Linear Quadratic Gaussian (LQG) Control

NDSU ECE 463/663

Lecture #25

Inst: Jake Glower

Please visit Bison Academy for corresponding lecture notes, homework sets, and solutions

Pole-Placement Flashback:

- Kx = ppl(A, B, Poles);
- Places the poles *anywhere*

Where *should* the poles be placed?

What gains are best?

Linear Quadratic Gaussian Control (LQG):

- Define a cost function
- Find Kx to minimize that cost function

LQG is just another way to find Kx:

- Kx = lqr(A, B, Q, R);

LQG Control Solution:

Assume you have a linear system with an arbitrary initial condition

$$sX = AX + BU$$

$$Y = CX$$

Find Kx to minimize

$$J = \int_0^\infty (X^T QX + U^T RU)dt$$

$$U = -K_x X$$

The solution is

$$K_x = R^{-1} B^T P$$

where P is the solution to the Ricatti equation

$$0 = -A^T P - PA - Q + PBR^{-1} B^T P$$

Comments:

- Essentially, the cost function is the matrix form of

$$J = \int_0^\infty \left(\Sigma\, q_i x_i^2 + \Sigma\, r_i u_i^2 \right)$$

- This cost function has a solution (a big plus)
- The resulting gains are constants (another big plus)
- You could use other cost functions
  - but that would make the solution *much* harder to obtain.

This is termed *optimal control*

- *It's optimal for an arbitary cost function.*
- *Any* stabilizing control law is optimal for some Q and R
- That sort of makes the word *optimal* meaningless.

LQR is a tool similar to pole placement to find feedback gains.

Example: Heat Equation

Find the *optimal* feedback gains for the heat eqation with

$$J = \int_0^\infty (y^2 + u^2)dt$$

$$J = \int_0^\infty (10^4 y^2 + u^2)dt$$

$$J = \int_0^\infty (y^2 + 10^4 u^2)dt$$

Matlab Solution: First, input the system (A, B)

```
A = [-2,1,0,0;1,-2,1,0;0,1,-2,1;0,0,1,-1]

 - 2.     1.     0.     0.
   1.   - 2.     1.     0.
   0.     1.   - 2.     1.
   0.     0.     1.   - 1.
```

```
B = [1;0;0;0]

     1.
     0.
     0.
     0.
```

Define the weighting matrices (Q, R)

```
C = [0,0,0,1]

     0.     0.     0.     1.

Q = C'*C

     0.     0.     0.     0.
     0.     0.     0.     0.
     0.     0.     0.     0.
     0.     0.     0.     1.

R = 1;
```

Solve the Ricatti equation to find the full-state feedback gains:

```
Kx = lqr(A, B, Q, R)

     0.0426938    0.0862990    0.1279791    0.1572416
```

The "optimal" location of the closed-loop poles are:

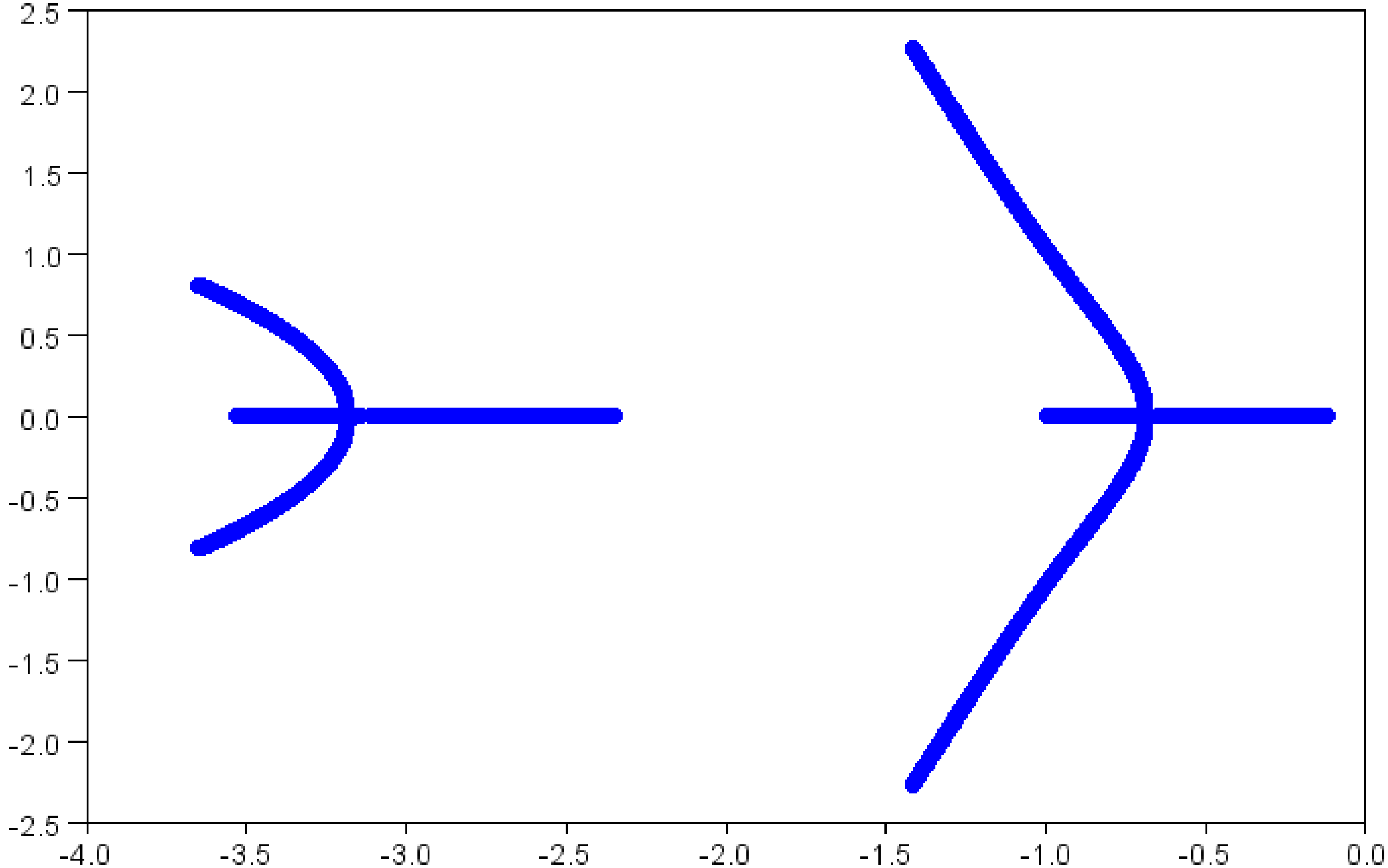
```
eig(A-B*Kx)

   -3.5322276
   -2.3461035
   -1.0089118
   -0.0700632
```

