

# MATLAB and Simulink Modelling of a Full State Observer Controller of Armature Controlled and Field Controlled DC Motor using State Space Approach

Vedatroyee Ghosh<sup>1</sup>, Akash Sharma<sup>2</sup>, Viraj Bansal<sup>3</sup>

**Abstract:** In this paper, a dynamic model of DC motor using state space approach has been established. A state space observer model has been developed for the armature of a DC motor based on its torque, speed and electrical system equations and similarly flux control for field controlled DC motor. The system's controllability and observability is studied in each case and the performance of the DC motor is studied under various conditions and it's determined whether such a system with given parameters can be realized or not. A state feedback system is also determined using pole-placement method. The implementation of the systems and their verification is done using MATLAB and Simulink.

**Keywords:** State Space, DC Motor, MATLAB, Simulink, Pole placement, State Observer

## 1. Introduction

The main advantages of DC motor that makes it the most reliable energy conversion device is the fact that it can develop constant torque over a wide speed range and flexible volt-ampere characteristic. DC motors have the easiest speed control methods hence they are with time, being preferred over AC motors. Owing to the easiness, there are multiple applications that involve controlling the speed, current and position characteristics. Similarly, flux control exercises its importance in the fact that it also deals with speed and position control which ultimately makes DC motor one of the most versatile and easy to work with devices. [1]

State-space approach or state-space modelling helps us to determine a well-defined relationship between the input and output variables which are all physical variables. Hence with state-space modelling, a physical feedback can be established with the help of state variables, the implementation of any system with physical variables becomes straightforward, and the solution of the state equations tells us the behaviour of the system parameters with respect to time. [2]

In this paper, we have implemented a complete state observer model with the help of pole-placement method with which we have predicted the feasibility of each system and whether such a system can be realized or not. With this paper, we have aimed at controlling the speed, position and current of an armature controlled DC motor and flux control through either speed control. With pole-placement method and using Ackermann's formula, the system is made to be controllable and observable at user-defined poles after the controllability and observability has been checked. With this we find K and Ke matrices which facilitate the implementation of our system as a full-state observer controller in the Simulink environment. [3]-[6]

## 2. Armature Controlled Dc Motor

For the controller-observer designs, we have considered the following motor parameters:

- Armature resistance,  $R_a = 1.1 \Omega$

- Armature inductance,  $L_a = 0.00891 \text{ H}$
- Moment of inertia of the motor and load,  $J = 0.0044 \text{ N-m-s}^2/\text{rad}$
- Frictional coefficient of motor and load,  $B = 0.011 \text{ N-m/(rad/sec)}$
- Back emf constant,  $K_b = 0.22 \text{ V/(rad/sec)}$
- Torque constant,  $K_t = 0.22 \text{ N-m/A}$

## 3. Position Control

The different equations related to position control of DC motor are as follows:

$$i_a R_a + L_a \frac{di_a}{dt} + e_b = v_a \quad (1)$$

$$\text{Torque, } T = K_t i_a \quad (2)$$

$$J \frac{d^2\theta}{dt^2} + B \frac{d\theta}{dt} = T \quad (3)$$

$$e_b = K_b \frac{d\theta}{dt} \quad (4)$$

Assuming the state variables as

$$x_1 = i_a, x_2 = w = \frac{d\theta}{dt} \text{ and } x_3 = \theta$$

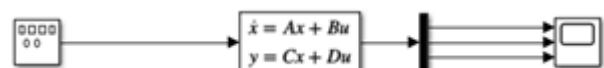
We get the following state matrix:

$$\begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \frac{dx_3}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_b}{L_a} & 0 \\ \frac{K_t}{J} & -\frac{B}{J} & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} \\ 0 \\ 0 \end{bmatrix} [u]$$

$$\text{For } [y] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

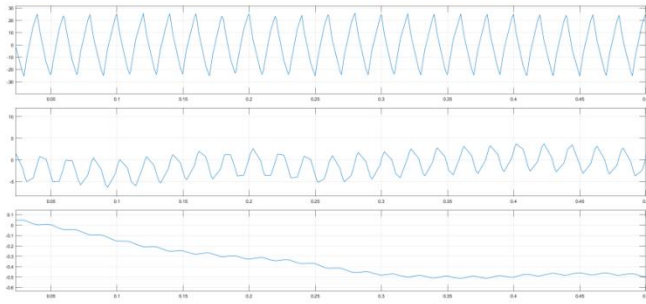
We have the complete control of the system.

With Initial conditions: 0.1 10 0



**Figure 1:** Simulink diagram to simulate motor dynamics

Input: Square Wave of Amplitude 50 V and frequency 50 Hz.



**Figure 2:** Angular Current, Angular Position of Shaft and Angular Velocity of Shaft

For the position control of the system:

$$[y] = [0 \ 0 \ 1] \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix}$$

Hence we have the following equations:

$$A = \begin{bmatrix} -123.5 & -24.69 & 0 \\ 50 & 2.5 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 112.2 \\ 0 \\ 0 \end{bmatrix}$$

$$C = [0 \ 0 \ 1]$$

$$D = [0]$$

#### a) State Feedback Gain Matrix K Design

Let us chose the control signal to be

$$u = -Kx$$

This is the feedback signal which is obtained from the state feedback and is determined by an instantaneous rate. This is called state feedback system where the K matrix which is a  $1 \times n$  matrix, is called the state feedback gain matrix, with the which we can force the system to have closed loop poles at the desired locations, provided that the original system is completely state controllable. So the first step lies in determining the controllability of the system.

