Analysis of State Space Systems

Akshya Swain

Department of Electrical,Computer & Software Engineering,
The University of Auckland.

Learning Outcomes

After completion of this module, the students should have learned the following:

- How to compute transfer function from state variable model
- How to Compute the Solution of State Equations

Computation of Transfer Function from State Variable Model

- Consider a system described by

$$\dot{x} = Ax + Bu \quad \& \quad y = Cx + Du \tag{1}$$

- Taking the Laplace transform of (1) with zero initial conditions give

$$sX(s) = AX(s) + BU(s) \;\&\; Y(s) = CX(s) + DU(s) \tag{2}$$

- Solving for $X(s)$ from the first equation of (2) gives

$$sX(s) - AX(s) = BU(s) \implies (sI - A)\, X(s) = BU(s)$$

$$\implies X(s) = (sI - A)^{-1} BU(s) \tag{3}$$

- Substituting $X(s)$ from (3) into $Y(s)$ in (2) gives

$$Y(s) = C\,(sI - A)^{-1} BU(s) + DU(s) = \left[C\,(sI - A)^{-1} B + D\right] U(s)$$

$$Hence \quad G(s) = \frac{Y(s)}{U(s)} = C\,(sI - A)^{-1} B + D \tag{4}$$

- The matrix $C\,(sI - A)^{-1} B + D$ is the transfer function matrix of the system.

The Characteristic Equation of System from State Space Model

- If y and u are scalars then, the transfer function of the system is given by

$$T(s) = \frac{Y(s)}{U(s)} = C\,(sI - A)^{-1}\,B + D$$

But

$$(sI - A)^{-1} = \frac{adj(sI - A)}{det(sI - A)} = \frac{adj(sI - A)}{|sI - A|}$$

Hence

$$\boxed{T(s) = C\frac{adj(sI - A)}{det(sI - A)}B + D = \frac{Cadj(sI - A)B + D\,|sI - A|}{|sI - A|}}$$

Thus $|sI - A| = 0$ is the characteristic equation of the system and its roots are equal to eigenvalues of the system matrix.

Example: Computation of Transfer Function from State Model

- Consider the system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u = \mathrm{A}\mathrm{x} + \mathrm{B}u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mathrm{C}\mathrm{x}$$

$$Thus \quad A = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \; C = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$Now \quad sI - A = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} = \begin{bmatrix} s & -1 \\ 1 & s+1 \end{bmatrix}$$

- Let us compute the inverse of a 2×2 matrix ; as we will frequently use it in many of the computations.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

- Swap the diagonal elements and change the sign of off diagonal elements divided by the determinant.

- Therefore

$$\begin{bmatrix} s & -1 \\ 1 & s+1 \end{bmatrix}^{-1} = \frac{1}{s(s+1)+1} \begin{bmatrix} s+1 & 1 \\ -1 & s \end{bmatrix}$$

$$= \frac{1}{s^2+s+1} \begin{bmatrix} s+1 & 1 \\ -1 & s \end{bmatrix}$$

Example: Computation of Transfer Function from State Model(contd)

- Let us compute $C(sI - A)^{-1}B$ as $D = 0$:

$$C(sI - A)^{-1}B = \begin{bmatrix} 1 & 0 \end{bmatrix} \frac{1}{s^2 + s + 1} \begin{bmatrix} s+1 & 1 \\ -1 & s \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{s^2+s+1} \\ \frac{s}{s^2+s+1} \end{bmatrix} = \frac{1}{s^2 + s + 1}$$

- The poles of the system are located at $-0.5 \pm j0.866$.
- The eigenvalues of the system matrix is computed as:

$$\left| \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \right| - \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} = 0 \implies \left| \begin{bmatrix} \lambda & -1 \\ 1 & \lambda + 1 \end{bmatrix} \right| = 0$$
$$\implies \lambda(\lambda + 1) + 1 = 0 \implies \lambda^2 + \lambda + 1 = 0$$

- This gives $\lambda_1, \lambda_2 = -0.5 \pm j0.866$

Some Basic MATLAB Commands to Manipulate State Space Based Systems

1. Given the system, input, output and transmission matrices, how to create a state space system

   MATLAB Code

```
sys=ss(A,B,C,D);
```

2. Given that we have defined our state space system, how can we extract the matrices A,B,C and D

   MATLAB Code

```
[A,B,C,D]=ssdata(sys);    % Returns matrices A,B,C and D.
```


Some Basic MATLAB Commands to Manipulate State Space Based Systems

3. Converting transfer function model to state space model :

   Given the system transfer function in terms of numerator and denominator polynomial vectors, how to create a state space system

   MATLAB Code

```
[A,B,C,D]=tf2ss(num,den);
sys=ss(A,B,C,D);
```

4. Converting state space model to transfer function model : Given that we have the matrices A,B,C and D, how to create transfer function model ?

