Teaching State Variable Feedback to Technology Students Using MATLAB and SIMULINK

**Kathleen A.K. Ossman, Ph.D.**
University of Cincinnati

I.   Introduction

This paper describes a course and laboratory in State Variable Feedback developed as a technical elective for students in the Electrical and Computer Engineering Technology (ECET) program at the University of Cincinnati. Students pursuing the EET degree are required to take courses in Signals and Systems and Feedback Controller Design. They learn to design lead, lag, and proportional-integral-derivative controllers (both analog and digital) for single-input single-output systems. The technical elective in State Variable Feedback allows students to explore the added benefits of a state variable controller when applied to multi-input multi-output systems and/or systems where constraints are imposed on the controller (i.e., tracking in minimum time or tracking with minimum control effort expended). In the past, a course in State Variable Feedback would have been too mathematically rigorous to include in a technology program. However, with the use of MATLAB and SIMULINK[1], students are able to design state variable feedback controllers and simulate the system responses without getting caught up in complicated matrix manipulations and integral calculations.

II.   Modeling Systems

The course begins with modeling of several types of systems including electrical circuits, mechanical translational systems, motors, a robot gripper, an inverted pendulum on a cart, and pitch control for a helicopter using both differential equations and state variable models[2,3]. Students have been exposed to system modeling using differential equations and transfer functions in the pre-requisite courses, but state equations are an entirely new concept. In course assignments and in lab, students build models of systems in SIMULINK using both transfer function blocks and state space blocks in order to better understand the similarities between these system models. In MATLAB, the functions *ss2tf* and *tf2ss* are introduced as tools for switching between the two types of system models for single-input-single-output (SISO) systems.

As an example, one of the systems explored in lab is the two-mass system shown in Figure 1.
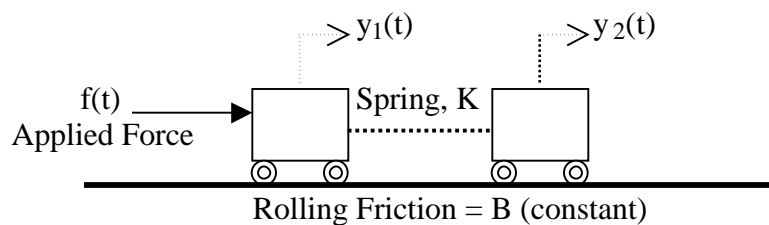
Figure 1: Two-mass System

The connection between the two carts is modeled using a spring with a spring constant, K. The friction is assumed to be a constant, B. M_1 and M_2 represent the mass of Cart 1 and Cart 2 respectively. Students are first asked to derive the differential equations that model this system:

$$M_1 d^2 y_1/dt^2 = f(t) - K(y_1 - y_2) - B dy_1/dt$$
$$M_2 d^2 y_2/dt^2 = K(y_1 - y_2) - B dy_2/dt$$

The next step is to derive the state model by defining the states to be the position and velocity of each of the carts. Defining the states as physical system variables as opposed to the states of the controllability model is extremely important in promoting technology students' understanding of the concept of a state. By defining $x_1 = y_1$ (position of cart 1), $x_2 = y_2$ (position of cart 2), $x_3 = dy_1/dt$ (velocity of cart 1), and $x_4 = dy_2/dt$ (velocity of cart 2), students derive the following state variable model:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -K/M_1 & K/M_1 & -B/M_1 & 0 \\ K/M_2 & -K/M_2 & 0 & -B/M_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1/M_1 \\ 0 \end{bmatrix} f(t)$$