Repeating for weights of $\{10^4, 10^2, 1, 10^{-2}, 10^{-4}\}$ for Q:

Q	"Optimal" Feedback Gain: Kx			
10^{-4}	0.0000053	0.0000105	0.0000154	0.0000188
10^{-2}	0.0005257	0.0010516	0.0015389	0.0018713
1	0.0426938	0.0862990	0.1279791	0.1572416
10^{2}	0.6842631	1.6026342	2.8230412	3.9399371
10^{4}	3.128746	11.152018	29.584146	55.140089

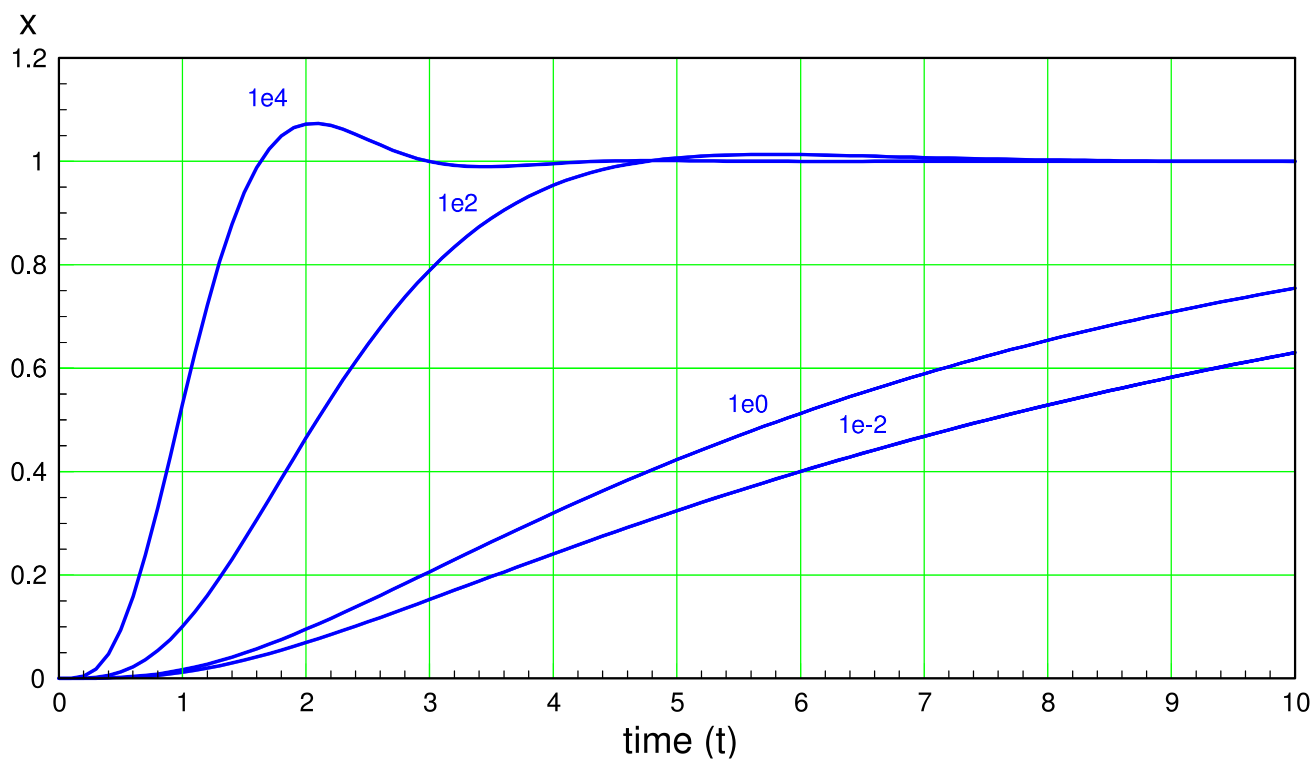
The corresponding location of the closed-loop poles are:

Q	"Optimal" Location of Closed-Loop Poles
10^{-4}	−3.53, −2.34, −0.99, −0.12
10^{-2}	−3.53, −2.34, −0.99, −0.12
1	−3.53 −2.34, −0.99, −0.17
10^{2}	−3.51, −2.45, −0.85 + j0.65
10^{4}	−3.64 + j0.81, −1.41 + j2.27



Location of "Optimal" Closed Loop Poles for R=1, $10^{-4} < Q < 10^{+4}$

```
Kx = lqr(A, B, Q, R);
DC = -C*inv(A-B*Kx)*B;
Kr = 1/DC;
G = ss(A-B*Kx,B*Kr,C,0);
y = step(G,t);
plot(t,y);
```



Tuning the Step Response:

Adjust Q and R to tune the response:

Faster System: Increase the weight on y = CX

$$Q = C^T C$$

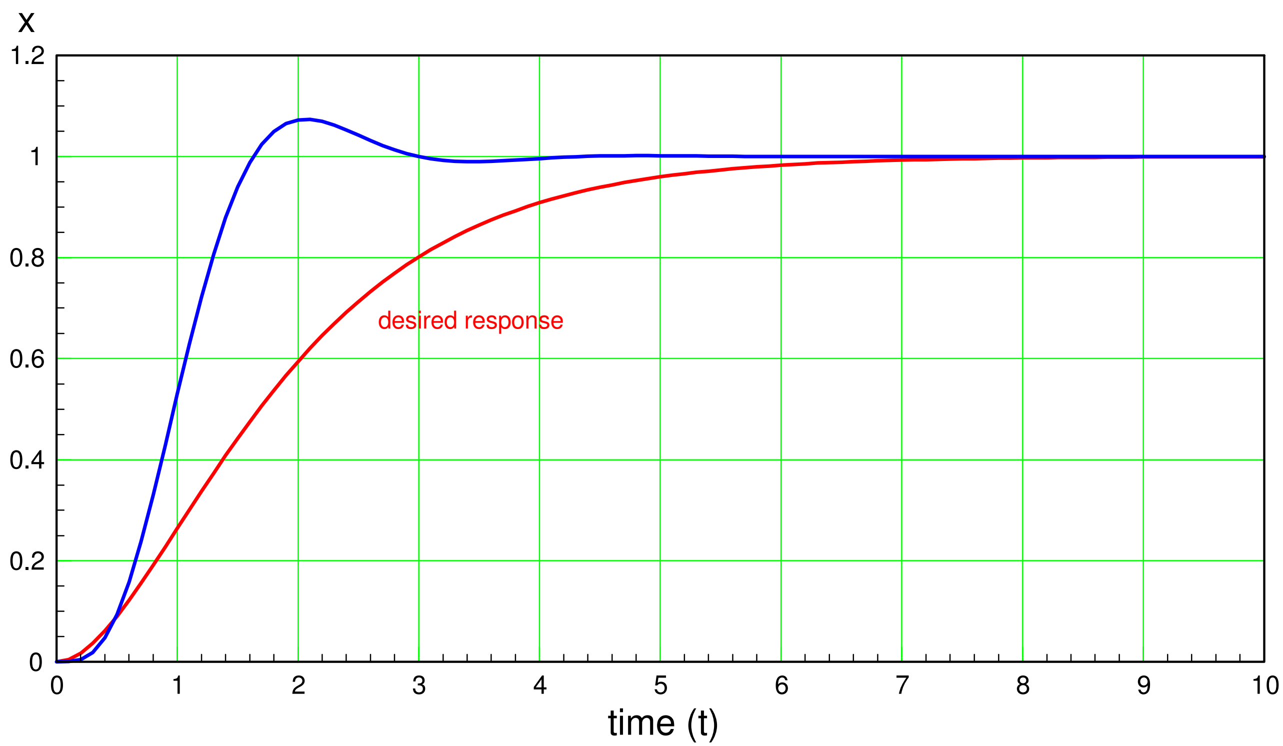
Slow Down or Less Oscillation: Weight y' = (CA)X

$$Q = (CA)^T CA$$

$$Q = \alpha C^T C + \beta (CA)^T CA$$

Example: Design a feedback controller so that the 4th-order heat equation has

- No overshoot for a step input, and
- A 2% settling time of 4 seconds



Step response for $Q = 10^4 C^T C$ (blue) as well as the desired response

Adjust the weightings on y and y'

$$Q = 10^4 \cdot C^T C + 3 \cdot 10^4 (CA)^T CA$$