   MATLAB Code

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

Scalar Case:

- Consider a system described by scalar differential equation

$$\dot{x} = ax \tag{5}$$

- Taking the Laplace transform of (5) gives

$$sX(s) - x(0) = aX(s) \tag{6}$$

where $X(s)$ is the Laplace transform of $x(t)$

- Solving (6) for $X(s)$ gives

$$X(s) = \frac{x(0)}{s-a} = (s-a)^{-1}x(0)$$
$$\Longrightarrow x(t) = e^{at}x(0)$$

$$\boxed{\dot{x} = ax, \qquad x(t) = e^{at}x(0)}$$

Solution of Homogeneous State Equations: Laplace Tranaform Approach

Vector Case:

- Consider the vector differential equation

$$\dot{\mathrm{x}} = \mathrm{A}\mathrm{x} \tag{7}$$

- Taking the Laplace transform of (7) gives

$$s\mathrm{X}(s) - x(0) = \mathrm{A}\mathrm{X}(s) \tag{8}$$

where $\mathrm{X}(s)$ is the Laplace transform of x(t)

- Solving (8) for X(s) gives

$$\mathrm{X(s)} = \frac{x(0)}{s\mathrm{I} - \mathrm{A}} = (s\mathrm{I} - \mathrm{A})^{-1}x(0)$$

$$\Longrightarrow \mathrm{x}(t) = \mathcal{L}^{-1}\left[(s\mathrm{I} - A)^{-1}\right]\mathrm{x}(0) = \Phi(t)\mathrm{x}(0) = e^{\mathrm{A}t}\mathrm{x}(0)$$

Solution of Homogeneous State Equations: Laplace Transform Approach

- Note that

$$(s\mathrm{I} - A)^{-1} = \frac{\mathrm{I}}{s} + \frac{\mathrm{A}}{s^2} + \frac{\mathrm{A}^2}{s^3} + \ldots$$

- Hence the inverse Laplace transform of $(s\mathrm{I} - \mathrm{A})^{-1}$ gives

$$\mathcal{L}^{-1}\left[(s\mathrm{I} - A)^{-1}\right] = \mathrm{I} + \mathrm{A}t + \frac{\mathrm{A}^2 t^2}{2!} + \frac{\mathrm{A}^3 t^3}{3!} + \ldots = e^{\mathrm{A}t}$$

- Thus the solution of state equation can alternately be written as

$$\boxed{\mathrm{x}(t) =) e^{\mathrm{A}t} x(0) = \Phi(t) x(0)}$$

where $\Phi(t)$ is called the State Transition Matrix

1. $\Phi(0) = \mathrm{I}$
2. $\Phi^{-1}(t) = \Phi(-t)$
3. $\Phi(t_1)\Phi(t_2) = \Phi(t_2)\Phi(t_1)$
4. $[\Phi(t)]^n = \Phi(nt)$
5. $\Phi(t_2 - t_1)\Phi(t_1 - t_0) = \Phi(t_2 - t_0)$

Proof of Properties of State Transition Matrix

1. $\Phi(0) = \mathrm{I}$

   Proof:

$$\Phi(t) = e^{\mathrm{A}t} \implies \Phi(0) = e^{\mathrm{A}0} = \mathrm{I}$$

2. $\Phi^{-1}(t) = \Phi(-t)$

   Proof:

$$\Phi(t) = e^{\mathrm{A}t} = \left(e^{-\mathrm{A}t}\right)^{-1} = [\Phi(-t)]^{-1} \implies \Phi^{-1}(t) = \Phi(-t)$$

3. $\Phi(t_1)\Phi(t_2) = \Phi(t_2)\Phi(t_1)$

   Proof:

$$\Phi(t_1 + t_2) = e^{\mathrm{A}(t_1+t_2)} = e^{\mathrm{A}t_1} e^{\mathrm{A}t_2} = e^{\mathrm{A}t_2} e^{\mathrm{A}t_1} = \Phi(t_2)\Phi(t_1)$$

