

ECE 327: Lecture 6

Pole Placement

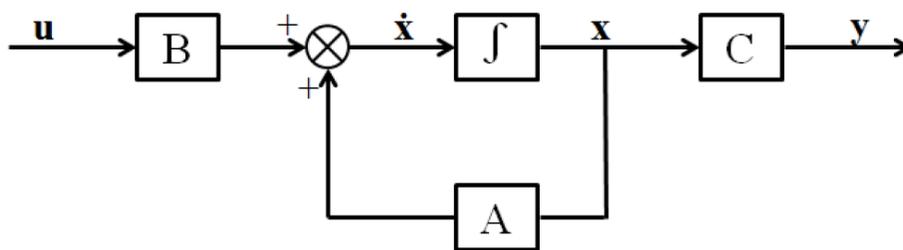
Stability

In order to examine whether a system is stable, we have to calculate the eigenvalues of the system matrix A (equal to the poles of the transfer function). Given that we have already defined the system matrix A in the workspace, we can achieve that by using the command `eigs = eig(A)`. Alternatively, given that we have already defined our system in the workspace inside variable `sys` (and not just matrix A) we can use the command `isstable(sys)`. Matlab then returns a logical value of whether 1 or 0, where 1 means "True" and 0 means "False".

Additionally, as we show next when we need to perform the pole placement technique in order to make a system stable or change its response, we first examine whether the system is controllable or not. If it is controllable, then the control signal u , can control the behavior of each state variable in x . In the case that the control signal u cannot affect one or more state variables (i.e., the system is uncontrollable), then pole placement technique is not possible.

Control Design Using Pole Placement

When the system is controllable, the poles of the closed loop system can be placed at any desired point using an appropriate **feedback gain matrix**. In the following, we present an analysis of the procedure to introduce additional parameters into a system so that we can control the location of closed-loop poles.



State space system representation without state feedback

Consider the following state space system

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t)\end{aligned}$$

Define a new input $u(t)$ of the form $u(t) = -Kx(t) + r(t)$, where

$$K = [k_1, k_2, \dots, k_n] \in \mathbb{R}^{1 \times n}$$

is the feedback gain matrix and $r(t)$ is a new input (reference) signal. Thus,

$$u(t) = -[k_1 \quad k_2 \quad \dots \quad k_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + r(t) \Rightarrow u(t) = -k_1 x_1 - k_2 x_2 - \dots - k_n x_n + r(t).$$

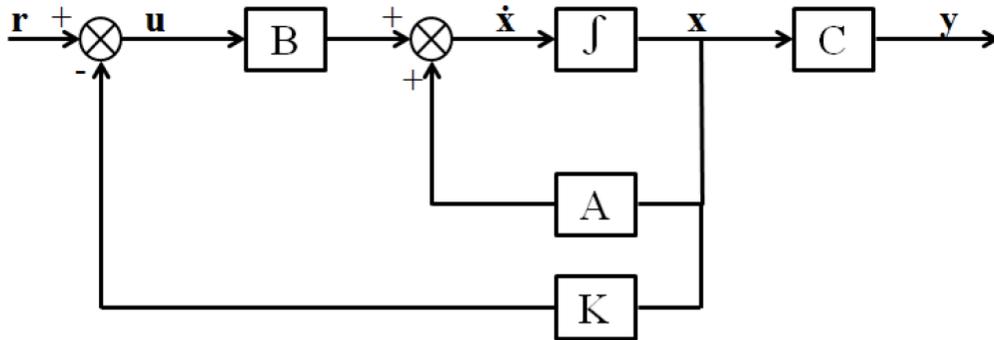
This is called state feedback, since we feed back to the input $u(t)$ the states x_1, \dots, x_n multiplied by the elements of matrix K , which are called gains. By replacing $u(t)$ in our state space equation we get

$$\dot{x}(t) = Ax(t) + B(-Kx(t) + r(t)) = (A - BK)x(t) + Br(t).$$

Hence, the state space system has the following form

$$\begin{aligned} \dot{x}(t) &= (A - BK)x(t) + Br(t) \\ y(t) &= Cx(t). \end{aligned}$$

Now, system stability depends on the location of the eigenvalues of matrix $A - BK$. Our purpose is to find the appropriate gains, in other words matrix K , in order to have the desired placement of the eigenvalues (of matrix $A - BK$). Of course, this placement will take place in the left-half plane of the complex numbers, in order for our system to be stable.



State space system representation with state feedback

If our system is controllable, pole placement can be achieved with the command $K = \text{place}(A, B, P)$. By placing our desired eigenvalues in variable P , Matlab then calculates and returns matrix K , inside the variable K . Variables A and B contain matrices A and B , respectively.

Example. For the plant

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ -1 & -2 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u$$

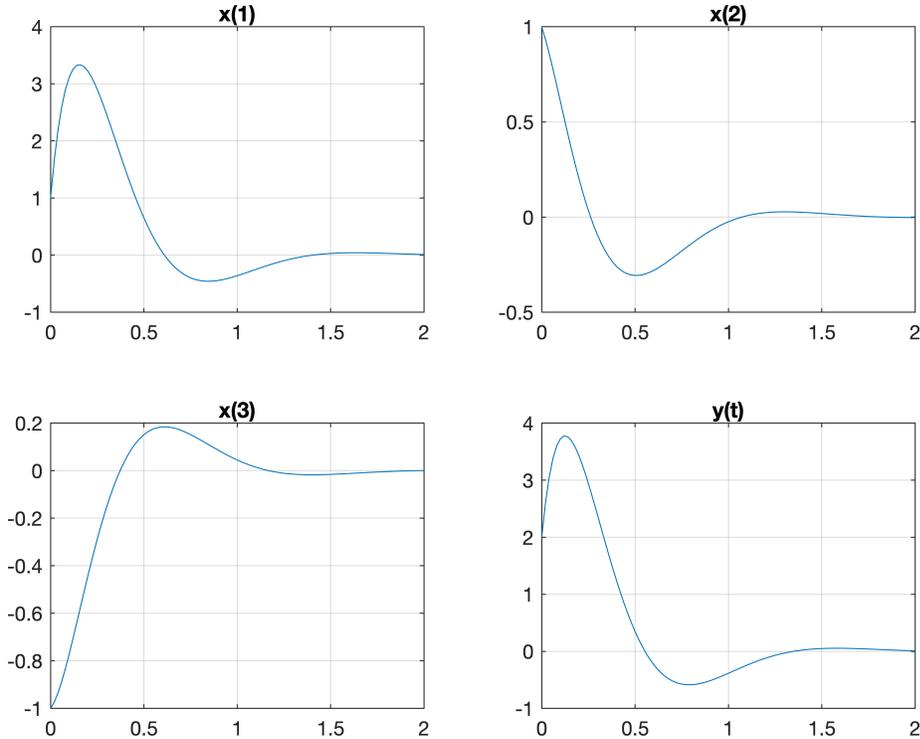
$$y = [1 \ 1 \ 0]x$$

design a state feedback control to place the closed-loop pole at $-3 \pm j4$ and -8 . Obtain the initial condition response, given $x_1(0) = 1$, $x_2(0) = 1$ and $x_3(0) = -1$.

```
A = [-1 0 0; -1 -2 0; 1 0 0];
B = [1; 0; 0];
C = [1 1 0]; D = 0;
k = place(A,B,[-3+4i,-3-4i,-8])
```

```
k = 1x3
    11.0000    51.0000   100.0000
```

```
t = 0:.02:2;
r = zeros(1,length(t));
x0 = [1 1 -1];
[y,x] = lsim(A-B*k,B,C,D,r,t,x0);
subplot(221),plot(t,x(:,1)),title('x(1)'),grid;
subplot(222),plot(t,x(:,2)),title('x(2)'),grid;
subplot(223),plot(t,x(:,3)),title('x(3)'),grid;
subplot(224),plot(t,y),title('y(t)'),grid;
```



```
%% or, alternatively
```

```
clf
sys=ss(A-B*k,B,C-D*k,D);
[y,t,x] = initial(sys,x0);
subplot(221),plot(t,x(:,1)),title('x(1)'),grid;
subplot(222),plot(t,x(:,2)),title('x(2)'),grid;
subplot(223),plot(t,x(:,3)),title('x(3)'),grid;
subplot(224),plot(t,y),title('y(t)'),grid;
```