#### i) Controllability Test

To check the controllability condition of the system. This can be done in MATLAB with the following code:

```
Qc = ctrb(A,B);
rankQc = rank(Qc);
disp('Controllable Matrix is Qc = ');
disp(Qc);
if(rankQc == 3)
disp('Given System is Controllable.');
```

With this, for our system we get the output: Given System is Controllable.

Hence we can proceed.

#### ii) State Feedback Gain Matrix K Design Using MATLAB

Further using the Ackermann's formula in the MATLAB environment, we can find the gain matrix K, where  $K = [K1 \ K2 \ K3]$ .

Let us place the closed loop controller poles at  $-100+100i$ ,  $-100-100i$  and  $-200$

We enter the following code in the Command window:

```
J=[-100+100i -100-100i -200];
K = acker(A, B, J)
or
K= place(A, B, J)
And the output matrix we get is:
K = [2.4417  10.2949  712.8]
```

#### b) Full Order State Observer Gain Matrix Ke Design Using MATLAB

Using pole placement by state feedback method requires the feedback of all state variables. In real life scenario, however, not all state variables are available for feedback and measurement. In these cases or situations, the measurement of unmeasurable states needs to be done. The method of doing the same is called Observation. A state observer is a device that observes or estimates the state variables. A full order state observer is one that observes all the state variables in a given system.

The mathematical model of the observer can be given by:

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}).$$

$$\hat{y} = C\hat{x} + Du,$$

And the observer error satisfies the equation:

$$\dot{e} = (A - LC)e.$$

Here we take  $L = Ke$ , Observer gain matrix which is a  $n \times 1$  matrix. Here it is of the form:

$$Ke = \begin{bmatrix} Ke1 \\ Ke2 \\ Ke3 \end{bmatrix}$$

If it is possible to prove that the system is completely observable, then we can chose the observable gain matrix Ke, such that  $(A - KeC)$  has desired eigen values according to the chosen observer poles.

#### i) Observability Test

To check if the given system is observable or not, the following code can be typed in the command window of MATLAB:

```
Qb = obsv(A, C);
rankQb = rank(Qb);
disp('Observable Matrix is Qb = ');
disp(Qb);
if(rankQb == 3)
disp('Given System is Observable.');
```

We get the output as: Given System is Observable.

So, we proceed to the next step.

#### ii) State Observer Gain Matrix Ke design Using MATLAB

Using Ackermann's formula, we can determine the Ke matrix with the following code when we decide to place the observer poles at:  $-50-50i$ ,  $-50+50i$  and  $-50$ .

```
L = [-50-50i; -50+50i; -50];
Ke = acker(A.', C.', L)
Ket = Ke.'
```

Output is as follows:

$$K_{et} = \begin{bmatrix} -9145.5 \\ 5428.4 \\ 24 \end{bmatrix}$$

### c) Modified Observer based Controller

Now, we know that the modified observer based controller is of the following form:

$$AA = A - K_{et} * C - B * K$$

$$BB = K_{et}$$

$$CC = K$$

$$DD = [0; 0; 0]$$

So, we have the following modified system parameters:

$$AA = \begin{bmatrix} -398 & -1180 & -70855 \\ 50 & -2 & -5428 \\ 0 & 1 & -24 \end{bmatrix}$$

$$BB = \begin{bmatrix} -9145.5 \\ 5428.4 \\ 24 \end{bmatrix}$$

$$CC = [2.4417 \quad 10.2949 \quad 712.8]$$

$$DD = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

### d) Response to initial conditions with designed K and Ke values

In this section, we have derived the response of the system to the given initial conditions;  $x_1(0) = 1$ ,  $x_2(0) = 0$ ,  $x_3(0) = 0$ ,  $e_1(0) = 1$ ,  $e_2(0) = 0$  and  $e_3(0) = 0$ . The following program is used to get the response curves for the system:

$$AA = [A - B * K, B * K; \text{zeros}(3,3), A - K_{et} * C];$$

$$\text{Sys} = \text{ss}(AA, \text{eye}(6), \text{eye}(6), \text{eye}(6));$$

$$t = 0:0.01:5;$$

$$x_0 = [1; 0; 1; 0; 1; 0];$$

$$x = \text{initial}(\text{Sys}, x_0, t);$$

$$x_1 = [1 \ 0 \ 0 \ 0 \ 0] * x';$$

$$x_2 = [0 \ 1 \ 0 \ 0 \ 0] * x';$$

$$x_3 = [0 \ 0 \ 1 \ 0 \ 0] * x';$$

$$e_1 = [0 \ 0 \ 1 \ 0 \ 0] * x';$$

$$e_2 = [0 \ 0 \ 0 \ 1 \ 0] * x';$$

$$e_3 = [0 \ 0 \ 0 \ 0 \ 1] * x';$$

$$\text{subplot}(3,3,1);$$

$$\text{plot}(t, x_1); \text{grid}$$

$$\text{xlabel}('t(\text{sec})'), \text{ylabel}('x_1')$$

$$\text{subplot}(3,3,2);$$

$$\text{plot}(t, x_2); \text{grid}$$

$$\text{xlabel}('t(\text{sec})'), \text{ylabel}('x_2')$$

$$\text{subplot}(3,3,3);$$

$$\text{plot}(t, x_3); \text{grid}$$

$$\text{xlabel}('t(\text{sec})'), \text{ylabel}('x_3')$$

$$\text{subplot}(3,3,4);$$

$$\text{plot}(t, e_1); \text{grid}$$

$$\text{xlabel}('t(\text{sec})'), \text{ylabel}('e_1')$$

$$\text{subplot}(3,3,5);$$

$$\text{plot}(t, e_2); \text{grid}$$

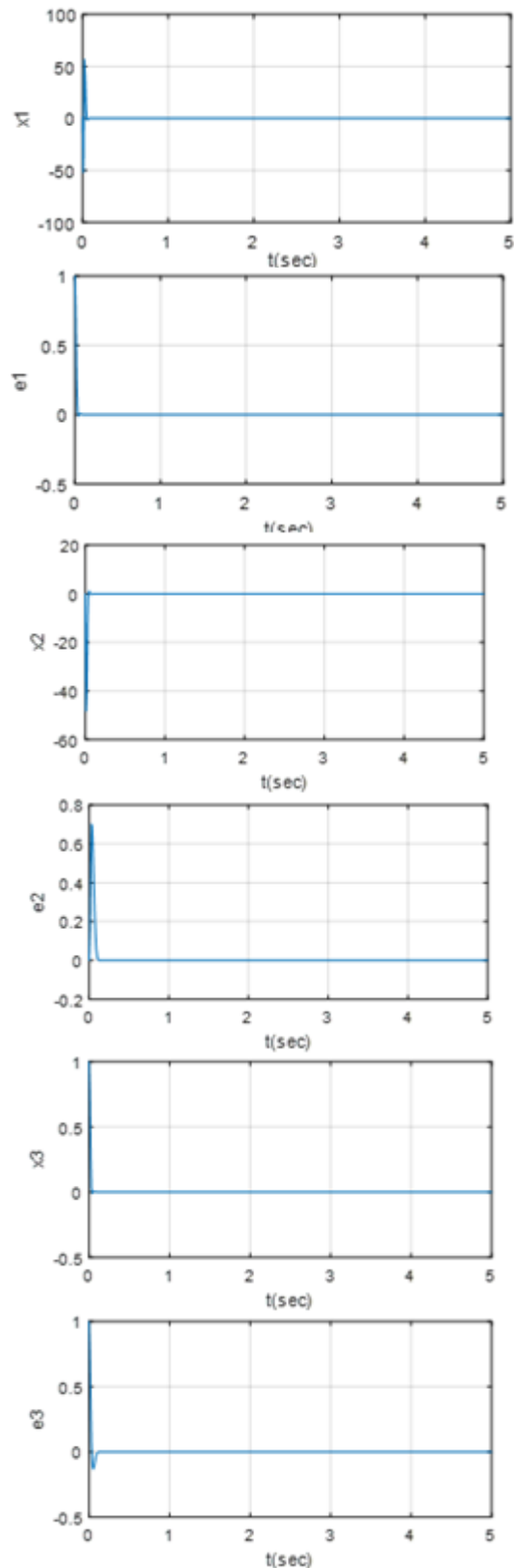
$$\text{xlabel}('t(\text{sec})'), \text{ylabel}('e_2')$$

$$\text{subplot}(3,3,6);$$

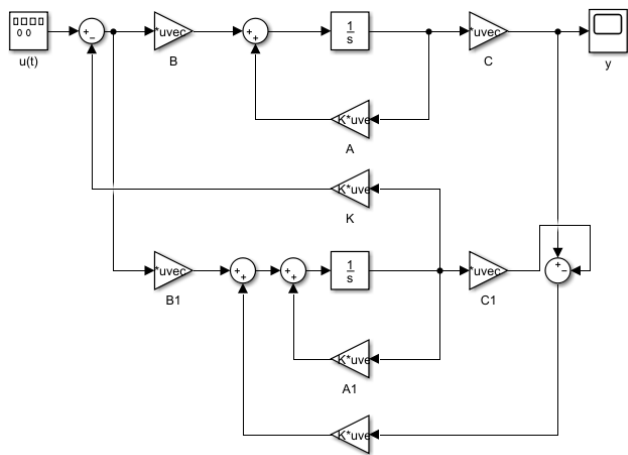
$$\text{plot}(t, e_3); \text{grid}$$

$$\text{xlabel}('t(\text{sec})'), \text{ylabel}('e_3')$$

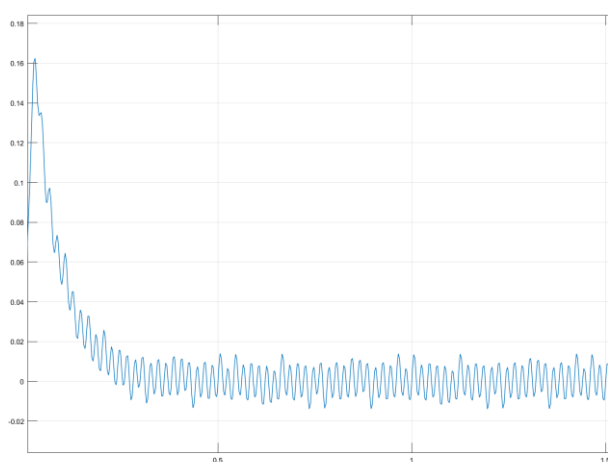
Output:



e) Simulink Model



The Simulink model has been developed with the given AA, BB, CC and DD matrices and the complete state observer control model is simulated. The output graph is as follows:



**f) Result**

The input signal generator block generates the signal u(t) which aids as the excitation voltage for the motor, modelled by the defined state-space block which produces the state y as its output. The system output y is then studied, and as we can see, the system stabilizes to 0 after certain time showing that the output feedback controller as designed, ensures the stability of the system. With a different input value, we can stabilize the system at another value.