Proof of Properties of State Transition Matrix

4. $[\Phi(t)]^n = \Phi(nt)$

   Proof:

$$[\Phi(t)]^n = \left[e^{\mathrm{A}t}\right]^n = e^{\mathrm{A}nt} = \Phi(nt)$$

5. $\Phi(t_2 - t_1)\Phi(t_1 - t_0) = \Phi(t_2 - t_0)$

   Proof:

$$\begin{aligned}\Phi(t_2 - t_1)\Phi(t_1 - t_0) &= e^{\mathrm{A}(t_2 - t_1)} e^{\mathrm{A}(t_1 - t_0)} \\ &= e^{\mathrm{A}t_2} e^{-\mathrm{A}t_1} e^{\mathrm{A}t_1} e^{-\mathrm{A}t_0} = \Phi(t_2 - t_0)\end{aligned}$$

Example: Computation of State Transition Matrix

- Compute the state transition matrix of the system

$$\dot{\mathrm{x}} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \mathrm{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \tag{9}$$

  Solution:

  For this system

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \tag{10}$$

- The state transition matrix $\Phi(t)$ is given by

$$\Phi(t) = \mathcal{L}^{-1}\left[(s\mathrm{I} - A)^{-1}\right] \tag{11}$$

Solution :Example: Computation of State Transition Matrix

- Now

$$(sI - A) = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} = \begin{bmatrix} s & -1 \\ 2 & s+3 \end{bmatrix} \tag{12}$$

- The inverse of $(sI - A)$ is given by

$$\begin{aligned}(sI - A)^{-1} &= \begin{bmatrix} s & -1 \\ 2 & s+3 \end{bmatrix}^{-1} = \frac{1}{s(s+3)+2} \begin{bmatrix} s+3 & 1 \\ -2 & s \end{bmatrix} \\ &= \begin{bmatrix} \frac{s+3}{(s+1)(s+2)} & \frac{1}{(s+1)(s+2)} \\ \frac{-2}{(s+1)(s+2)} & \frac{s}{(s+1)(s+2)} \end{bmatrix}\end{aligned} \tag{13}$$

Solution :Example: Computation of State Transition Matrix

- Taking Laplace inverse of $(sI - A)$ gives the state transition matrix as follows:

$$(sI - A)^{-1} = \begin{bmatrix} \frac{s+3}{(s+1)(s+2)} & \frac{1}{(s+1)(s+2)} \\ \frac{-2}{(s+1)(s+2)} & \frac{s}{(s+1)(s+2)} \end{bmatrix} = \begin{bmatrix} \frac{s+3}{(s+1)(s+2)} & \frac{1}{(s+1)(s+2)} \\ \frac{-2}{(s+1)(s+2)} & \frac{s}{(s+1)(s+2)} \end{bmatrix} = \begin{bmatrix} \frac{2}{s+1} - \frac{1}{s+2} & \frac{1}{s+1} - \\ \frac{22}{s+1} + \frac{2}{s+2} & \frac{-1}{s+1} + \end{bmatrix}$$

- Now

$$\Phi(t) = \mathcal{L}^{-1}\left[(sI - A)^{-1}\right] = \mathcal{L}^{-1}\begin{bmatrix} \frac{2}{s+1} - \frac{1}{s+2} & \frac{1}{s+1} - \frac{1}{s+2} \\ \frac{22}{s+1} + \frac{2}{s+2} & \frac{-1}{s+1} + \frac{2}{s+2} \end{bmatrix}$$

$$= \begin{bmatrix} 2e^{-t} - e^{-2t} & e^{-t} - e^{-2t} \\ -2e^{-t} + 2e^{-2t} & -e^{-t} + 2e^{-2t} \end{bmatrix} \tag{14}$$

Solution to Non-homogeneous State Equations : Vector case

- Consider the system described by the state equation

$$\dot{\mathrm{x}} = \mathrm{Ax} + \mathrm{B}u$$

and the output equation

$$\mathrm{x} = \mathrm{Cx} + \mathrm{D}u$$

- Taking the Laplace transform of both sides of state equation yields

$$\begin{aligned} s\mathrm{X}(s) - x(0) &= \mathrm{AX}(s) + \mathrm{BU}(s) \\ \implies (s\mathrm{I} - \mathrm{A})\,\mathrm{X}(s) &= x(0) + \mathrm{BU}(s) \end{aligned}$$

where Iis an $(n \times n)$ identity matrix where n is the order of the system.

Solution to Non-homogeneous State Equations : Vector case

- By premultiplying $(s\mathrm{I} - \mathrm{A})^{-1}$, gives the solution as:

$$\mathrm{X}(s) = (s\mathrm{I} - \mathrm{A})^{-1}\, x(0) + (s\mathrm{I} - \mathrm{A})^{-1}\, \mathrm{BU}(s) \tag{15}$$

- Taking inverse Laplace transform of the state equation gives

$$\begin{aligned} x(t) &= e^{At}x(0) + \int_0^t e^{A(t-\tau)}\mathrm{B}u(\tau)d\tau \\ &= \Phi(t)x(0) + \int_0^t \Phi(t-\tau)\mathrm{B}u(\tau)d\tau \end{aligned} \tag{16}$$

Procedure to Compute State and Output Response of Non-homogeneous State Equations

- Step-1: Compute the Laplace Transform of $x(t)$ from

$$\begin{aligned} \mathrm{X}(s) &= (s\mathrm{I} - \mathrm{A})^{-1}\, x(0) + (s\mathrm{I} - \mathrm{A})^{-1}\, \mathrm{BU}(s) \\ &= (s\mathrm{I} - \mathrm{A})^{-1} \left[x(0) + \mathrm{BU}(s) \right] \end{aligned} \tag{17}$$

- Step-2: Compute the transform of output $y(t)$ from

$$\mathrm{Y}(s) = \mathrm{CX}(s) + \mathrm{DU}(s)$$

- Step-3: Compute the inverse Laplace transform of $\mathrm{X}(s)$ and $\mathrm{Y}(s)$ to get $x(t)$ and $y(t)$

Example: Computation of State and Output Response from Non-homogeneous State Equations

Consider the system

$$\dot{\mathrm{x}} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \mathrm{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, y = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathrm{x}, \mathrm{x}(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{18}$$

Obtain both the state and output response for a unity step input.

Solution: From the system equation we have

$$\mathrm{A} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}, \mathrm{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Solution: Example (contd)

- Let us first compute $(s\mathrm{I} - \mathrm{A})^{-1}$

$$(sI - A)^{-1} = \begin{bmatrix} s & -1 \\ 2 & s+3 \end{bmatrix}^{-1} = \frac{1}{s(s+3)+2} \begin{bmatrix} s+3 & 1 \\ -2 & s \end{bmatrix}$$

$$= \begin{bmatrix} \frac{s+3}{(s+1)(s+2)} & \frac{1}{(s+1)(s+2)} \\ \frac{-2}{(s+1)(s+2)} & \frac{s}{(s+1)(s+2)} \end{bmatrix} = \begin{bmatrix} \mathrm{s}_{11} & \mathrm{s}_{12} \\ \mathrm{s}_{21} & \mathrm{s}_{22} \end{bmatrix} \tag{19}$$

- Next compute $x(0) + \mathrm{B}\,U(s)$ which for the given data becomes

$$x(0) + \mathrm{B}\mathrm{U}(s) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \frac{1}{s} = \begin{bmatrix} 0 \\ \frac{1}{s} \end{bmatrix} \tag{20}$$

Solution : Example (contd)

- Compute $X(s) = (sI - A)^{-1}[x(0) + BU(s)]$. This gives

$$X(s) = (sI - A)^{-1}[x(0) + BU(s)] = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \begin{bmatrix} 0 \\ \frac{1}{s} \end{bmatrix} = \begin{bmatrix} \frac{s_{12}}{s} \\ \frac{s_{22}}{s} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{s(s+1)(s+2)} \\ \frac{1}{(s+1)(s+2)} \end{bmatrix} = \begin{bmatrix} \frac{0.5}{s} - \frac{1}{s+1} + \frac{0.5}{s+2} \\ \frac{1}{(s+1)} - \frac{1}{(s+2)} \end{bmatrix} \tag{21}$$

Hence

$$x(t) = \begin{bmatrix} 0.5 - e^{-t} + 0.5e^{-2t} \\ e^{-t} - e^{-2t} \end{bmatrix}$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} x(t) = 0.5 - e^{-t} + 0.5e^{-2t}$$

Simulating Differential Equation Model of a System using ode45-1

- The simulations are often carried out by using 4th and 5th order Rungga-Kutta method of numerical integration using ode45 function in MATLAB.
- Note that ode45 will essentially integrate a first order ordinary differential equation (either linear or nonlinear). The procedure for getting the response are summarised below.

