Topic #13

16.30/31 Feedback Control Systems

State-Space Systems

- Full-state Feedback Control Performance/Robustness

- Reading: FPE 7.9.1

LQ Servo Introduction

- Can use scaling $\overline{N}$ can achieve zero steady state error, but the approach is **sensitive** to accurate knowledge of all plant parameters

- Can modify LQ formulation to ensure that zero steady state error is **robustly** achieved in response to constant reference commands.
  - Done by augmenting integrators to the system output and then including a penalty on the integrated output in LQ cost.

- **Approach:** If the relevant system output is $\mathbf{y}(t) = C_y\mathbf{x}(t)$ and reference $\mathbf{r}(t)$, add extra states $\mathbf{x}_I(t)$, where

$$\dot{\mathbf{x}}_I(t) = \mathbf{e}(t) = \mathbf{r}(t) - \mathbf{y}(t)$$

  - Then penalize both $\mathbf{x}(t)$ and $\mathbf{x}_I(t)$ in the cost

- If state of the original system is $\mathbf{x}(t)$, then the dynamics are modified to be

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{\mathbf{x}}_I(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C_y & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{x}_I(t) \end{bmatrix} + \begin{bmatrix} B_u \\ 0 \end{bmatrix} \mathbf{u}(t) + \begin{bmatrix} 0 \\ I \end{bmatrix} \mathbf{r}(t)$$

  and define $\overline{\mathbf{x}}(t) = \begin{bmatrix} \mathbf{x}^T(t) & \mathbf{x}_I{}^T(t) \end{bmatrix}^T$

- The optimal feedback for the cost

$$J = \int_0^\infty \left[\overline{\mathbf{x}}^T R_{\mathrm{xx}} \overline{\mathbf{x}} + \mathbf{u}^T R_{\mathrm{uu}} \mathbf{u}\right] dt$$

  is full-state feedback (found using LQR approach in Topic 12)

$$\mathbf{u}(t) = -\begin{bmatrix} K & K_I \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_I \end{bmatrix} = -\overline{K}\overline{\mathbf{x}}(t)$$

- Once we have used LQR to design the control gains K and K_I, we typically implement the controller using the architecture:

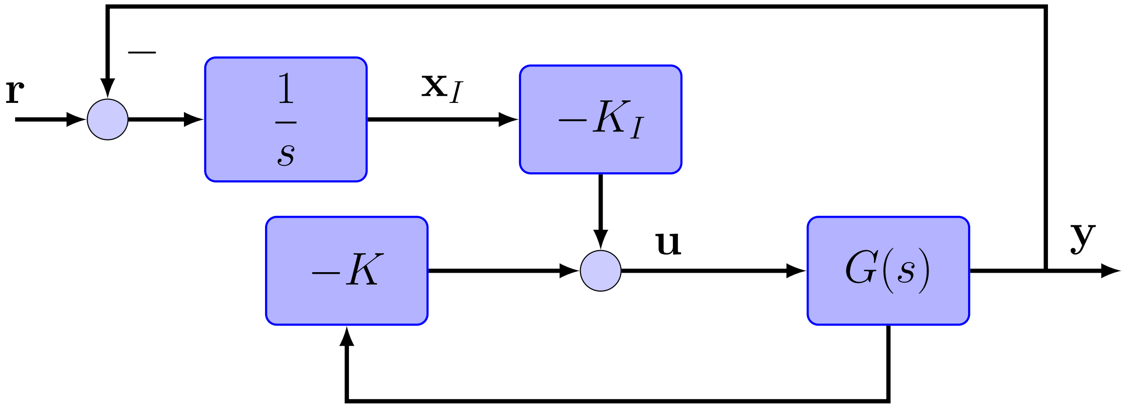

Fig. 1: Typical implementation of the LQ servo

- Example – compare the integrator and $\overline{N}$ (see 11-??) approaches when the model of the system dynamics is wrong.

  - **Case 1:** nominal, so the C used to design the K's and N's is the same used in the simulations

  - **Case 2:** off-nominal, so the C used to design the K's and N's differs from the one used in the simulations

- Obvious that the $\overline{N}$ approach inaccurate at steady state, but the integrator approaches still do OK $\Rightarrow$ robustness to modeling errors.

Code: LQR Servo

```
% LQ servo
% Fall 2010
% Jonathan How, MIT
close all;clear all;j=sqrt(-1);

G=tf([1 2],conv([1 -1],[1 -1 2]));
[a,b,c,d]=ssdata(G);
% consider possibility that c is wrong in the design
% use c in design of K's and N's, but simulate with cpert
nom=2;
if nom
    cpert=c; % nominal case
else
    cpert=1.5*c; % off-nominal case
end
ns=size(a,1);
z=zeros(ns,1);
Rxx=c'*c;
Ruu=1;

klqr=lqr(a,b,Rxx,Ruu); % basic LQR gains
Nbar=inv(-c*inv(a-b*klqr)*b); % design with c
syslqr=ss(a-b*klqr,b*Nbar,cpert,0); % simulate with cpert

Rxx1=[Rxx z;z' 30];
Rxx2=[Rxx z;z' 1];

abar=[a z;-c 0];bbar=[b;0]; % design with c
kbar1=lqr(abar,bbar,Rxx1,Ruu);
kbar2=lqr(abar,bbar,Rxx2,Ruu);
abarpert=[a z;-cpert 0];  % simulate with cpert
sys1=ss(abarpert-bbar*kbar1,[z;1],[cpert 0],0);
sys2=ss(abarpert-bbar*kbar2,[z;1],[cpert 0],0);

figure(1);
t=[0:.1:10]';
[ylqr,tlqr]=step(ss(syslqr),t);
[y1,t1]=step(ss(sys1),t);
[y2,t2]=step(ss(sys2),t);
plot(t,ylqr,t,y1,t,y2,'LineWidth',2)
xlabel('Time')
legend('LQR with N','Int Penalty 30','Int Penalty 1','Location','SouthEast')

if nom
    export_fig lqrservo1 -pdf
else
    export_fig lqrservo2 -pdf
end
```

