MATLAB Experiment III – State Space

3.1 Introduction

MATLAB can also be used to transform the system model from transfer function to state space, and vice versa.

3.2 Transfer Function to State Space

Given a transfer function of the form:

$$H(s) = \frac{B(s)}{A(s)} = \frac{\text{num polynomial in } s}{\text{den polynomial in } s} = \frac{\text{num}}{\text{den}}.$$

MATALB can be used to obtain a state-space representation of the transfer function with the following command

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

It is important to note that the state space representation is not unique, i.e. there are many state-space representations for the same system. The MATLAB command gives just one possible state-space equation.

Example. Consider the transfer function

$$H(s) = \frac{B(s)}{A(s)} = \frac{s}{(s+10)(s^2+4s+16)}.$$

A possible state-space representation for this system is

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -160 & -56 & -14 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ -14 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + [0]u.$$

For the example transfer function, MATLAB produces the following state-space representation:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \end{bmatrix} = \begin{bmatrix} -14 & -56 & -160 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u$$

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

with the following programme:

```
>> num = [0  0  1  0];
>> den = [1  14  56  160];
>> [A, B, C, D] = tf2ss(num, den)
A =
   -14   -56  -160
```

```
     1     0     0
     0     1     0
B =
     1
     0
     0
C =
     0     1     0
D =
     0
```

3.3 State Space to Transfer Function

To obtain the transfer function from state-space equations, use the following MATLAB code

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

where `num_in` must be specified for systems with more than one input. For example, if there is only one input, then `num_in = 1`, if there are two inputs then `num_in = 2`, etc. For systems with only one input, then the following code is allowed

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

Example. Given the state-space equations:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -5 & -25 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 25 \\ -120 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix},$$

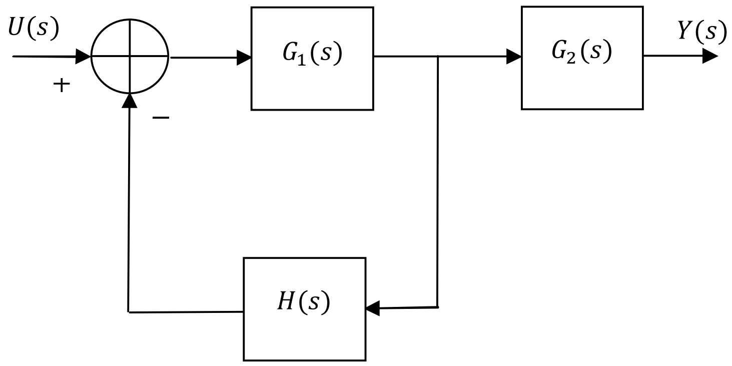
MATLAB produces the following system transfer function:

$$H(s) = \frac{B(s)}{A(s)} = \frac{25s + 5}{s^3 + 5s^2 + 25s + 5}.$$

The MATLAB implementation of this is given by the following code:

```
>> A = [0  1  0;  0  0  1; -5  -25  -5];
>> B = [0;  25; -120];
>> C = [1  0  0];
>> D = [0];
>> [num, den] = ss2tf(A, B, C, D)
num =
        0    0.0000   25.0000    5.0000
den =
    1.0000    5.0000   25.0000    5.0000
```

Example. Using MATLAB, obtain the transfer function for the control system in the figure below.



where

$$\mathbf{G}_1 = \begin{pmatrix} 1 & 3 \\ 2 & 5 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u; \qquad \mathbf{C}_1 = (1 \quad 2)$$

$$\mathbf{G}_2 = \begin{pmatrix} -4 & 0 \\ 0 & 3 \end{pmatrix} x + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u; \qquad \mathbf{C}_2 = (3 \quad 0)$$

$$\mathbf{H} = \begin{pmatrix} 3 & 2 \\ 1 & 0 \end{pmatrix} x + \begin{pmatrix} 1 \\ 1 \end{pmatrix} u; \qquad \mathbf{C}_3 = (1 \quad 1)$$

Solution: To define the system computer first we have to define the matrices as:

```
>> a1 = [1  3; 2  5];
>> b1 = [0;  1];
>> c1 = [1  2];
>> d1 = 0;
>> a2 = [-4  0; 0  3];
>> b2 = [1;  0];
>> c2= [3  0];
>> d2 = 0;
>> a3 = [3  2; 1  0];
>> b3 = [1;  1];
>> c3= [1  1];
>> d3 = 0;
```

The following MATLAB code will convert the system from state-space to transfer function:

```
>> [numg1, deng1] = ss2tf(a1,b1,c1,d1);
>> [numg2, deng2] = ss2tf(a2,b2,c2,d2);
>> [numh, denh] = ss2tf(a3,b3,c3,d3);
>> [numf, denf] = feedback(numg1,deng1,numh,denh);
>> [nums, dens] = series(numf,denf,numh,denh);
>> printsys(nums,dens)

num/den =

              4 s^4 - 10 s^3 - 14 s^2 - 4 s
   -----------------------------------------------
   s^6 - 12 s^5 + 44 s^4 - 22 s^3 - 87 s^2 - 40 s - 4
```

MATLAB can also be used to convert state-space representations to parallel form or controller canonical form using the command

```
cannon(A,B,C,D, 'type')
```

where
`type = modal`, yields parallel form
`type = companion`, yields controller canonical form.