T H R E E

Modeling   
 in the Time Domain

SOLUTIONS TO CASE STUDIES CHALLENGES

**Antenna Control: State-Space Representation**

14. • Sp 3-12 (Chapter objective problem. Finding the state-space representation for each subsystem of a positional control system)

For the power amplifier, = . Taking the inverse Laplace transform, +150ea =

150vp. Thus, the state equation is



For the motor and load, define the state variables as x1 **=** m and x2 = m. Therefore,

1 = x2 (1)

Using the transfer function of the motor, cross multiplying, taking the inverse Laplace transform,

and using the definitions for the state variables,

2 = - x2 + ea (2)

Using the gear ratio, the output equation is

y = 0.2x1 (3)

Also, Jm = Ja+5()2 = 0.05+0.2 = 0.25, Dm = Da+3()2 = 0.01+0.12 = 0.13, =

= 0.8, and = 1.32. Using Eqs. (1), (2), and (3) along with the previous values, the

state and output equations are,



**Aquifer:** **State-Space** **Representation**

C1= qi1-qo1+q2-q1+q21 = qi1-0+G2(h2-h1)-G1h1+G21(H1-h1) =

-(G2+G1+G21)h1+G2h2+qi1+G21H1

C2= qi2-q02+q3-q2+q32 = qi2-qo2+G3(h3-h2)-G2(h2-h1)+0 = G2h1-[G2+G3]h2+G3h3+qi2-qo2

C3= qi3-qo3+q4-q3+q43 = qi3-qo3+0-G3(h3-h2)+0 = G3h2-G3h3+qi3-qo3

Dividing each equation by Ci and defining the state vector as **x** = [h1 h2 h3]T



where u(t) = unit step function.

Answers to Review Questions

**1.** (1) Can model systems other than linear, constant coefficients; (2) Used for digital simulation

**2.** Yields qualitative insight

**3.** That smallest set of variables that completely describe the system

**4.** The value of the state variables

**5.** The vector whose components are the state variables

**6.** The n-dimensional space whose bases are the state variables

**7.** State equations, an output equation, and an initial state vector (initial conditions)

**8.** Eight

**9.** Forms linear combinations of the state variables and the input to form the desired output

**10.** No variable in the set can be written as a linear sum of the other variables in the set.

**11.** (1) They must be linearly independent; (2) The number of state variables must agree with the order of the differential equation describing the system; (3) The degree of difficulty in obtaining the state equations for a given set of state variables.

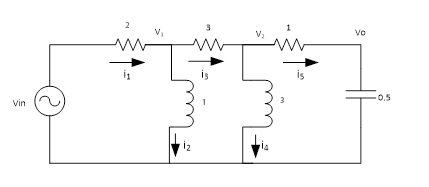
**12.** The variables that are being differentiated in each of the linearly independent energy storage elements

**13.** Yes, depending upon the choice of circuit variables and technique used to write the system equations. For example, a three -loop problem with three energy storage elements could yield three simultaneous second-order differential equations which would then be described by six, first-order differential equations. This exact situation arose when we wrote the differential equations for mechanical systems and then proceeded to find the state equations.

**14.** The state variables are successive derivatives.

SOLUTIONS to Problems

**1.**



The state variables are , and .

We have that , and .

Applying KVL around the external loop one gets

And in each one of the nodes

and

Substituting

or

Solving for one gets

Thus

Also

and

Finally

In matrix form we have

**2.**

Let C1 be the grounded capacitor and C 2 be the other. Now, writing the equations for the energy storage components yields,

 (1)

Thus the state vector is .

Now, find the three loop currents in terms of the state variables and the input.

Writing KVL around Loop 2 yields .

Or, 

Writing KVL around the outer loop yields 

Or,



Also, . Hence, 

Substituting the loop currents in equations (1) yields the results in vector-matrix form,



Since , the output equation is



**3.**

Equations of motion in Laplace:



2

Equations of motion in the time domain:



Define state variables:



Substituting Eq. (1) in (2), (3) in (4), and (5) in (6), we obtain, respectively:



Substituting Eqs. (1) through (6) into the equations of motion in the time domain and solving for the derivatives of the state variables and using Eqs. (7) through (9) yields the state equations:



The output is *x*1 = *z*1.