**I. Speed Control**

As we know, how important speed control is to a DC motor system, it becomes all the more necessary for us to control its levels and its response.

For speed and current control, we have the same DC motor equations as previously stated.

Assuming the state variables as

$$x1 = ia \text{ and } x2 = w = \frac{d\theta}{dt}$$

$$A = \begin{bmatrix} -\frac{Ra}{L\alpha} & -\frac{Kb}{L\alpha} \\ \frac{Kt}{J} & -\frac{B}{J} \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{1}{L\alpha} \\ 0 \end{bmatrix}$$

$$C = [0 \ 1]$$

$$D = [0]$$

**a) State Feedback Gain Matrix K Design**

With the system model being found to be controllable, we find the K matrix in a similar manner as previously done at the poles -100 and -100 as follows:

$$K = [0.6597 \ 1.4740]$$

**b) Full Order State Observer Gain Matrix Ke Design Using MATLAB**

With the code as abovesaid, we find that the system model is observable. We can find the Ke, i.e. observer gain matrix at the poles -50+50i and -50-50i.

$$Ke = \begin{bmatrix} -25.9568 \\ -142.6281 \end{bmatrix}$$

**c) Modified Observer based Controller**

The modified observer based controller system is found to be:

$$AA = \begin{bmatrix} -171.5432 & -190.1250 \\ 192.6281 & -2.5 \end{bmatrix}$$

$$BB = \begin{bmatrix} -25.9568 \\ -142.6281 \end{bmatrix}$$

$$CC = [0.6597 \ 1.4740]$$

$$DD = [0]$$

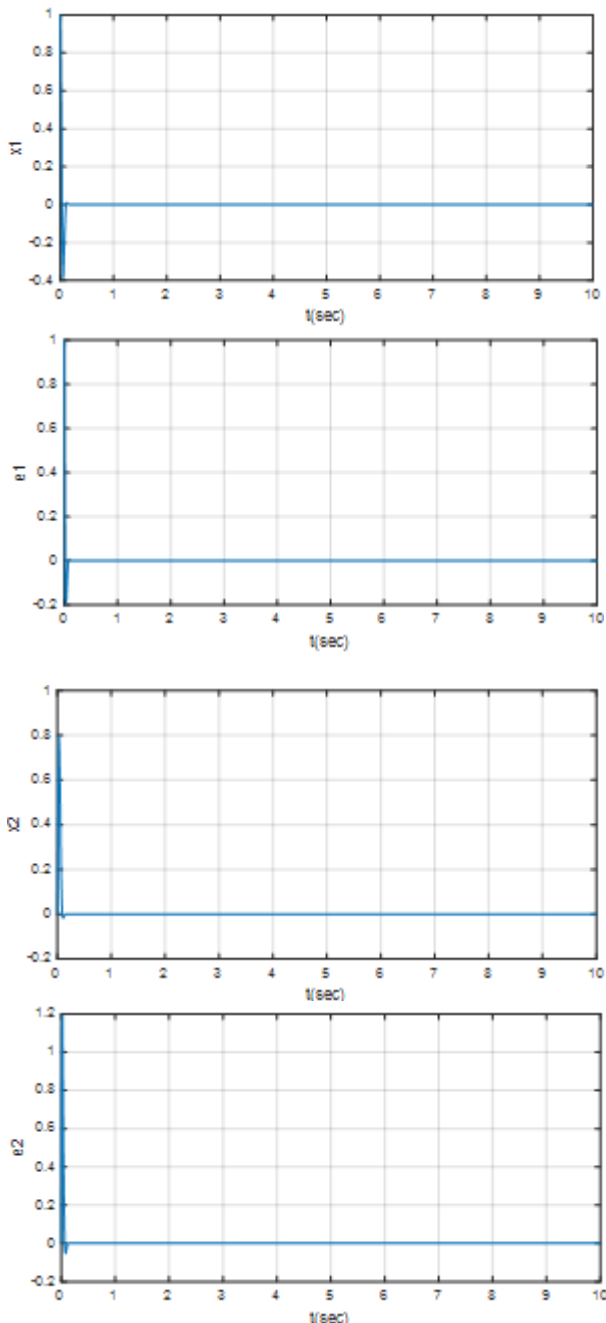
**d) Response to initial conditions with designed K and Ke values**