Step-1 :Represent the n-th order differential equation model of the system by n number of first order differential equations by selecting suitable state variables.
Step-2 : Create a function file (a m-file) using either Matlab's editor or any text editor, e.g. "notepad" to represent the system dynamics.
Step-3 : Create the main file of simulation specifying the initial conditions, time step and final time of simulation.

- There are two options:

Option-1: Define the input and other parameters (if any) inside the function.
Option-2: Pass the input and other parameters (if any) from outside i.e. (the main file) (preferred option for control).

Example-1: Simulating a Mass-Spring-Damper System-1

Consider a mass-spring-damper system which is modelled by the differential equation

$$m\ddot{x}(t) + f\dot{x}(t) + kx(t) = u(t)$$

where m,f and k denote the mass, damping constant and spring constant of the system respectively and x is the displacement and u is the force.

- The objective is to find $x(t)$,$\dot{x}(t)$ for a step input $u(t)$ considering $m = 1$,$f = 1$ and $k = 1$. Assume initial conditions $x(0) = 2$ and $\dot{x}(0) = 0$ .Compute various responses such as displacement, acceleration) of

Solution

Step-1 :Represent the system dynamics by 2 number of first order differential equations.Let us select $x_1 = x$ and $x_2 = \dot{x}$. This gives

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{k}{m}x_1 - \frac{f}{m}x_2 + \frac{1}{m}u\end{aligned}$$

Example-1: Simulating a Mass-Spring-Damper System-2

Step-2 : Lets us create a function file (a m-file) using Matlab's editor.

Option-1: The input and other parameters are specified inside the function.

Let us call this function mass-spring1.m

MATLAB Code: mass-spring1.m

```
function dx=mass-spring1(t,x);
m=1;f=1;k=1;u=1;
x1dot=x(2);
x2dot=(-k/m)*x(1)-(f/m)*x(2)+(1/m)*u;
dx=[x1dot;x2dot];
```


Example-1: Simulating a Mass-Spring-Damper System-3

Step-3 : Lets us create the main file (a m-file) and call the function file test-mass-spring1.m.

Save this file as test-mass-spring1.m.

MATLAB Code: test-mass-spring1.m

```
x0=[0.0 0.0];     % Initial Conditions;a row vector
y(1,1)=x0(1);
x1(1)=x0(1,1);    % Save the state variables
x2(1)=x0(1,2);
timestep=0.01;    % Integration step.
t0=0.0; tfinal=10.0;
[t,x]=ode45(@mass-spring1,[t0 tfinal],x0);
```

Example-1: Simulating a Mass-Spring-Damper System by Passing Parameters from the Main File

Step-1 : Lets us create a function file (a m-file) and call the function file mass-spring2.m

For this example, let us pass the input as well; as the three parameters m,f and k.

MATLAB Code: mass-spring2.m

```
function dx=mass-spring2(t,x,u,m,f,k);
x1dot=x(2);
x2dot=(-k/m)*x(1)-(f/m)*x(2)+(1/m)*u;
dx=[x1dot;x2dot];
```


Example-1: Simulating a Mass-Spring-Damper System by Passing Parameters from the Main File

Step-2 : Lets us create the mail file (a m-file) using Matlab's editor.
Let us call this function test-mass-spring2.m
MATLAB Code: test-mass-spring2.m

```
m=1.0;k=25.0;wn=sqrt(k/m);zeta=0.6;f=2*zeta*wn;
x0=[0.0 0.0];    % Initial Conditions y(1,1)=x0(1);x1(1)=x0(1,1);x2(1)=x0(1,2);
timestep=0.01;   % Integration step.
t0=0.0;tfinal=0.0;iterations=1000;
u(1)=1.0;t(1)=t0;
for i=2:iterations
uc=u(i-1); t0=tfinal;tfinal=tfinal+timestep; t(i)=tfinal;
[dum,allx]=ode45(@mass_spring1,[t0 tfinal],x0,[],u,m,f,k);
xx=flipud(allx); [m1 n1]=size(xx); x0=xx(1,1:n1);
x1(i)=xx(1,1);x2(i)=xx(1,2);y(i,1)=x1(i);
u(i,1)=1.0;
end;
plot(t(1:length(x1)),y)
grid xlabel('Time (s)') ylabel('Output y ')
```