In vector-matrix form:



**4.**

The impedance equations are:

|  |
| --- |
|  |
|  |
|  |

Taking the inverse Laplace transform

|  |
| --- |
|  |
|  |
|  |

|  |
| --- |
|  |
|  |
|  |

Define the state variables

; ; ; ; ;

The equations are rewritten as

In matrix form





**5.**

**4 T**



**50**

**1600**

Writing the equations of motion,



Taking the inverse Laplace transform and simplifying,



Defining the state variables as



Writing the state equations using the equations of motion and the definitions of the state variables



In vector-matrix form,



**6.**

Drawing the equivalent circuit,



10T

(1/10)(102 ) = 10 N-m/rad

200(1/10)2 =2 N-m/rad

Writing the equations of motion,



Taking the inverse Laplace transform,

 (1)

 (2)

 (3)

From (3),

 and  (4)

assuming zero initial conditions.

From (1)

 (5)

Substituting (4) and (5) into (2) yields the state equation (notice there is only one equation),



The output equation is given by,



**7.**

Solving Eqs. (3.44) and (3.45) in the text for the transfer functions  and :

 and 

Thus,



and



Multiplying each of the above transfer functions by *s* to find velocity yields pole/zero cancellation at the origin and a resulting transfer function that is third order.

**8.**

**a. .** Using the standard form derived in the textbook,



**b.** Using the standard form derived in the textbook,



**9.**

**Program:**

'a'

num=100;

den=[1 20 10 7 100];

G=tf(num,den)

[Acc,Bcc,Ccc,Dcc]=tf2ss(num,den);

Af=flipud(Acc);

A=fliplr(Af)

B=flipud(Bcc)

C=fliplr(Ccc)

'b'

num=30;

den=[1 8 9 6 1 30];

G=tf(num,den)

[Acc,Bcc,Ccc,Dcc]=tf2ss(num,den);

Af=flipud(Acc);

A=fliplr(Af)

B=flipud(Bcc)

C=fliplr(Ccc)

**Computer response:**

ans =

a

Transfer function:

100

---------------------------------

s^4 + 20 s^3 + 10 s^2 + 7 s + 100

A =

0 1 0 0

0 0 1 0

0 0 0 1

-100 -7 -10 -20

B =

0

0

0

1

C =

100 0 0 0

ans =

b

Transfer function:

30

------------------------------------

s^5 + 8 s^4 + 9 s^3 + 6 s^2 + s + 30

A =

0 1 0 0 0

0 0 1 0 0

0 0 0 1 0

0 0 0 0 1

-30 -1 -6 -9 -8

B =

0

0

0

0

1

C =

30 0 0 0 0

**10.**

**a.** Using the standard form derived in the textbook,



**b.** Using the standard form derived in the textbook,

22

**11.**

**Program:**

'a'

num=[8 10];

den=[1 5 1 5 13]

G=tf(num,den)

[Acc,Bcc,Ccc,Dcc]=tf2ss(num,den);

Af=flipud(Acc);

A=fliplr(Af)

B=flipud(Bcc)

C=fliplr(Ccc)

'b'

num=[1 2 12 7 6];

den=[1 9 13 8 0 0]

G=tf(num,den)

[Acc,Bcc,Ccc,Dcc]=tf2ss(num,den);

Af=flipud(Acc);

A=fliplr(Af)

B=flipud(Bcc)

C=fliplr(Ccc)

**Computer response:**

ans =

ans =

a

den =

1 5 1 5 13

Transfer function:

8 s + 10

----------------------------

s^4 + 5 s^3 + s^2 + 5 s + 13

A =

0 1 0 0

0 0 1 0

0 0 0 1

-13 -5 -1 -5

B =

0

0

0

1

C =

10 8 0 0

ans =

b

den =

1 9 13 8 0 0

Transfer function:

s^4 + 2 s^3 + 12 s^2 + 7 s + 6

------------------------------

s^5 + 9 s^4 + 13 s^3 + 8 s^2

A =

0 1 0 0 0

0 0 1 0 0

0 0 0 1 0

0 0 0 0 1

 0 0 -8 -13 -9

B =

0

0

0

0

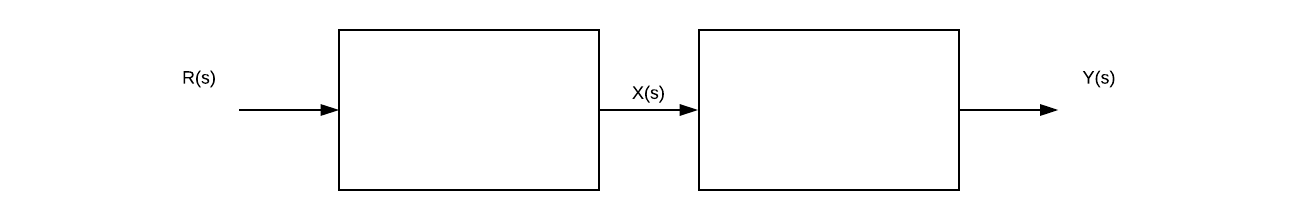
1

C =

6 7 12 2 1

**12.**

The transfer function can be represented as a block diagram as follows:



The differential equation of the first box is

Define the state variables

; ;

so

From the second box

In vector-matrix form

**13.**

**a.** G(s)=**C**(s**I**-**A**)-1**B**

**; ;**

Therefore, . Note that in this case the result could have also been obtained by inspection.