For this system, the code will be as follows:

```

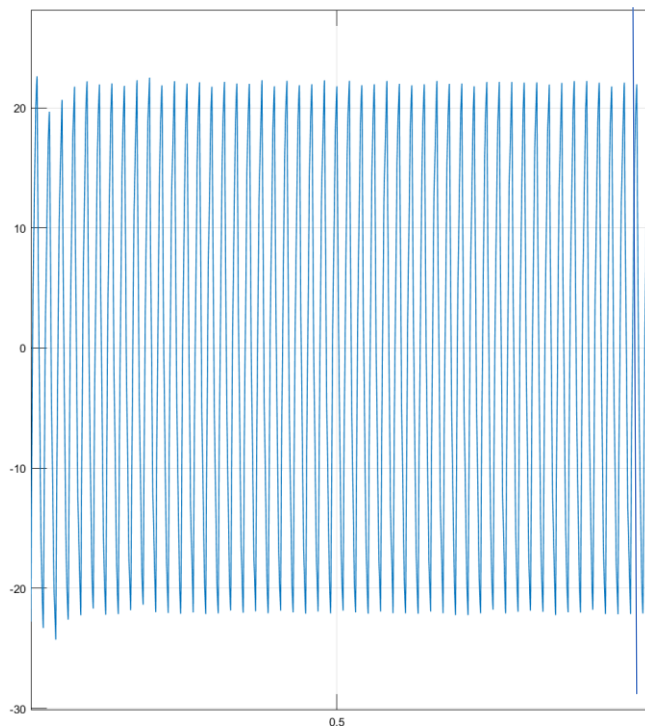
AAs = [A-Bs*Kp, Bs*Ks; zeros(2,2), A-Kets*Cs];
Sys = ss(AAs,eye(4),eye(4),eye(4));
t=0:0.01:10;
x0=[1;0;1;0];
x=initial(Sys,x0,t);
x1= [1 0 0 0]*x';
x2= [0 1 0 0]*x';
e1= [0 0 1 0]*x';
e2= [0 0 0 1]*x';
figure
subplot(2,2,1);
plot(t,x1);grid
xlabel('t(sec)'), ylabel('x1')
subplot (2,2,2);
plot(t,x2);grid
xlabel('t(sec)'), ylabel('x2')
subplot(2,2,3);
plot(t,e1);grid
xlabel('t(sec)'), ylabel('e1')
subplot (2,2,4);
plot(t,e2);grid
xlabel('t(sec)'), ylabel('e2')
    
```

The output is as follows:



**e) Simulink output**

The Simulink model is the same as that of the one in position control of armature controlled DC motor. The output can be seen below.



**f) Result**

We can see that the system model has ripples and isn't attaining a constant state. The system is oscillating between 20.2A to -20.2A, 0A being the initial state of the system. Hence the system doesn't attain a constant state and to control it's value, we have to place poles for the controller and observer design at different values.

**Current Control**

Current control becomes necessary as the torque of a DC motor can be controlled by controlling the current, as we know from the equation:  $T=Kia$ . Hence here, we have aimed at controlling the speed through state space approach. The model equations are the same as that of speed control, however as we aim at controlling the speed, the C matrix becomes:

$$C=[1 \ 0]$$

**a) State Feedback Gain Matrix K Design**

We find the K matrix in a similar manner as previously done at the poles  $-100+100i$   $-100-100i$  as follows:

$$K=[0.6597 \ 3.2560]$$

**b) Full Order State Observer Gain Matrix Ke Design Using MATLAB**

With the code as previously used, we find that the system model is observable. We can find the Ke, i.e. observer gain matrix at the poles  $-50$  and  $-50$ .

$$Ke= \begin{bmatrix} 83.2266 \\ -25.9568 \end{bmatrix}$$

**c) Modified Observer based Controller**

The modified observer based controller system is found to be (depending on the following equations):

$$AA = A-Ke^t * C-B * K$$

$$BB = Ke^t$$

$$CC = K$$

DD = 0

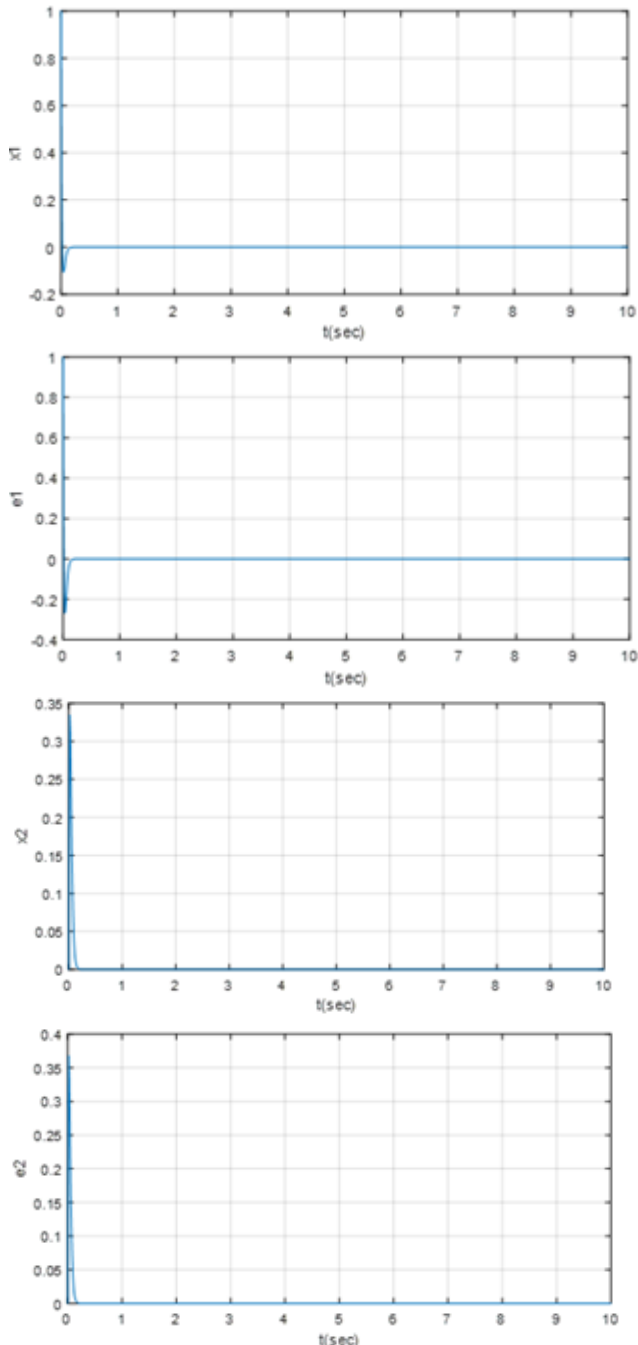
$$AA = \begin{bmatrix} -197.5 & -473.3516 \\ 50 & 23.4568 \end{bmatrix}$$

$$BB = \begin{bmatrix} 83.66 \\ -25.9568 \end{bmatrix}$$

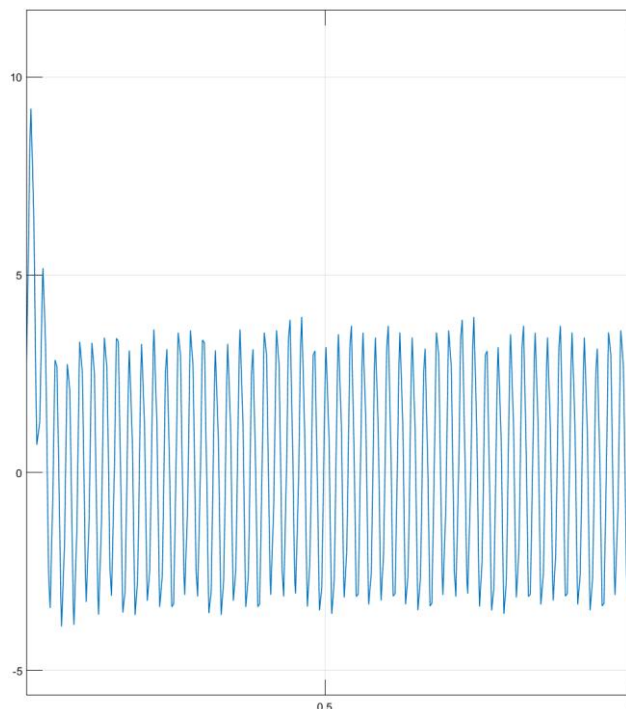
$$CC = [0.65973 \ 2.2560]$$

$$DD = [0]$$

d) Response to initial conditions with designed K and Ke values



e) Simulink Output



f) Result

In this output, we can see the system is not stable and is oscillating in a sinusoidal fashion, hence the output cannot be exactly controlled, so the system poles have to be changed for the system to stabilize at a particular value.

#### 4. Field Controlled Dc Motor

The flux control of field controlled DC motor is very important to the functioning of a DC motor and controlling it externally without complicated circuitry. The speed of a dc motor, as we know, is inversely proportional to the flux per pole and hence by decreasing the flux, speed can be increased and vice versa. Hence speed control of the same becomes an integral part of flux control, hence we have implemented it in the paper.

For the controller-observer designs, we have considered the following motor parameters:

- Armature resistance,  $R_f = 24 \Omega$
- Armature inductance,  $L_f = 0.01 \text{ H}$
- Moment of inertia of the motor and load,  $J = 0.0044 \text{ N-m-s}^2/\text{rad}$
- Frictional coefficient of motor and load,  $B = 0.011 \text{ N-m}/(\text{rad}/\text{sec})$
- Back emf constant,  $K_b = 0.22 \text{ V}/(\text{rad}/\text{sec})$
- Torque constant,  $K_t = 0.22 \text{ N-m}/\text{A}$
- Load Torque,  $T_l = 10 \text{ N-m}$

The equations related to the speed control of DC motor are given below:

$$i_f R_f + L_f \frac{di_f}{dt} = v_f \tag{4}$$

$$\text{Torque, } T = K_t i_f \tag{5}$$

$$J \frac{d^2\theta}{dt^2} + B \frac{d\theta}{dt} = T \tag{6}$$

Assuming the state variables as

$$x1 = ia \text{ and } x2 = w = \frac{d\theta}{dt}$$

we have the following state matrix:

$$\begin{bmatrix} \frac{dx1}{dt} \\ \frac{dx2}{dt} \end{bmatrix} = \begin{bmatrix} -\frac{Ra}{L_a} & -\frac{K_b}{L_a} \\ \frac{K_t}{J} & -\frac{B}{J} \end{bmatrix} \begin{bmatrix} x1 \\ x2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} \\ 0 \end{bmatrix} [u]$$

For  $[y] = [0 \ 1] \begin{bmatrix} x1 \\ x2 \end{bmatrix}$

So,

$$A = \begin{bmatrix} -24000 & 0 \\ 50 & -2 \end{bmatrix} B = \begin{bmatrix} 100 \\ 0 \end{bmatrix}$$

$$C = [0 \ 1] \text{ and } D = [0]$$

**a) State Feedback Gain Matrix K Design**

After finding out that our system is controllable through the MATLAB code, we find the K matrix in a similar manner as previously done at the poles -100+100i -100-100i:

$$K = [-238.0250 \ 3.9013]$$

**b) Full Order State Observer Gain Matrix Ke Design Using MATLAB**

With the previously used code, we find that the system model is observable, hence we can find the Ke, i.e. observer gain matrix at the poles -50+50i and -50-50i.

$$Ke = 1.0e+07 *$$

$$\begin{bmatrix} 1.1424 \\ -0.0024 \end{bmatrix}$$

**c) Modified Observer based Controller**

The modified observer based controller system based on the controller gain matrix and observer gain matrix is found with the following model formulae:

$$AA = A - Ke * C - B * K$$

$$BB = Ke$$

$$CC = K$$

$$DD = 0$$

The final system is found to be:

$$AA = 1.0e+07 * 0$$

$$\begin{bmatrix} -0.0000 & -1.1425 \\ 0.0000 & 0.0024 \end{bmatrix}$$

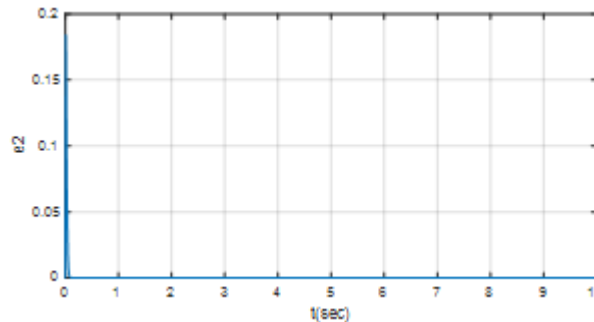
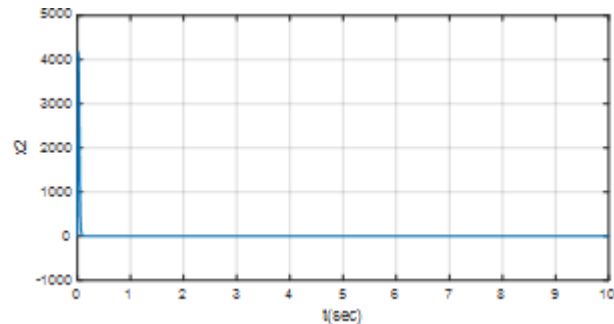
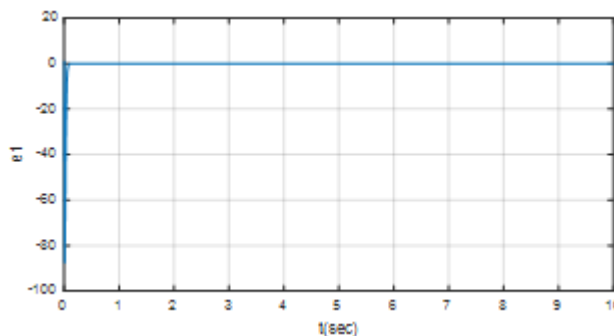
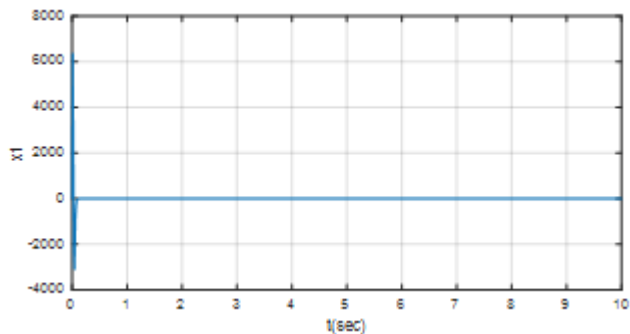
$$BB = 1.0e+07 *$$