The final step of this exercise is for students to build the system in SIMULINK and explore the system step response using several different values for the spring constant, K, and the coefficient of friction, B. The following example illustrates the effect of the spring constant on relative cart displacement. Figure 2(a) shows the block diagram constructed in SIMULINK and Figure 2 (b) shows the resulting cart displacements.
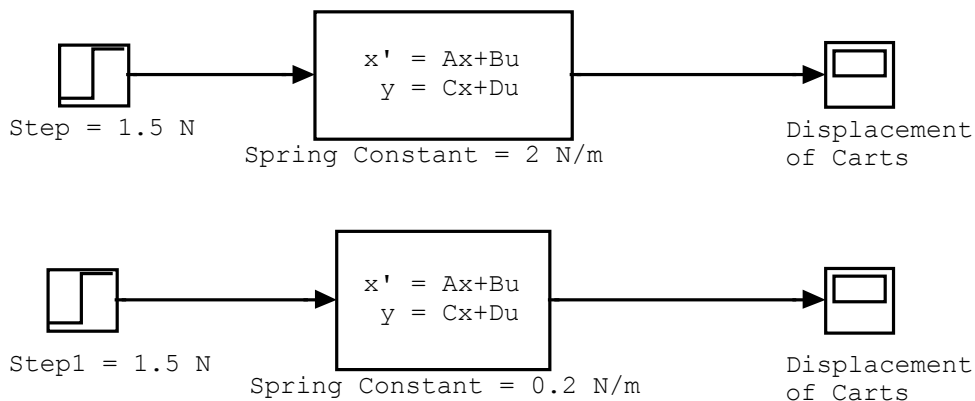
Figure 2 (a): Two-mass System in SIMULINK
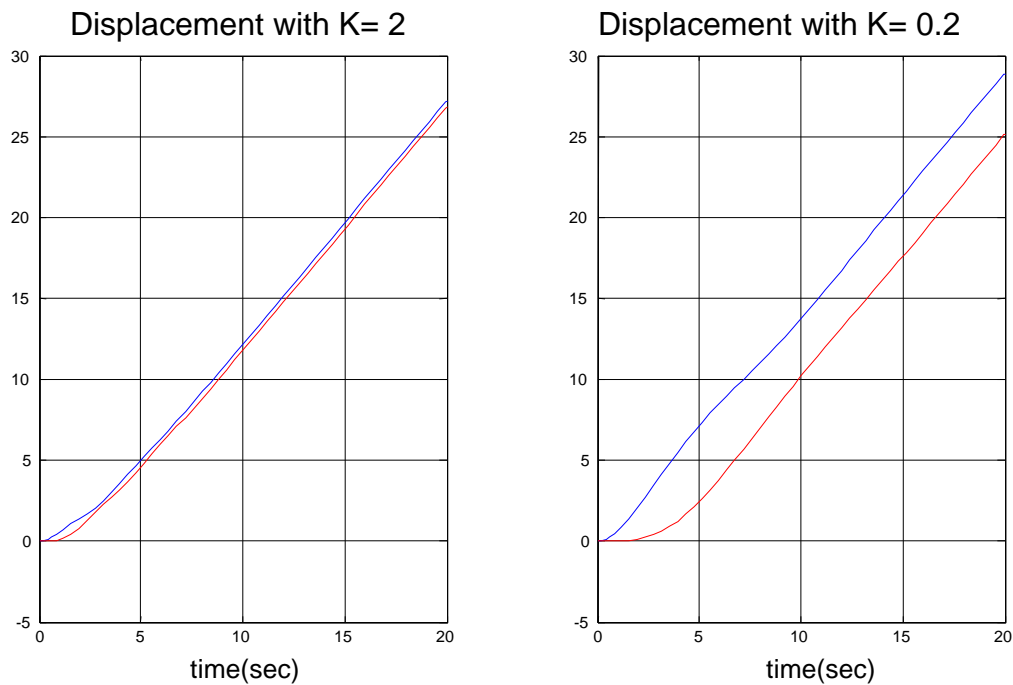
Figure 2 (b): Cart Displacements with Varying Spring Constants

Three important teaching considerations emerge from this section on modeling:

➢ Using a wide variety of system models generates student interest in the subject.
➢ Defining the states as physical system variables as opposed to the states of the controllability model is extremely important in promoting technology students' understanding of the concept of a state.
➢ Building the system in SIMULINK gives students a much clearer picture of the system model than using MATLAB commands to generate system responses.

