An Undergraduate State-Space Theory Course with Emphasis on Designs

Chiu H. Choi, Ph.D., P.E.

**Electrical Engineering Program
Division of Engineering
University of North Florida
Jacksonville, Florida 32224**

Abstract

A new approach of enhancing undergraduate engineering courses is proposed in this paper. The enhancement is the integration into the courses a wide range of practical design problems of which the solutions require in-depth knowledge of computational software package. This approach was tried on an undergraduate-level state-space theory course. The reasons for choosing this course were that the enrollment of this course had been low and the students who took this course before did not show much interest in the course contents. It was desired to investigate whether the new approach could increase the student's enthusiasm about the course and the enrollment. The new approach put much emphasis on the applications of the theory to solve a wide range of control design problems using MATLAB[®]. The associated theorems of the theory and their proofs were still covered in the course. The evaluation indicated that the enhancement had significantly increased the students' interest in the course and the enrollment. The design problems covered in the course included pole placement design using full-state feedback, full-order observer design, pole placement design using full-order observer, linear-quadratic regulator design, Kalman filter design, linear-quadratic Gaussian design. This design approach allowed the students to appreciate the application of the state-space theory at a deeper level. It is expected that enhancing a course with a wide-range of practical design problems will improve the teaching evaluation of the course significantly.

I. Introduction

Our students consider the undergraduate level state-space theory course as one of the hardest courses in the undergraduate electrical engineering curriculum. The course covers the basic methods for control system design and analysis using the state-space theory. Topics include linear algebra, review of dynamics, state-space modeling of control systems, time-domain responses of state space models, transformations, diagonalization, BIBO stability, asymptotic and marginal stability, controllability, observability, state feedback and pole placement, full-order observer design, reduced-order observer design, linear-quadratic regulator problem, Kalman filtering, linear-quadratic Gaussian problem, and the numerical solution of algebraic Riccati equations. Many of these topics are discussed in a number of textbooks such as [1], [2], [3], and [4].

The course is a sequel to our first controls course, which covers classical control methods developed up to 1948. The topics of the classical control course include Routh stability test (1877), Nyquist stability criterion (1932), frequency response methods (1938), and root locus techniques (1948). These topics are covered by many control textbooks such as [5], [6], [7], [8], [9], and [10]. A common feature of these classical methods is the use of transfer functions in the modeling of the control systems.

The state-space theory course has been in our curriculum for a number of years. It was first taught as a special topic course covering the theory in 1997. The enrollment was low. Later, the course was enhanced with simulation techniques [11]. The intention was to increase the students' interest in this course. The students did respond favorably to the techniques; however, the enrollment did not increase. The course was revised in Fall 2004 with the goal of improving the enrollment and to further increase the student's interest in the course. The revised course put more emphasis on the applications of the theory to solving practical control design problems using MATLAB and Control System Toolbox ([12], [13], [14]). The theory with its associated theorems and proofs were still covered.

There were several reasons to revise the course this way. First, it was anticipated that enhancing the course with a wide-range of practical design problems could increase the enrollment. Students are more interested in seeing the applications of the theory in solving practical problems. They show less interest in learning just the theory and their proofs. However, they know that it is important to understand the underlying theory and the proofs.

Second, the students want to see that there is a wide range of problems that can be solved by the theory. They show less interest in the course if the theory that they learn can only solve just a few problems.

Third, the students want to learn how to use computational tools for solving practical design problems and to acquire the computational design skills at a deeper level.

The course was revised accordingly. More emphasis was put on the applications. A wider range of control design problems was tackled. The procedures for solving these problems were clearly described and delivered to the students. Computer-aided design tools were taught at a deeper level. With these improvements, the students' knowledge and interest in state-space systems were much increased. They considered that the state-space course was among the best that they had ever taken. The enrollment for this enhanced course was fifteen in Fall 2004 and that for the previous offering was only three.