**b.** G(s)=**C**(s**I**-**A**)-1**B**

**; ;**

Therefore .

**c.** G(s)=**C**(s**I**-**A**)-1**B**

**; ;**

Therefore .

**14.**

**Program:**

'a'

A=[0 1 5 0;0 0 1 0;0 0 0 1;-7 -9 -2 -3];

B=[0;5;8;2];

C=[1 3 6 6];

D=0;

statespace=ss(A,B,C,D)

[num,den]=ss2tf(A,B,C,D);

G=tf(num,den)

'b'

A=[3 1 0 4 -2;-3 5 -5 2 -1;0 1 -1 2 8;-7 6 -3 -4 0;-6 0 4 -3 1];

B=[2;7;8;5;4];

C=[1 -2 -9 7 6];

D=0;

statespace=ss(A,B,C,D)

[num,den]=ss2tf(A,B,C,D);

G=tf(num,den)

**Computer response:**

ans =

a

a =

x1 x2 x3 x4

x1 0 1 5 0

x2 0 0 1 0

x3 0 0 0 1

x4 -7 -9 -2 -3

b =

u1

x1 0

x2 5

x3 8

x4 2

c =

x1 x2 x3 x4

y1 1 3 6 6

d =

u1

y1 0

Continuous-time model.

Transfer function:

75 s^3 - 96 s^2 - 2331 s - 210

------------------------------

s^4 + 3 s^3 + 2 s^2 + 44 s + 7

ans =

b

a =

x1 x2 x3 x4 x5

x1 3 1 0 4 -2

x2 -3 5 -5 2 -1

x3 0 1 -1 2 8

x4 -7 6 -3 -4 0

x5 -6 0 4 -3 1

b =

u1

x1 2

x2 7

x3 8

x4 5

x5 4

c =

x1 x2 x3 x4 x5

y1 1 -2 -9 7 6

d =

u1

y1 0

Continuous-time model.

Transfer function:

-25 s^4 - 292 s^3 + 1680 s^2 + 1.628e004 s + 3.188e004

------------------------------------------------------

s^5 - 4 s^4 - 32 s^3 + 148 s^2 - 1153 s - 4480

**15.**

**Program:**

syms s

'a'

A=[0 1 5 0

0 0 1 0

0 0 0 1

-7 -9 -2 -3];

B=[0;5;8;2];

C=[1 3 4 6];

D=0;

I=[1 0 0 0

0 1 0 0

0 0 1 0

0 0 0 1];

'T(s)'

T=C\*((s\*I-A)^-1)\*B+D;

T=simple(T);

pretty(T)

'b'

A=[3 1 0 4 -2

-3 5 -5 2 -1

0 1 -1 2 8

-7 6 -3 -4 0

-6 0 4 -3 1];

B=[2;7;6;5;4];

C=[1 -2 -9 7 6];

D=0;

I=[1 0 0 0 0

0 1 0 0 0

0 0 1 0 0

0 0 0 1 0

0 0 0 0 1];

'T(s)'

T=C\*((s\*I-A)^-1)\*B+D;

T=simple(T);

pretty(T)

**Computer response:**

ans =

a

ans =

T(s)

3 2

59 s - 148 s - 2241 s - 140

-----------------------------

4 3 2

s + 3 s + 2 s + 44 s + 7

ans =

b

ans =

T(s)

4 3 2

(- 7 s - 408 s + 1708 s + 14582 s +

5 4 3 2

27665) / (s - 4 s - 32 s + 148 s -

1153 s - 4480)

>>

**16.**

The equivalent cascade transfer function is as shown below.







Writing the state and output equations:

1 = x2

2 = x3

3 = - x1**-** x2- x3+ (t)

y = (t) = + x = x1+x2

In vector-matrix form,



**17.**

Since Tm = Jeq + Deqm, and Tm = Kt ia,

Jeq + Deqm = Kt ia (1)

Or,

= - m + ia

But, m = L.

