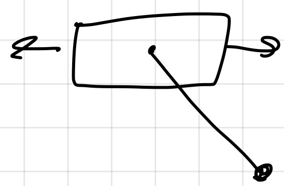
$$x = \bar{x} + \Delta x, \quad u = \bar{u} + \Delta u$$

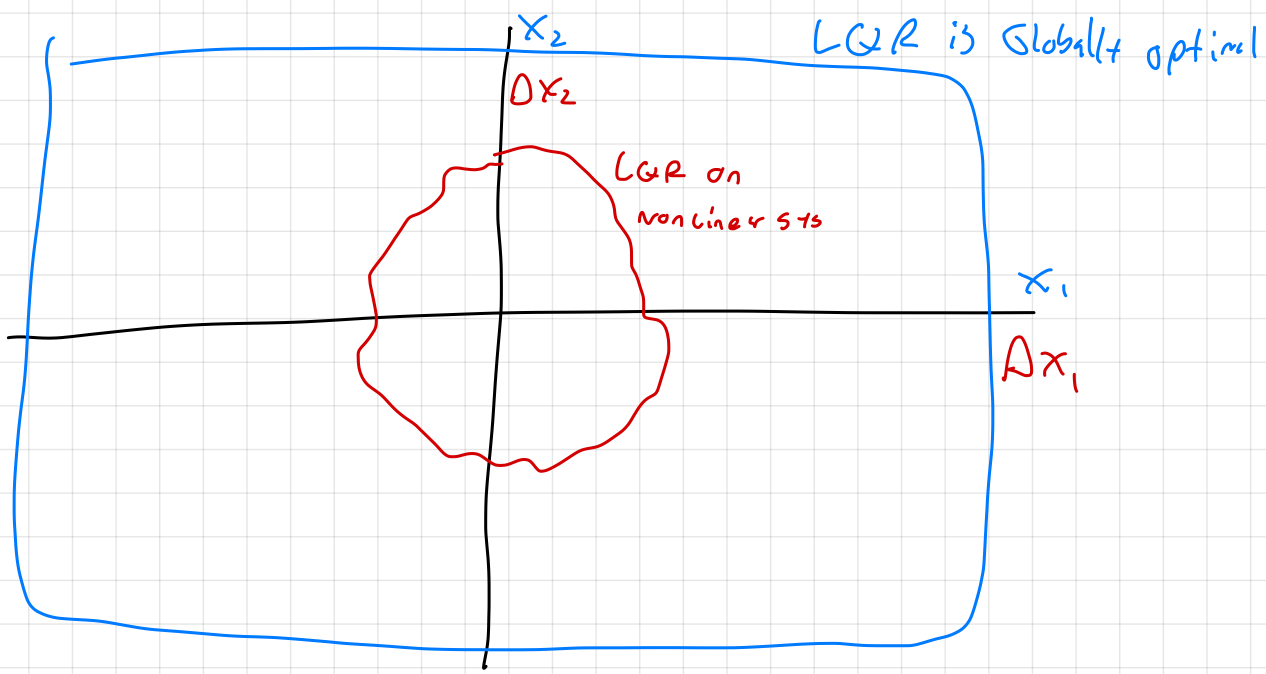
$$x_{k+1} \approx f(\bar{x}, \bar{u}) + \overbrace{\left[\frac{\partial f}{\partial x}\Big|_{\bar{x},\bar{u}}\right]}^{\bar{A}}(x - \bar{x}) + \overbrace{\left[\frac{\partial f}{\partial u}\Big|_{\bar{x},\bar{u}}\right]}^{\bar{B}}(u - \bar{u})$$

$$\cancel{\bar{x}} + \Delta x_{k+1} \approx \cancel{f(\bar{x}, \bar{u})} + A \Delta x + B \Delta u$$

Alternatively, we can write this in terms of deltas and get an approximate linear system that we can call dLQR on.

$$(2) \quad \boxed{\Delta x_{k+1} = \bar{A} \Delta x_k + \bar{B} \Delta u_k}$$





LQR on a linear system is globally optimal since our linear dynamics are exact.

LQR for a nonlinear system is only really going to work in a region of the state space close to where the linearized model was formed.