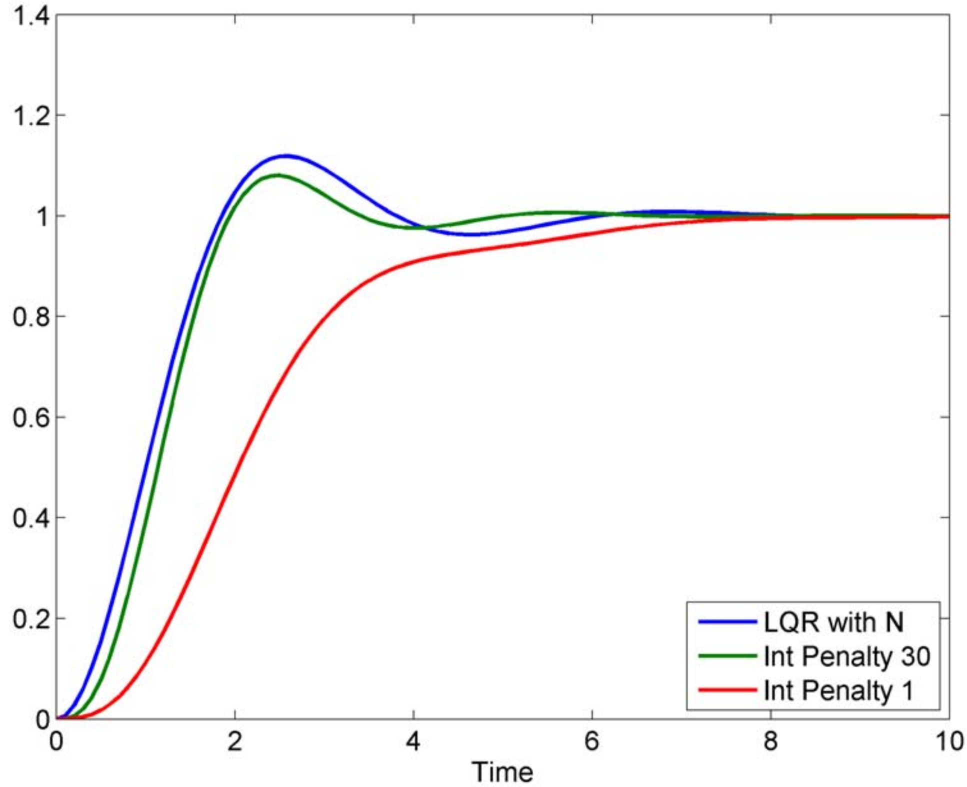

Fig. 2: Comparison of use of integrator and N weighting - nominal case

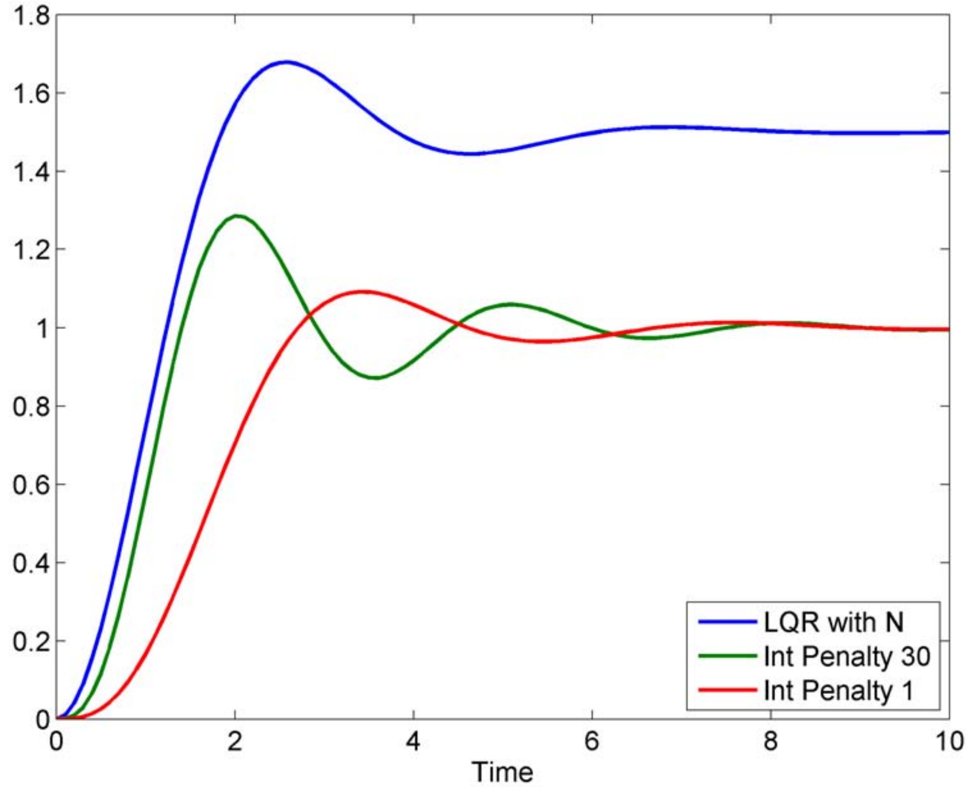

Fig. 3: Comparison of use of integrator and N weighting - off-nominal case