The contents of the rest of this paper are as follows: some of the control design problems developed for the course are described in Section II. They are the state feedback and pole placement, full-order observer design, linear-quadratic regulator design, Kalman filter design, and linear-quadratic Gaussian design problems. The design procedures and the simulation results are also provided. Section III describes how the course was enhanced by MATLAB. Section IV presents a summary of the students evaluation on this course. Section V gives an account of the lessons learned from this enhancement and recommendations. Section VI contains some concluding remarks.

II. Design Problems

In this section, five design problems are presented. The procedures for solving these problems are described and the simulation results are included. Sufficient details of the design procedures are provided so that the appropriate readers will have enough information to implement these designs into their own courses. MATLAB (version 6.5 R13) and the Control System Toolbox were used in the design process. The simulations were run on a PC equipped with Pentium® 4 CPU, 2.66GHz, and 512 MB of RAM.

Control design problem # 1. This is a problem of pole placement using full state feedback. Consider the following state-space model

$$\begin{aligned}\dot{x} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -5 & -6 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \\ y &= [12 \quad 0 \quad 0]x\end{aligned}\quad (1)$$

The problem is to design a controller to drive the step response of the compensated system to attain the following performance criteria: overshoot to be approximately 7 %, 2% settling time to be approximately 4.5 units, and the steady-state error to be approximately zero. This problem can be solved by placing the poles of the compensated system at certain locations using full state feedback. The procedure for solving this problem using MATLAB is described in the following.

1. Assume that A, B, C, and D matrices of (1) were created in MATLAB already. Form the state-space object `sys` for (1).

```
>> sys=ss(a,b,c,d)
```

```
a =
      x1      x2      x3
x1      0       1       0
x2      0       0       1
x3      0      -5      -6
```

```
b =
      u1
x1      0
x2      0
x3      1
```

```
c =
      x1      x2      x3
y1     12       0       0
```

```
d =
      u1
      y1  0
```

2. Specify the desired pole locations for 7% overshoot and 4.5 units of settling time.

```
>> p=[-10 -0.889+i*1.056 -0.889-i*1.056]
```

```
p =
-10.0000          -0.8890 + 1.0560i  -0.8890 - 1.0560i
```

3. Compute the gain matrix K so as to place the poles at the desired locations

```
>> k=place(a,b,p)
```

```
k =
19.0546    14.6855     5.7780
```

4. A way to remove the steady state error is to multiply the output y by a gain factor K_c, where K_c is given by

$$K_c = \frac{1}{-C(A-BK)^{-1}B}$$

Compute the gain factor K_c by MATLAB.

```
>> kc=1/(c*(-inv(a-b*k))*b)
```

```
kc =
1.5879
```

5. For simulation purpose, K_c is absorbed into the C matrix

```
>> c=kc*c
```

```
c =
19.0546          0          0
```

6. Create the state-space model of the compensated system

```
>> sys1=ss(a-b*k,b,c,d)
```

```
a =
      x1      x2      x3
x1      0      1      0
x2      0      0      1
x3 -19.05 -19.69 -11.78
```

b =

	u1
x1	0
x2	0
x3	1

c =

	x1	x2	x3
y1	19.05	0	0

d =

	u1
y1	0

7. Plot the step response of the compensated system to verify that the performance criteria are met. The step response is shown in Figure 1. Notice that the parameters of the step response are close to the desired parameters. If the parameters are not satisfactory, change the pole location slightly and repeat the design procedure until the desired parameters are attained.

```
>> step(sys1)
```

Figure 1: Step response of the compensated system for Design Problem #1.

Design problem # 2. This is a problem of pole placement using an estimate of the state vector generated by the full-order observer. Consider the state-space model below.

$$\begin{aligned}\dot{x} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y &= [1 \quad 0]x\end{aligned}\quad (2)$$

The problem is to design an observer to produce an estimate of the state vector x and use this estimate in lieu of the actual state vector to place the poles of the compensated system at such locations that the step response of the compensated system attains the following performance criteria: overshoot to be approximately 7 %, 2% settling time to be approximately 4.5 units, and steady-state error to be approximately zero.

Using the separation theorem, the observer design problem can be tackled separately from the pole placement problem. The states obtained from the observer are good estimates of the actual states (if the original system is observable) and can be used for placing the poles of the compensated system at the desired locations. Let the state-space equations for a state-space model be

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

Then, the observer equation is:

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + Bu + L(y - \hat{y}) = (A - LC)\hat{x} + Bu + Ly \\ \hat{y} &= C\hat{x}\end{aligned}\quad (3)$$

and the input equation is:

$$u = r - K\hat{x}, \quad (4)$$

where K is the gain matrix that drives the compensated system to have the desired poles.

The combined original system equation and the observer equation is

$$\begin{aligned}\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} &= \begin{bmatrix} A & -BK \\ LC & A - BK - LC \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} r \\ y &= [C \quad 0] \begin{bmatrix} x \\ \hat{x} \end{bmatrix}\end{aligned}\quad (5)$$

Define the error $e = x - \hat{x}$. The error equation can be obtained as $\dot{e} = (A - LC)e$. If the original system is observable, the poles of the error equation can be placed sufficiently far from the imaginary axis and in the left half plane so that the error will die away very quickly.

The procedure for solving this design problem using MATLAB and the simulation results are described in the following.

1. Assume that the A, B, C, and D matrices of (2) were created in MATLAB already.

Choose the observer poles to be at $-2+j2$ and $-2-j2$. To place the observer poles at such locations, the observer gain matrix L can be computed by the MATLAB command

```
>> l=place(a',c',[-2+j*2 -2-j*2])'
```

l =

```
4.0000
8.0000
```

The computed observer gain is $L = \begin{bmatrix} 4 \\ 8 \end{bmatrix}$. The observer equation (3) becomes

$$\dot{\hat{x}} = \begin{bmatrix} -4 & 1 \\ -8 & 0 \end{bmatrix} \hat{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 4 \\ 8 \end{bmatrix} y$$

2. To achieve 7% overshoot and 4.5 units of 2% settling time in the step response, the desired poles are $-0.889+j*1.056$ and $-0.889-j*1.056$. Use the “place” command to compute the gain matrix K for such pole placement.

```
>> k=place(a,b,[-0.889+j*1.056 -0.889-j*1.056])
```

k =

```
1.9055    1.7780
```

The input equation (4) becomes $u = r - [1.9055 \quad 1.7780]\hat{x}$.

3. Form the coefficient matrices of (5). This equation is needed in order to simulate the step response of the compensated system.

```
>> a1=[a -b*k; l*c a-l*c-b*k]
```

a1 =

```
0    1.0000    0    0
0    0    -1.9055    -1.7780
4.0000    0    -4.0000    1.0000
8.0000    0    -9.9055    -1.7780
```

```
>> b1=[b;b]
```

b1 =

```

0
1
0
1

>> c1=[c 0 0]

c1 =

    1    0    0    0

>> d1=[0]

d1 =

    0

```

4. A way to remove the steady state error is to multiply the output y by a gain factor K_c where

$$K_c = \frac{1}{-C(A-BK)^{-1}B}$$

Calculate K_c by MATLAB.

```

>> kc=1/(c*(-inv(a-b*k))*b)

kc =

    1.9055

```

5. For simulation purpose, absorb this gain K_c into the C1 matrix.

```

>> c1=c1*kc

c1 =

    1.9055    0    0    0

```

6. Form the state-space model of the compensated system.

```

>> sys1=ss(a1,b1,c1,d1)

a =

      x1      x2      x3      x4
x1      0      1      0      0
x2      0      0  -1.905  -1.778
x3      4      0      -4      1
x4      8      0  -9.905  -1.778

```

```

b =

```



```

u1
x1  0
x2  1
x3  0
x4  1

```

```

c =
      x1      x2      x3      x4
y1  1.905      0      0      0

```

```

d =
      u1
y1  0

```

7. Plot the step response. It is shown in Figure 2. Notice that the parameters of the step response are close to the desired parameters. If the parameters are not satisfactory, change the pole location slightly and repeat the design procedure until the desired parameters are attained.

```
>> step(sys1)
```

Figure 2: Step response of the compensated system for Design Problem #2.

Design problem # 3. This is a problem of linear quadratic regulator design problem. Consider a state-space model

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

Define the cost function: $J = \frac{1}{2} \int_0^{\infty} (x^T Qx + u^T Ru + 2x^T Nu) dt$

The objective is to find the input u to minimize the cost function J and to asymptotically stabilize the state-space system. There is a solution to the problem if (A, B) is stabilizable and (C_q, A) is detectable, where $Q = C_q^T C_q$.

The input u is found to be of the form $u = -Kx$, where $K = -R^{-1} B^T P x$. The matrix P is the positive semi-definite solution of the algebraic Riccati equation

$$A^T P + PA + Q - (PB + N)R^{-1}(B^T P + N^T) = 0$$

The MATLAB command `[K, P, E]=lqr(A, B, Q, R, N)` computes the feedback gain matrix K , the solution P of the algebraic Riccati equation, and the closed-loop poles of the optimal system $A-BK$. The matrix N is set to zero when omitted. An example of using this command to solve an LQR problem is given below.

Example: Consider a state-space model with $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $C = [1 \ 0]$, $D = [0]$,

$Q_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, and $R=1$. The steps for solving this problem is described in the following.

1. Form the matrices A through R .

```
>> a=[0 1 ;0 0];
>> b=[0;1];
>> c=[1 0];
>> d=0;
>> q=[1 0; 0 0];
>> r=1;
```

2. Apply the `lqr` command to compute K .

```
>> [k,p,e] = lqr(a,b,q,r)
```

$k =$

```
1.0000    1.4142
```

p =

$$\begin{bmatrix} 1.4142 & 1.0000 \\ 1.0000 & 1.4142 \end{bmatrix}$$

e =

$$\begin{bmatrix} -0.7071 + 0.7071i \\ -0.7071 - 0.7071i \end{bmatrix}$$

3. The input $u = -[1 \ 1.4142]x$ minimizes the cost function and stabilizes the system.

Design problem #4. This is a problem of Kalman filter design. Consider the state-space model

$$\begin{aligned}\dot{x} &= Ax + Bu + Gw \\ y &= Cx + Du + Hw + v\end{aligned}$$

The noise sources w and v are zero-mean Gaussian with covariance matrices $R_0 = E[ww^T]$, $Q_0 = E[vv^T]$, and $N_0 = E[ww^T]$. The objective is to obtain an estimate $\hat{x}$ of the state x that minimizes the estimation error:

$$Error = E[(x - \hat{x})^T (x - \hat{x})]$$

The optimal estimate $\hat{x}$ is given by the following equations:

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + Bu + L(y - C\hat{x} - Du) \\ \hat{y} &= C\hat{x} + Du\end{aligned}\tag{6}$$

The filter gain matrix L is given by $L = \Sigma C^T R_0^{-1}$. The matrix Σ satisfies the following algebraic Riccati equation:

$$A\Sigma + \Sigma A^T + GQ_0G^T - (\Sigma C^T + GN_0)R_0^{-1}(C\Sigma + N_0^T G^T) = 0$$

The MATLAB command `[Ksys, L, Sigma] = kalman(sys, Q0, R0, N0)` computes the coefficient matrices of the Kalman estimator `Ksys`, the filter gain matrix `L`, and the solution Σ of the associated algebraic Riccati equation. The first argument `sys` in the `kalman` command is the state-space model `ss(A, [B G], C, [D H])` and N_0 is assumed to be a zero matrix when omitted.

The `Ksys` generated by the `kalman` command consists of a set of four coefficient matrices:

$[A-LC]$, $[B-LD \ L]$, $\begin{bmatrix} C \\ I \end{bmatrix}$, and $\begin{bmatrix} D \\ 0 \end{bmatrix}$. These matrices are the coefficient matrices of the equations

$$\begin{aligned}\dot{\hat{x}} &= (A - LC)\hat{x} + (B - LD)u + Ly \\ \begin{bmatrix} \hat{y} \\ \hat{x} \end{bmatrix} &= \begin{bmatrix} C \\ I \end{bmatrix} \hat{x} + \begin{bmatrix} D \\ 0 \end{bmatrix} u\end{aligned}\quad (7)$$

which are a rearranged version of the Kalman filter equations (6).

Following is an example of using this command to solve a Kalman filter design problem.

Example: Consider a state-space model with $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $C = [1 \ 0]$, $D = [0]$,

$G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $H = [0 \ 0]$, $Q_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $R_0 = 1$, $N_0 = \text{zero matrix}$. The steps for solving this problem are given below.

1. Form the matrices A through R₀.

```
>> a=[0 1; 0 0];
>> b=[0;1];
>> c=[1 0];
>> d=0;
>> g=[1 0; 0 1];
>> h=[0 0];
>> q0=[1 0; 0 1];
>> r0=1;
```

2. Create the state-space object sys.

```
>> sys= ss(a,[b g],c,[d h])
```

a =

	x1	x2
x1	0	1
x2	0	0

b =

	u1	u2	u3
x1	0	1	0
x2	1	0	1

c =

	x1	x2
y1	1	0

d =

	u1	u2	u3
y1	0	0	0

3. Invoke the `kalman` command to obtain the coefficient matrices of the Kalman filter.

```
>> [Ksys,L,Sigma] = kalman(sys,q0,r0)
```

a =

	x1_e	x2_e
x1_e	-1.732	1
x2_e	-1	0

b =

	u1	y1
x1_e	0	1.732
x2_e	1	1

c =

	x1_e	x2_e
y1_e	1	0
x1_e	1	0
x2_e	0	1

d =

	u1	y1
y1_e	0	0
x1_e	0	0
x2_e	0	0

I/O groups:

Group name	I/O	Channel(s)
KnownInput	I	1
Measurement	I	2
OutputEstimate	O	1
StateEstimate	O	2,3

Continuous-time model.

L =

1.7321
1.0000

Sigma =

1.7321	1.0000
1.0000	1.7321

The Kalman filter equation (7) can be constructed from K_{sys} as

$$\dot{\hat{x}} = \begin{bmatrix} -1.732 & 1 \\ -1 & 0 \end{bmatrix} \hat{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 1.732 \\ 1 \end{bmatrix} y$$

Design problem #5. This is a problem of linear quadratic Gaussian problem. Consider a state-space model

$$\begin{aligned} \dot{x} &= Ax + Bu + Gw \\ y &= Cx + Du + Hw + v \end{aligned}$$

The noise sources w and v are zero-mean Gaussian with covariance matrices $R_0 = E[ww^T]$, $Q_0 = E[vv^T]$, and $N_0 = E\{wv^T\}$. The objective is to find a control law for $u(t)$ that minimizes the cost function

$$J = \frac{1}{2} E \left[\int_0^{\infty} (x^T Q x + u^T R u + 2x^T N u) dt \right]$$

The solution of this problem is divided into the Kalman filtering problem and the LQR problem. In the Kalman filtering problem, the estimate $\hat{x}$ of the state x is obtained from the equations

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + Bu + L(y - C\hat{x} - Du) \\ \hat{y} &= C\hat{x} + Du \end{aligned} \quad (8)$$

The matrix L can be obtained by using the `kalman` command.

Next, solve the LQR problem

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \\ J &= \frac{1}{2} \int_0^{\infty} (x^T Q x + u^T R u + 2x^T N u) dt \end{aligned}$$

using the `lqr` command.

The feedback gain matrix K obtained from the `lqr` command is used in the control law $u(t) = -K\hat{x}(t)$. Such $u(t)$ will minimize J .

With such $u(t)$, the Kalman filter equation becomes

$$\dot{\hat{x}} = (A - BK - LC + LDK)\hat{x} + Ly \quad (9)$$

Following is an example of using these commands to solve a linear quadratic Gaussain design problem.

Example: Consider a state-space model with $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $C = [1 \ 0]$, $D = [0]$,
 $G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $H = [0 \ 0]$, $Q_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $R=1$, $Q_0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $R_0 = 1$, and $N_0 =$ zero matrix.

The steps for solving this linear quadratic Gaussain problem are given below.

1. Form all the matrices.

```
>> a=[0 1; 0 0];
>> b=[0;1];
>> c=[1 0];
>> d=0;
>> g=[1 0;0 1];
>> h=[0 0];
>> q0=[1 0;0 1];
>> r0=1;
>> q=[1 0; 0 0];
>> r=1;
```

2. Create the state-space object sys.

```
>> sys=ss(a,[b g],c,[d h])
```

```
a =
      x1      x2
x1      0      1
x2      0      0
```

```
b =
      u1      u2      u3
x1      0      1      0
x2      1      0      1
```

```
c =
      x1      x2
y1      1      0
```

```
d =
      u1      u2      u3
y1      0      0      0
```

3. Obtain the Kalman filter equation.

```
>> [ksys,L,Sigma]=kalman(sys,q0,r0)
```

```
a =
```

	$x1_e$	$x2_e$
$x1_e$	-1.732	1
$x2_e$	-1	0

```
b =
```

	$u1$	$y1$
$x1_e$	0	1.732
$x2_e$	1	1

```
c =
```

	$x1_e$	$x2_e$
$y1_e$	1	0
$x1_e$	1	0
$x2_e$	0	1

```
d =
```

	$u1$	$y1$
$y1_e$	0	0
$x1_e$	0	0
$x2_e$	0	0

```
I/O groups:
```

Group name	I/O	Channel(s)
KnownInput	I	1
Measurement	I	2
OutputEstimate	O	1
StateEstimate	O	2,3

```
Continuous-time model.
```

```
L =
```

```
1.7321
1.0000
```

```
Sigma =
```

```
1.7321    1.0000
1.0000    1.7321
```

The Kalman filter equation (8) can be constructed from Ksys as

$$\dot{\hat{x}} = \begin{bmatrix} -1.732 & 1 \\ -1 & 0 \end{bmatrix} \hat{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 1.732 \\ 1 \end{bmatrix} y \quad (9)$$

4. Obtain the gain matrix K for the LQR problem.

```
>> [k,p,e]=lqr(a,b,q,r)
```

k =

```
1.0000    1.4142
```

p =

```
1.4142    1.0000
1.0000    1.4142
```

e =

```
-0.7071 + 0.7071i
-0.7071 - 0.7071i
```

The control law $u(t)$ can be constructed as $u(t) = -K\hat{x}(t) = -[1 \quad 1.4142] \hat{x}(t)$.

5. With such $u(t)$, the Kalman filter equation (9) becomes

$$\dot{\hat{x}} = \begin{bmatrix} -1.732 & 1 \\ -2 & -1.414 \end{bmatrix} \hat{x} + \begin{bmatrix} 1.732 \\ 1 \end{bmatrix} y \quad (10)$$

This Kalman filter equation can also be obtained by the command `lqrreg`. This command returns the coefficient matrices of (10) and other matrices.

```
>> rlqg=lqgreg(ksys,k)
```

a =

```
      x1_e    x2_e
x1_e  -1.732    1
x2_e   -2    -1.414
```

b =

```
      y1
x1_e  1.732
x2_e   1
```

c =

```
      x1_e    x2_e
```

```

u1      -1   -1.414

d =
      y1
u1      0

I/O groups:
  Group name      I/O      Channel(s)
  Measurement      I          1
  Controls          O          1

```

III. How the course was enhanced by MATLAB?

MATLAB provides a comprehensive set of capability for solving control design problems. It is a powerful computational tools for control systems design. It was used as a teaching tool and a computational tool in the revised course.

The course was enhanced by MATLAB in the following ways. As a teaching tool MATLAB allowed the design procedures to be delivered effectively to the students and to be within their attention span. For example, in the pole placement problems described in the previous section, just a few commands could show that the performance criteria were met by changing the pole locations. MATLAB could also simulate and display the expected responses so that the students could quickly verify the working of pole placement and state feedback. This enhanced the students' understanding of the underlying theory efficiently.

Also, by using MATLAB many of the students' questions on the state-space theory can be answered without delay in class. The questions included finding matrix exponentials and matrix functions, finding transfer functions of state-space models, determination of controllability and observability and others.

As a computational tool MATLAB avoided the lengthy hand calculations, which was time-consuming, distracted the students from focusing on the designs, and cooled off the students' enthusiasm about the course. The students found that solving the design problems was a rewarding experience when MATLAB was used. The students, however, did solve some problems of small dimensions by hand first in order to grasp the concepts.

Also, higher dimensional practical design problems could be assigned to the students without concern that they would take too much time to finish. With hand calculations the dimension of the design problems assigned were mostly two or three and rarely at four or higher. With MATLAB, the dimension of the design problems became much higher. Moreover, lengthy design problems could be assigned to the students too. This increased the productivity of the students.

Further, longer and harder design problems were included in the in-class tests and exams by allowing the students to use MATLAB during the tests and exams. This allowed better testing of

the students' understanding of the concepts and theory. It also allowed deeper assessment of their computational skills.

IV. Students Feedback

Two surveys were conducted for this course in Fall 2004. The first survey was a general survey required by the University and collected numerical data. The second survey was specific to the course and was developed by the instructor to collect specific information from the students.

The first survey collected a large amount of numerical data and those related to the subject of this paper are shown in Table 1.

Table 1: Numerical results of the first survey.

Survey question	State-Space Course (Scale: 0-5)	Department (Scale: 0-5)	College (Scale: 0-5)	University (Scale: 0-5)
Enthusiasm for course material	4.82	4.28	4.49	4.51
Relates course material to current examples	4.73	4.03	4.35	4.44
Clearly explains complex concepts and ideas	4.91	3.77	4.16	4.18
Instructional materials used effectively	4.91	3.86	4.29	4.30
I found this class to be challenging	4.73	4.21	4.21	4.24

The first survey question in the table measures the enthusiasm about the course content. The second question includes the evaluation of practical design examples. The third question reflects the coverage of complex concepts and ideas. The fourth question includes the students' opinions on MATLAB, which is the major instructional tool in that course. The last question reflects the level of difficulty of this course. The evaluation for this course is better than those of the Department's, the College's and the University's. The evaluation was actually among the highest in that semester. This infers that a course with complex concepts can generate high enthusiasm among the students if integrated with a wide range of practical design problems and computational tools. Such a course can be rated highly and regarded as challenging by the students.

The second survey collected written comments that were specific to the course. The comments are summarized as follows:

- The students indicated that through the design problems the course materials became meaningful to them. Without the design problems and the use of MATLAB to solve them, the course would have been just a course on linear algebra, circuit analysis, dynamics, theorems and proofs. The design problems turned the course into a controls course.

- The students appreciated the effectiveness of MATLAB in helping them to solve the problems but also understood that the pencil and paper way was still needed. A number of them indicated that they preferred to have more coverage on MATLAB.
- All the students indicated that they learned a lot from this course and some indicated that the course was great and they enjoyed it. Some indicated that they had hard times in linear algebra and dynamics.
- The students thought that it would be better to require linear algebra as a prerequisite to this course, in addition to the classical control course.
- Half of the class thought that the course materials were difficult and the concepts were abstract but a few felt that they learned the materials comfortably.

V. Lessons Learned and Recommendation

Many students had hard times with matrix computations and linear algebra. Much time was used to cover these topics in the classroom. This caused several topics not being covered due to insufficient time. The topics omitted were linear quadratic regulator, kalman filtering, and linear quadratic Gaussian problems. The course will move along faster and smoother if all the students have taken linear algebra before taking this course.

MATLAB should be intensively used as a teaching tool in this course. The students would like to see the simulation results so that they could understand the working of the theory. For example, the effects of pole placement on the step response could be easily seen through the simulation.

In the modeling of mechanical systems as state-space models, the knowledge of dynamics was necessary. Many students had taken dynamics two years before they took this course. A short review on the equations of rotational motions in rectangular and polar coordinates will be helpful to the students.

VI. Concluding Remarks

The wide-range of practical control design problems significantly increased the students' interest in this course. The use of MATLAB facilitated their learning considerably. This resulted in very favorable student evaluation. The students appreciated the application of the state-space approach through solving the design problems. They also learned that a computer-aided control system design tool such as MATLAB was a must in tackling the state-space problems. This course also gave the students the opportunity to apply the knowledge such as circuit analysis and dynamics that they gained in previous courses to solve the problems. They came to know that some electrical engineering courses bore an interdisciplinary nature. The course evaluation results indicated that this new approach was well received by the students and the enrollment for this course was much increased.

Bibliography

- [1] C.T. Chen, *Linear System Theory and Design*, Oxford, 1999.
- [2] J. S. Bay, *Fundamentals of Linear State Space Systems*, WCB McGraw-Hill, 1999.
- [3] S.H. Zak, *Systems and Control*, Oxford, 2003
- [4] J.D. Aplevich, *The Essentials of Linear State-Space Systems*, John Wiley, 2000.
- [5] G. Franklin, et. al, *Feedback Control of Dynamic Systems*, 3rd Ed., Addison Wesley.
- [6] Dorf and Bishop, *Modern Control Systems*, 9th Ed., Prentice Hall.
- [7] Ogata, *Modern Control Engineering*, 3rd Ed., Prentice Hall.
- [8] Stefani, et al., *Design of Feedback Control Systems*, 4th Ed., Oxford.
- [9] Dorsey, *Continuous and Discrete Control Systems*, Mc Graw Hill.
- [10] Goodman, et al., *Control System Design*, Prentice Hall
- [11] C.H. Choi, "Enriching a modern control theory course with computer simulation," *Proceedings of the 1999 International Conference on Simulation in Engineering Education*, Society for Computer Simulation, pp. 77-82.
- [12] D. Frederick and J. Chow, *Feedback Control Problems using MATLAB and the Control System Toolbox*, Brooks/Cole Thomson Learning, 2000.
- [13] D. Hanselman and B. Littlefield, *Mastering MATLAB 6*, Prentice Hall, 2001.
- [14] J.B. Dabney and T.L. Harman, *Mastering SIMULINK*, Pearson Prentice Hall, 2004.

Biographical information

Dr. Choi received his Ph.D. degree in electrical and computer engineering from the University of California, Santa Barbara in 1988. He obtained several years of engineering experience in the industry before beginning his Ph.D. study. He is currently an Associate Professor in the Division of Engineering, University of North Florida. He has strong interest in undergraduate electrical engineering education. His teaching interests include automatic controls, microcontroller applications, digital system design, electromagnetics, signals and systems, circuit analysis, and others. His research interests include microcontroller-based system design, computational algorithms for controls, and control theory. Dr. Choi is a registered Professional Engineer in Florida. He could be reached at cchoi@unf.edu.