$$\begin{bmatrix} 1.1424 \\ -0.0024 \end{bmatrix}$$

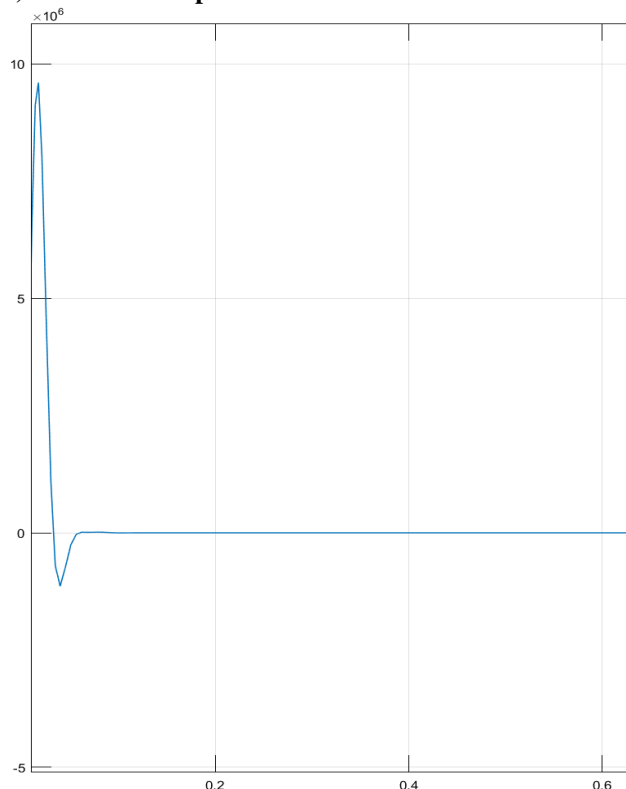
$$CC = [-238.0250 \ 3.9013]$$

$$DD = [0]$$

**d) Response to initial conditions with designed K and Ke values**



**e) Simulink Output**

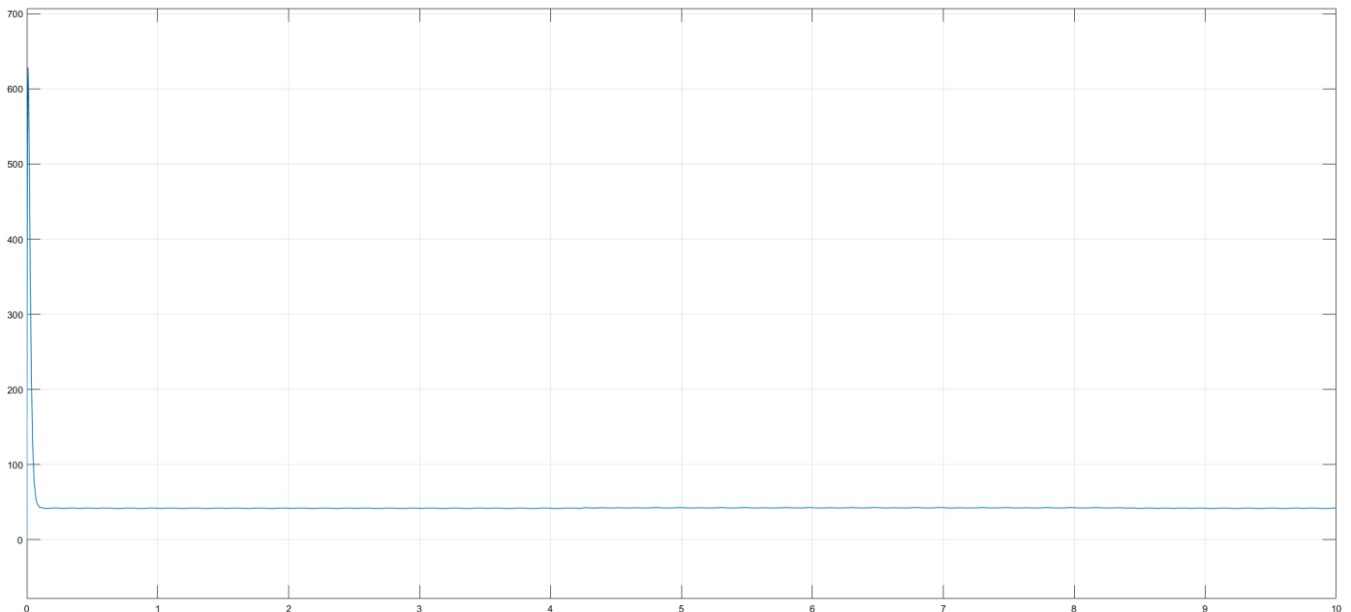


**f) Result**

The Simulink output shows us that the system has a ripple initially, but is eventually stabilized at a value of zero, and hence the output feedback controller is able to stabilize the system that we have considered here. Furthermore, the system can be stabilized at a value other than zero,

according to the input we feed into the system. In the simulation model, adding a constant value, instead of 0, the system can be stabilized at a reference value.

We can see it in the following output obtained:

**5. Conclusion**

The state-space model of dc motor, both armature-controlled and field-controlled has been developed and simulation has been carried out for all the proposed systems.  $K$  and  $K_{egain}$  matrices have been determined using MATLAB and the initial response of all the systems have been plotted. However, as previously mentioned, the controller gain matrix, i.e.  $K$  is not unique for a particular system, but it depends on the user-defined or desired closed loop pole location which further helps us to determine the speed, position, current and damping of the response. To address the problem of the unmeasurable states,  $K_e$ , i.e. observer gain, acts as a correction signal to the plant model. If substantial unknowns are involved, the feedback signal given to the matrix  $K_e$  should be relatively large. Further, as we examine the performance of the various systems found, we see that each system has a different output feedback response and not every system at the desired poles can be stabilized to a particular value. However, the stabilizable ones provide satisfactory estimates even in the presence of external noise input, or torque disturbances and with varied initial conditions.

Taylor and Francis group Boca Raton London New York 2010.

- [5] NagoorKani, "Advanced Control Theory", RBA Publications.  
 [6] T. Kailath, Linear Systems, Prentice Hall, 1980.

**References**

- [1] Dr. P.S. Bimbhra, "Electrical Machines", Khanna Publisher.  
 [2] Luenberger D. G., "An introduction to observers," IEEE Transaction on Automatic Control, 1971; AC-16: 596-602.  
 [3] Katsuhiko Ogata, "Modern Control Engineering", Pearson Publishers.  
 [4] Devendra K. Chaturvedi, "Modeling and simulation of system using MATLAB and Simulink", CRC Press

**Volume 8 Issue 4, April 2019**

[www.ijsr.net](http://www.ijsr.net)

Licensed Under Creative Commons Attribution CC BY