III.   Transient Response

An understanding of how pole and zero locations affect the transient response of a system is critical when designing a state feedback controller based on pole placement. The equations for 2nd order systems that relate percent overshoot and settling time for the step response to the damping coefficient and natural frequency of the closed-loop system poles are given by:

$$Percent\ Overshoot = \exp(-\pi\zeta / \text{sqrt}(1\text{-}\zeta^2)\ )$$
$$Settling\ Time = 4 / (\zeta\omega_n)$$

Where:

ω_n = natural frequency = magnitude of poles
ζ = damping ratio = $-$ Real[pole]/ω_n

Students are familiar with these equations from a previous course in output feedback. The effect of extra poles and zeros on the system response is then reviewed. In lab, students complete some

simple yet useful examples which illustrate that zeros in the vicinity of the dominant complex poles will increase the percent overshoot while extra poles can slow the system response significantly. One of the exercises showing the effect of zeros on system response is included here.
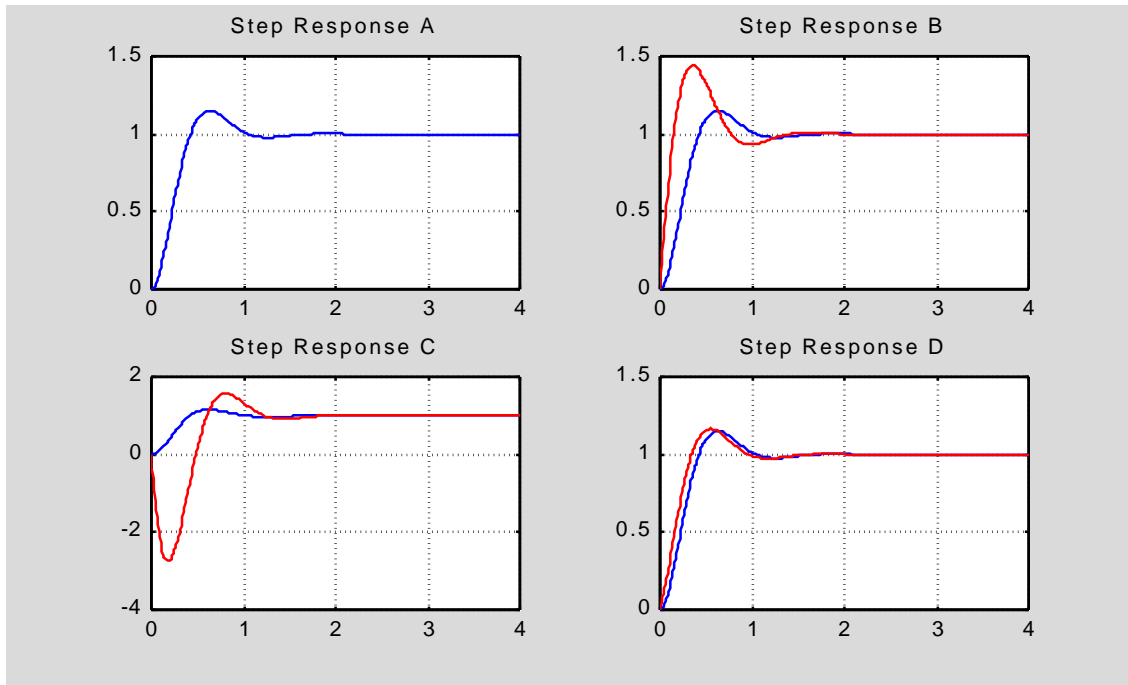
Figure 3: Effect of Zeros on System Transient Response

- System A is a 2nd order system with complex poles at $-3 \pm j5$.
- System B has the same poles as System A with an additional zero at -4. The graph shows the increase in percent overshoot due to this additional zero, which is located fairly close to the complex poles.
- System C has the same poles as System A with an additional zero at $+1$ (non-minimum phase system). This right half plane zero causes the system response to initially go the wrong way.
- System D has the same poles as System A with an additional zero at -15. Since this zero is not close to the complex pole, it has virtually no effect on the step response.

Three important teaching considerations emerge from this section on transient response:

- ➢ It is critical for students to understand the relationship between pole locations and transient response (percent overshoot, rise time, and settling time) in order to be able to choose desirable pole locations for a state feedback controller.
- ➢ It is also critical for students to understand how additional poles and/or zeros in the system can have undesirable effects on the transient response and to understand how to counteract these effects.
- ➢ Examples using MATLAB and SIMULINK facilitate students making the necessary connections between pole and zero locations and system transient response.

IV.   Advantages of State Feedback over Output Feedback

Although output feedback is sufficient for many systems, state feedback is very useful for multi-input multi-output systems and for control systems with optimal constraints such as those requiring minimal control effort or minimum time to final value. Several examples are discussed in lecture and lab that illustrate the advantages of state feedback. Only one example, a position control system for an armature-controlled dc motor, is included in this paper. The differential equations modeling the motor are derived in class. A state model is derived by defining two states: the position of the motor shaft and the velocity of the motor shaft. The system is first modeled in Simulink where students compare the motor response when feeding back only the shaft position (output feedback) to the motor response when both shaft position and shaft velocity are fed back (state feedback). A portion of the Simulink results is included here.

The first block diagram in Figure 4 is a position control system using output feedback and the second block diagram is a position control system using state feedback (both position and velocity). Students experiment with several different gain values and observe the effect on the system response.
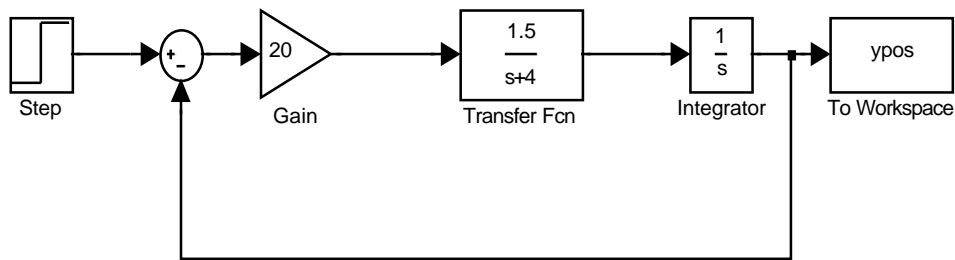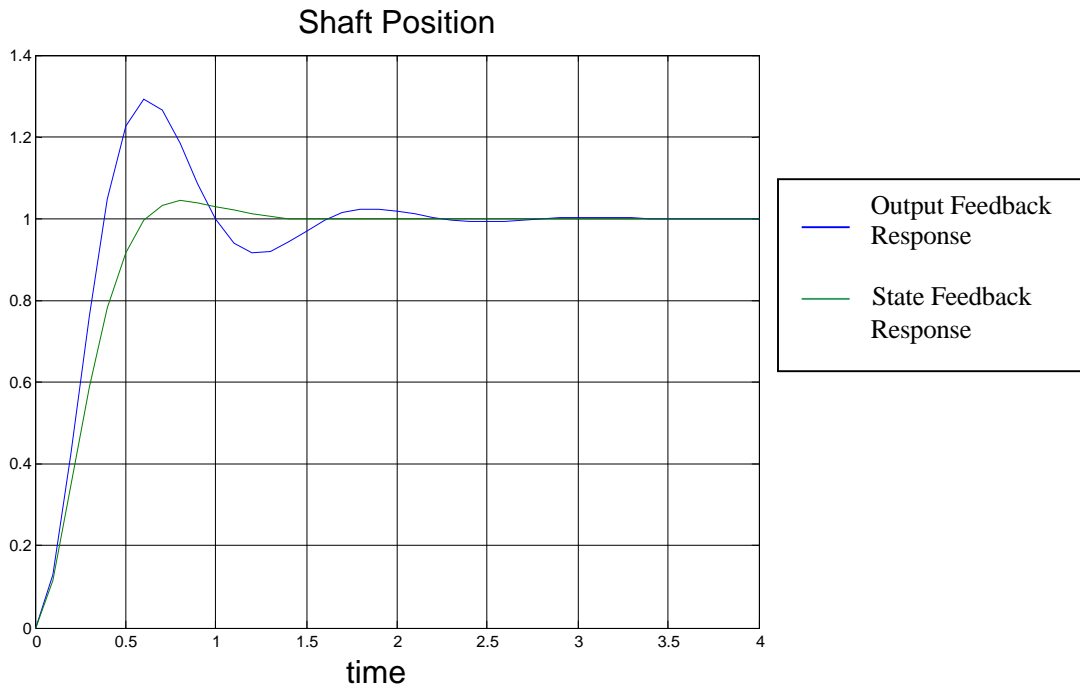
Output Feedback
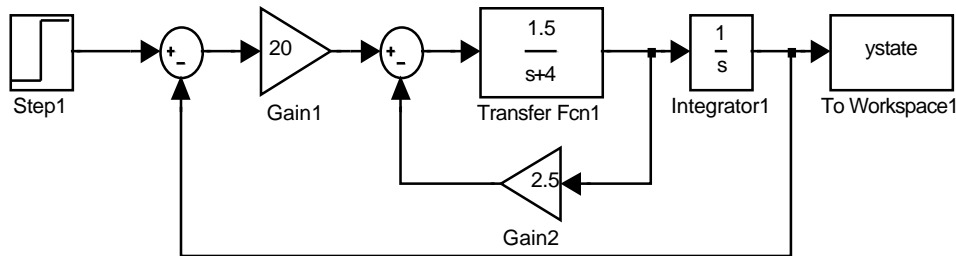
State Feedback

Figure 4: Output Feedback and State Feedback for a DC Motor

The response for one particular set of gains is shown in the scope plot in Figure 5. Including velocity feedback adds damping to the system (reduces overshoot) and speeds up the system response (reduces settling time).

Figure 5: Shaft Position of Motor

In lab, students experiment with output and state feedback using a Servo Trainer designed by Feedback Incorporated[4]. Students are able to vary the feed-forward gain and the tachometer gain and observe the effects on the motor response. This lab is valuable in providing a good visual illustration of the power of state feedback.

As a final exercise, students derive the transfer functions of the state feedback and output feedback systems leaving the gains as variables. They can then show state feedback gives them enough degrees of freedom to place the closed-loop poles in any desired position. Output feedback on the other hand, limits their choice of closed-loop pole positions.

Important teaching considerations that emerge from this section on state feedback are:

➢ Examples using MATLAB and SIMULINK that clearly illustrate the advantages of state feedback are critical in helping students understand the advantage of using the more complicated state-feedback controller.
➢ A hands-on lab such as the DC motor lab described above is an invaluable tool for demonstrating the concept of state-feedback, particularly if the states are physical variables of the system.

V.   Pole Placement Using State Feedback

The theory behind pole placement using state feedback is briefly discussed with the students. Proofs are not included in the lecture. We begin with the state model which students at this point have become comfortable with:

$$dx(t)/dt = Ax(t) + Bu(t)$$
$$y(t) = \;\; Cx(t) + Du(t)$$

The controllability matrix is then defined as $[B \; AB \; A^2B \; \ldots \; A^{n-1}B]$ where n is the system order or equivalently the number of states. The theorem concerning pole placement is then introduced.

Theorem: The system is controllable (i.e., the closed-loop poles can be placed in any desired position) if the determinant of the controllability matrix $[B \; AB \; A^2B \; \ldots \; A^{n-1}B]$ is non-zero.

The controllability model for a system is then introduced. It is shown that for any given set of desired poles, a set of feedback gains can be derived to place the system closed-loop poles at the desired positions. Students are reminded of the exercise with the dc motor (discussed in the previous section) where they derived a closed-loop transfer function and showed they had complete freedom in specifying the location of the poles by properly choosing the feedback gains. Finally, Ackermann's formula is introduced to compute the state-feedback gains to place the poles in the desirable positions:

$$K = \underset{(1 \; x \; n)}{[0 \;\; 0 \ldots \;\; 0 \;\; 1]} \; \underset{(n \; x \; n)}{[B \;\; AB \;\; A^2B \;\; \ldots \;\; A^{n-1}B]^{-1}} \; \underset{(n \; x \; n)}{\alpha_c(A)}$$

where $\alpha_c(s)$ is the desired characteristic polynomial for the closed-loop system.

One of the examples given in lecture is a state-feedback system to control the pitch angle of a helicopter. The equations of motion taken from [3] are given by:

$$d^2\theta/dt^2 = -0.415 \, d\theta/dt - 0.0111 \, dx/dt + 6.27 \, \delta$$
$$d^2x/dt^2 = 9.8 \, \theta - 1.43 \, d\theta/dt - 0.0198 \, dx/dt + 9.8 \, \delta$$

where θ is the pitch angle, x is the translation in the horizontal direction, and δ is the rotor angle.

Students are given a step-by-step procedure for designing a state-feedback controller. The steps with application to the pitch control system for the helicopter are included here. The performance specifications for this controller are a maximum 20% overshoot to a step change in the rotor angle and a maximum settling time of 10 seconds.

Step 1: *Derive the state model and enter it into MATLAB.*

The states are defined as:

$\quad\quad x_1 = \theta$ (pitch angle)

$\quad\quad x_2 = d\theta/dt$ (pitch velocity)

$\quad\quad x_3 = dx/dt$ (horizontal velocity)

The input is the rotor angle, δ, and the output is the resulting pitch angle, θ. The MATLAB statements for entering the state model are:

```
a=[0 1 0; 0 -0.415 -0.0111; 9.8 -1.43 -0.0198];
b=[0; 6.27; 9.8]; c=[1 0 0]; d=0;
```

Step 2: *Check for controllability of the system.*

```
det(ctrb(a,b))
```

ans = -2.4161e+003   ⇒ The system is controllable.

Step 3: *Find the zeros of the system.*

```
[num,den]=ss2tf(a,b,c,d); format short e
zeros=roots(num)
```

zeros = -2.4507e-003

Step 4: *Determine the desired pole locations based on the performance specifications.*

The damping coefficient corresponding to a percent overshoot of 20% is $\zeta = 0.45$. The natural frequency, ω_n, needed to satisfy the settling time specification and the resulting closed-loop poles are calculated using the MATLAB statements:

```
zeta=0.45; ts=10;
wn=4/(zeta*ts)
realpart=-zeta*wn
imagpart=wn*sqrt(1-zeta^2)
```

wn = 8.8889e-001
realpart = -4.0000e-001
imagpart = 7.9380e-001

The desired dominant complex poles are at $-0.4 \pm j\,0.8$. What about the other pole? Place the other pole at -0.0025 to cancel the effects of the zero. It is a very interesting exercise to show students what happens if this zero is neglected and the third pole is chosen far to the left of the complex poles, say at -2.0.

Step 5: *Find the feedback gains to place the poles at the desired positions.*

(a) *Find the desired denominator for the closed-loop transfer function:*

```
desired = conv([1 0.4 + 0.8j],[1 0.4-0.8j]);
desired = conv(desired, [1 0.0025]);
```

(b) *Calculate the feedback gains using Ackermann's formula*

```
k=[0 0 1]*inv(ctrb(a,b))*polyvalm(desired,a)
```

k = 1.2760e-001 6.1410e-002 -1.7697e-003

(c) Check to make sure the closed-loop poles are in the desired locations:

```
anew=a-b*k; [clnum,clden]=ss2tf(anew,b,c,d); poles=roots(clden)
```

```
poles =
 -4.0000e-001 +8.0000e-001i
 -4.0000e-001 -8.0000e-001i
 -2.5000e-003
```

Step 6: *Simulate the system response to check specifications:*

The system response can be generated using the MATLAB commands *step* or *lsim* but students get a better understanding of the feedback system by building it in SIMULINK using the State Variable block. The block diagram for the pitch angle control system and the system response to a step change in the rotor angle are shown in Figure 6.
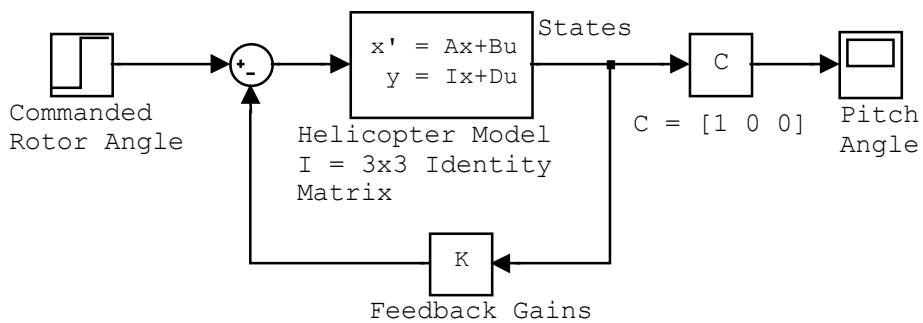
Figure 6 (a): State Feedback Controller for Pitch Angle of Helicopter
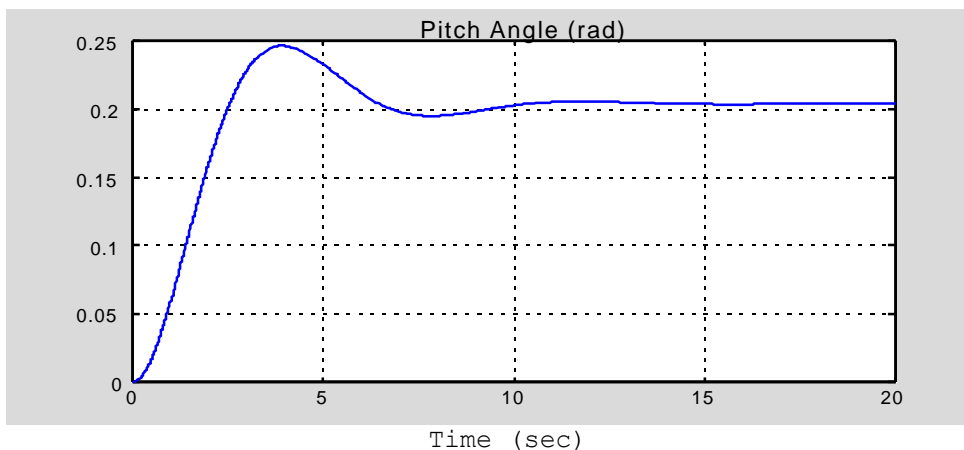
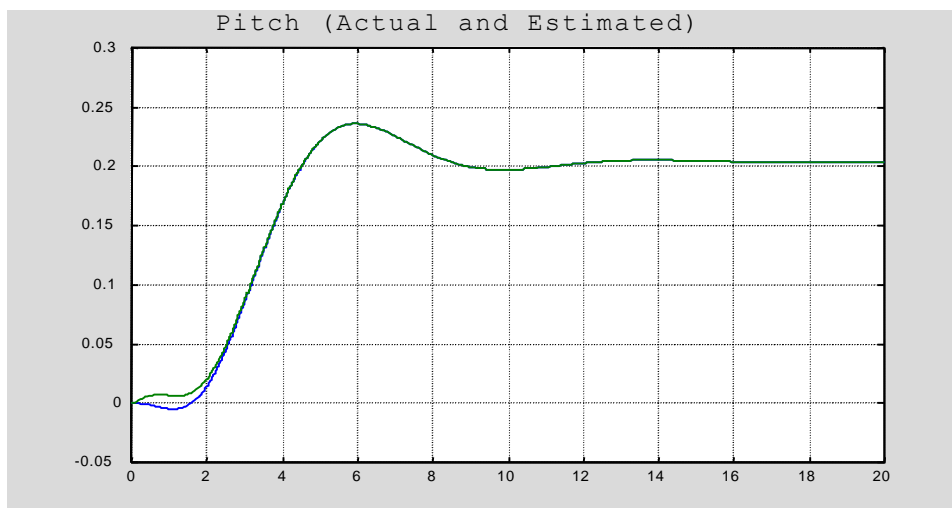**Figure 6 (b): Pitch Angle of Helicopter**

Teaching Considerations that emerge from this section on pole placement include:

➢ Although the state-feedback design and simulation can be done completely in MATLAB, building the system using SIMULINK gives students a better understanding of what the physical control system would look like.

➢ Having students check the closed-loop poles reinforces the validity of Ackermann's formula and the pole-placement technique. This is a useful "replacement" of the proofs included in a more rigorous course in state-feedback control.

VI.   Designing Observers for State Estimation

In many systems, some of the states of the system may not be measurable and available for state feedback. In this case, the actual state is replaced by an estimate of that state derived from a state estimator or a state observer. As in the case of state feedback, the theory behind state estimation is discussed in lecture without any formal proofs of theorems. Reduced order state estimators are discussed and designed in lecture and lab but will not be included in this paper.

Students are given a step-by-step procedure for designing a state estimator. The steps are similar to those outlined in the previous section and therefore are not included here. An example of state estimation to control pitch angle of the helicopter discussed in the previous section is included. The performance specifications for the controller are the same as those outlined in the previous section: a maximum 20% overshoot to a step change in the rotor angle and a maximum settling time of 10 seconds. Therefore, the feedback controller gains are identical to those computed in the previous section. A desired pitch angle of 0.2 radians is assumed. Also, the initial conditions of the system are assumed to be $\theta = 0$; $d\theta/dt=0$; and $dx/dt=15$. The initial conditions for the estimator are all assumed to be zero. The state-feedback controller and state estimator are built using SIMULINK. The system block diagram and the system response are shown in Figure 7.

Figure 7 (a): Actual and Estimated Pitch Angles
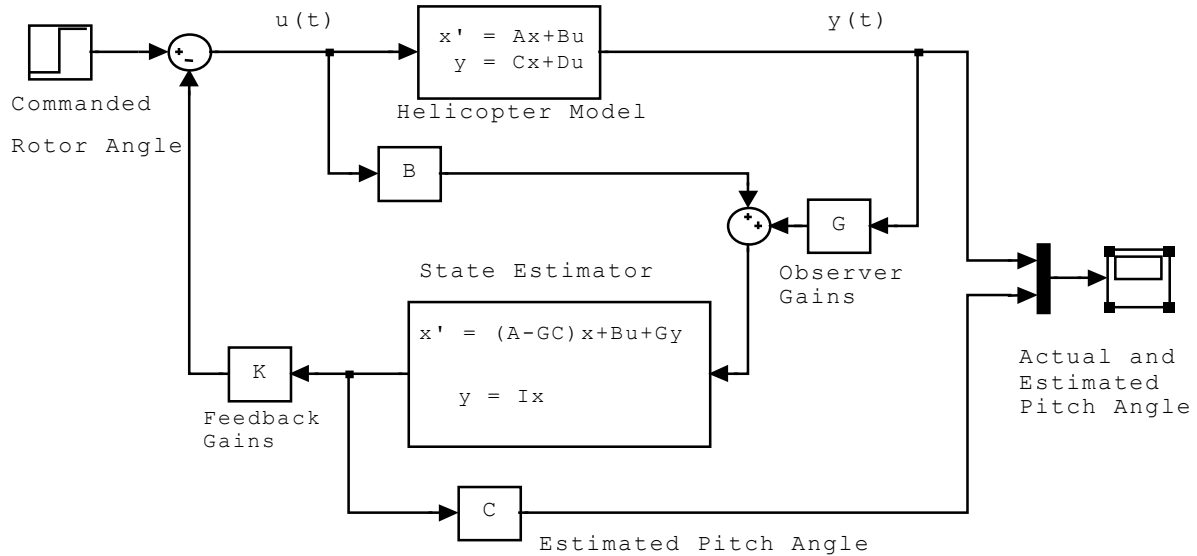
Figure 7 (b): Pitch Control for Helicopter using State Estimation

VII.   Conclusion

A course and lab in state-variable feedback was developed as a technical elective for students in Electrical Engineering Technology at University of Cincinnati. The lectures, exercises, and labs were designed using MATLAB and SIMULINK to illustrate the advantages of state-feedback control without using rigorous mathematics.

Bibliography

[1] MATLAB and SIMULINK, The MathWorks, Inc., 24 Prime Park Way, Natick, MA 01760.

[2] Feedback Control Systems, Phillips and Harbor, 4th Edition, Prentice Hall, 2000.

[3] Modern Control Systems, Dorf and Bishop, 7th Edition, Addison-Wesley, 1995.

[4] Feedback Incorporated, P.O. Box 400, 437 Dimmocks Mill Rd., Hillsborough N.C. 27278.

Biography

Dr. Kathleen Ossman is an assistant professor in the Electrical and Computer Engineering Technology Department at the University of Cincinnati. She received a BSEE and MSEE from Georgia Tech in 1982 and a Ph.D. from the University of Florida in 1986. Her interests include feedback control systems and digital signal processing.