Substituting in (1) and simplifying yields the first state equation,

= - L + ia

The second state equation is:

= L

Since

ea = Raia+La +Kbm = Raia+La +KbL,

the third state equation is found by solving for . Hence,

= - L - ia+ ea

Thus the state variables are: x1 = L, x2 = L , and x3 = ia.

Finally, the output is y = m = L .

In vector-matrix form,



where,



**18.**

Controller:

The transfer function can be represented as a block diagram as follows:



Writing the differential equation for the first box,



and solving for ,



From the second box,



Wheels:

The transfer function can be represented as a block diagram as follows:



Writing the differential equation for the block of the form,



and solving for  ,



The output equation is,

yw = xw

Vehicle:

The transfer function can be represented as a block diagram as follows:



Writing the differential equation for the block,



and solving for ,



The output equation is

yv = xv

**19.**

Adding displacements to the figure,



Writing the differential equations for noncontact,



Define the state variables as,



Writing the state equations, using the differential equations and the definition of the state variables, we get,



Assuming the output to be *xs*, the output equation is,



In vector-matrix form,



Writing the differential equations for contact,



Defining the state variables,



Using the differential equations and the definitions of the state variables, we write the state equations.



Differentiating the third differential equation and solving for d2*z*/d*t*2 we obtain,



But, from the fourth differential equation,



Substituting this expression back intoalong with the other definitions and then simplifying yields,



Continuing,



Assuming the output is *xs*,



Hence, the solution in vector-matrix form is



**20.**

**a.**

>> A=[-0.038 0.896 0 0.0015; 0.0017 -0.092 0 -0.0056; 1 0 0 -3.086; 0 1 0 0]

A =

-0.0380 0.8960 0 0.0015

0.0017 -0.0920 0 -0.0056

1.0000 0 0 -3.0860

0 1.0000 0 0

>> B = [-0.0075 -0.023; 0.0017 -0.0022; 0 0; 0 0]

B =

-0.0075 -0.0230

0.0017 -0.0022

0 0

0 0

>> C = [0 0 1 0; 0 0 0 1]

C =

0 0 1 0

0 0 0 1

>> [num,den] = ss2tf(A,B,C,zeros(2),1)

num =

0 0.0000 -0.0075 -0.0044 -0.0002

0 0 0.0017 0.0001 0

den =

1.0000 0.1300 0.0076 0.0002 0

>> [num,den] = ss2tf(A,B,C,zeros(2),2)

num =

0 -0.0000 -0.0230 0.0027 0.0002

0 -0.0000 -0.0022 -0.0001 0

den =

1.0000 0.1300 0.0076 0.0002 0

**b.**

From the MATLAB results

****

****

****

****

**21.**

**a.**



**b.**





**c.**





**22.**

**a.**

**b.**





**23.**

The armature loop equation is

The motor load equation is

Substituting the defined state variables one gets

Or

And

Or

Putting these equations in matrix form:

**24.**

**Let** . Cross multiplying the transfer function,



Taking the inverse Laplace transform,



Defining the state variables,



Writing the state equations,



In vector-matrix form,



**25.**

**a.**  The output equation is



**b.** Substituting the given values into the **A** matrix and **B** and **C** vectors yields,

; 

; D = 0

Using the following MATLAB M-file:

A=[0 1 0 0;29.8615 0 0 0;0 0 0 1;-0.9401 0 0 0];

B=[0;-1.1574;0; 0.4167];

C=[0.36 0 1 0];

D=0;

[numg, deng]=ss2tf(A, B, C, D, 1);

display G(s)=Y(s)/U(s);

G=tf(numg, deng)

The computer output (in the “Command” window) is:

G(s)=Y(s)/U(s)

G =

3.6e-05 s^2 - 2.378e-15 s - 11.36

---------------------------------

s^4 - 29.86 s^2

Continuous-time transfer function.

Note: As will be noted in chapter 6, a system with missing powers of ‘s’ and alternating signs of its denominator coefficients is unstable and needs to be stabilized.

**26.**

**a.**

****

****

****

****

****

****

****

****

****

****

****

****

****

****

****

****

****

****

Then just by direct substitution.

**b.**

Substituting values one gets:





**27.**

1. The following basic equations characterize the relationships between the state, input, and output variables for the HEV common forward path of the figure:



 (1)

, where ,

, 

1. Given that the state variables are the motor armature current, *Ia(t)*, and angular speed, *ω (t)*, we re-write the above equations as:

 (2)

 (3)

In matrix form, the resulting state-space equations are:

 (4)

 (5)

**28.**

The corresponding differential equation is:

We select the state variables as

So we can write

In matrix form:

Alternatively this can also be written as: