

State-Space Representation of System Dynamics

State-Space is a matrix-based formulation for a system's dynamics. The standard form for the dynamics of a linear system are

$$sX = AX + BU$$

$$Y = CX + DU$$

where Y is the system's output, U is the system's input, and X are 'dummy' states (termed internal states.) With this formulation, the transfer function will be

$$X = (sI - A)^{-1} BU$$

$$Y = \left(C(sI - A)^{-1} B + D \right) U$$

MATLAB is a matrix language which has functions that let you find the transfer function of a system in state-space form. It's much easier (and less prone to errors) if you express many systems in state-space form and let MATLAB determine the transfer function. It's also an easier way to implement a system on a microcomputer.

MATLAB Commands

- | | |
|-------------------------|--|
| • $G = ss(A, B, C, D);$ | input a system in state-space form |
| • $G = tf(num, den)$ | input a system in transfer function form |
| • $G = zp(z, p, k)$ | input a system in zeros, poles, gain form |
| • $ss(G)$ | determine A, B, C, D for system G (answer is not unique) |
| • $tf(G)$ | determine the transfer function of system G |
| • $zp(G)$ | determine the zeros, poles, and gain of system G |

Matrix Algebra

An $n \times m$ matrix has n rows and m columns. For example, a 2×3 matrix has 6 elements:

$$A_{2 \times 3} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$

Scalar Multiplication:

$$bA = \begin{bmatrix} ba_{11} & ba_{12} & ba_{13} \\ ba_{21} & ba_{22} & ba_{23} \end{bmatrix}$$

Matrix Addition:

Matrices with the same dimensions can be added:

$$A + B = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix} = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} & a_{13} + b_{13} \\ a_{21} + b_{21} & a_{22} + b_{22} & a_{23} + b_{23} \end{bmatrix}$$

Matrix Multiplication:

To multiply two matrices, the inner dimensions must match

$$A_{xy} \cdot B_{yz} = C_{xz}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} \\ a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} \end{bmatrix}$$

$$= \begin{bmatrix} \sum a_{1x}b_{x1} & \sum a_{1x}b_{x2} \\ \sum a_{2x}b_{x1} & \sum a_{2x}b_{x2} \end{bmatrix}$$

Matrix Inversion

$$A \cdot A^{-1} = I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad I = \text{the identity matrix} = \text{the matrix version of '1'}$$

For a 2x2 matrix,

$$A^{-1} = \begin{bmatrix} \frac{a_{22}}{\Delta} & \frac{-a_{12}}{\Delta} \\ \frac{-a_{21}}{\Delta} & \frac{a_{11}}{\Delta} \end{bmatrix} \quad \Delta = a_{11}a_{22} - a_{12}a_{21}$$

Placing a System in State-Space Form:

- i) Write N equations for the N voltage nodes
- ii) Solve for the highest derivative for each equation
- iii) Rewrite in matrix form.

Example. The following differential equation describe the water level in a two-tank system. Write this in state-space form and find the transfer function from U to Y.

$$\frac{dx_1}{dt} = x_2 - x_1 + 0.3u$$

$$\frac{dx_2}{dt} = x_1 - 1.3x_2$$

Write this as

$$sX = AX + BU$$

$$s \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1 & -1.3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.3 \\ 0 \end{bmatrix} U$$

If x2 is the output,

$$Y = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]U$$

To find the transfer function in MATLAB,

```
A = [-1, 1; 1, -1.3]
B = [0.3; 0]
C = [0, 1]
D = 0;
```

```
G = ss(A, B, C, D)
tf(G)
```

```
Transfer function:
      0.3
-----
s^2 + 2.3 s + 0.3
```

Example: The temperature along the length of a metal bar are described by

$$sx_1 = u - 2x_1 + x_2$$

$$sx_2 = x_1 - 2x_2 + x_3$$

$$sx_3 = x_2 - 2x_3 + x_4$$

$$sx_4 = x_3 - x_4$$

Find the transfer function from U to X4.

Solution: Express this in state-space form

$$s \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} U$$

$$y = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + [0]U$$

Input this into MATLAB

```
A = [-2,1,0,0;1,-2,1,0;0,1,-2,1;0,0,1,-1];
B = [1;0;0;0];
C = [0,0,0,1];
D = 0;
```

```
G = ss(A,B,C,D);
tf(G)
```

Transfer function:

$$\frac{1}{s^4 + 7s^3 + 15s^2 + 10s + 1}$$

or if you prefer factored form:

`zpk(G)`

Zero/pole/gain:

$$\frac{1}{(s+1)(s+2.347)(s+3.532)(s+0.1206)}$$

Suppose you wanted to measure the average temperature of the bar. The only change is what you measure:

$$y = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + [0]U$$

`C = [0.25,0.25,0.25,0.25];`
`D = 0;`

`G = ss(A,B,C,D);`
`tf(G)`

Transfer function:

$$\frac{0.25s^3 + 1.5s^2 + 2.5s + 1}{s^4 + 7s^3 + 15s^2 + 10s + 1}$$

or if you prefer factored form:

`» zpk(G)`

Zero/pole/gain:

$$\frac{0.25(s+3.414)(s+2)(s+0.5858)}{(s+1)(s+2.347)(s+3.532)(s+0.1206)}$$

note: You don't have to use MATLAB. You can also write this using Laplace notation as four equations for four unknowns. Using algebra (and about three hours), you can simplify and get the same answer.

Canonical Forms

Objective:

- Give a block-diagram representation for a transfer function in various canonical forms
- Give a state-space representation for a transfer function in various canonical forms

State-space is the way MATLAB and other programs represent transfer functions. One feature of state-space is there are an infinite number of ways to represent the same transfer function. Some forms have standard names - these are termed 'canonical forms.'

Problem: Place the following system in state-space form

$$Y = \left(\frac{a_3 s^3 + a_2 s^2 + a_1 s + a_0}{s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0} \right) U$$

Controller Canonical Form:

Change the problem to

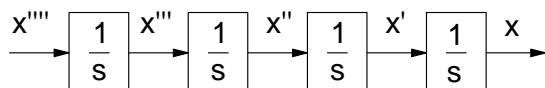
$$X = \left(\frac{1}{s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0} \right) U$$

$$Y = (a_3 s^3 + a_2 s^2 + a_1 s + a_0) X$$

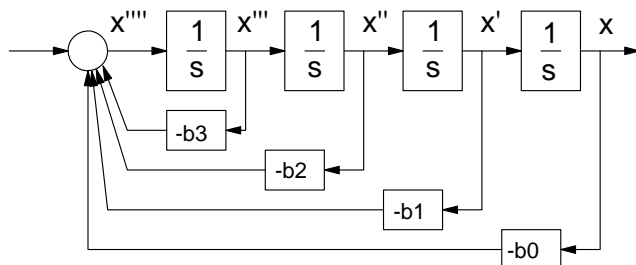
Solve for the highest derivative of X:

$$s^4 X = U - b_3 s^3 X + b_2 s^2 X + b_1 s X + b_0 X$$

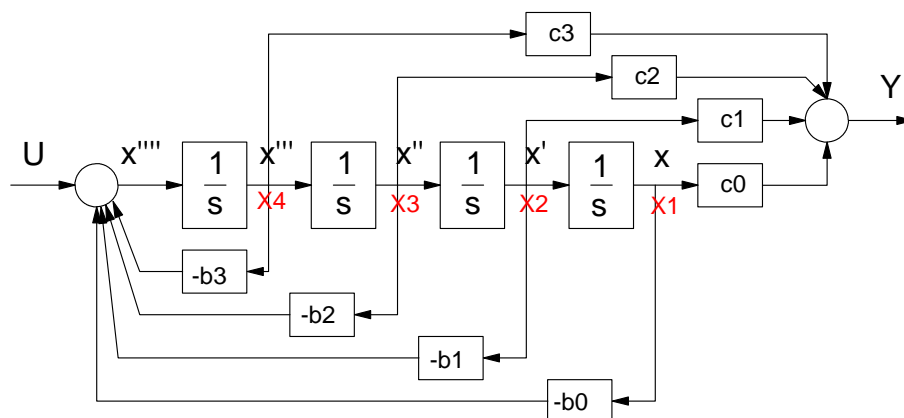
Integrate $s^4 X$ four times to get X: (note: x' means $\frac{dx}{dt}$)



Create $s^4 X$ from U and its derivatives



Create Y from the derivatives of X:



Controller Canonical Form

State Space Form: Define a state to be the output of each integrator (shown in red above). This gives

$$sX_1 = X_2$$

$$sX_2 = X_3$$

$$sX_3 = X_4$$

$$sX_4 = U - b_3X_4 + b_2X_3 + b_1X_2 + b_0X_1$$

or in matrix form:

$$s \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -b_0 & -b_1 & -b_2 & -b_3 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} U$$

$$Y = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + [0]U$$

Note that you can write this state-space model by inspection from the transfer function:

$$Y = \left(\frac{a_3s^3 + a_2s^2 + a_1s + a_0}{s^4 + b_3s^3 + b_2s^2 + b_1s + b_0} \right) U$$

This is what MATLAB uses to store transfer functions since it's so easy to determine once you know the transfer function. It's also easy to convert from controller canonical form to the transfer function.

This form is called 'controller form' since the input, U, can set the states at will. For example, if

$$u(t) = \delta(t)$$

the output of the first integrator jumps to 1 at $t=0+$. If

$$u(t) = \frac{d}{dt}(\delta(t))$$

(termed a doublet), the output of the second integrator jumps to 1 at $t=0+$.

This form also has some of the worst numerical properties. Nothing is free.

Observer Canonical Form:

If you transpose controller canonical form you get observer form:

$$A_o = (A_c)^T$$

$$B_o = (C_c)^T$$

$$C_o = (B_c)^T$$

or

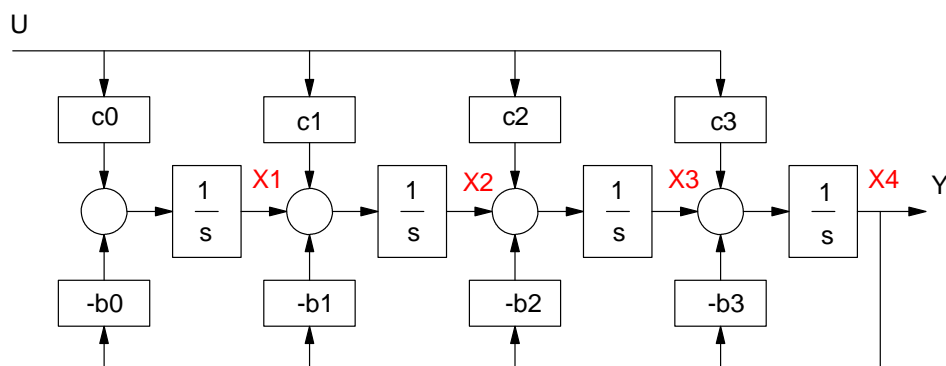
$$s \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -b_0 \\ 1 & 0 & 0 & -b_1 \\ 0 & 1 & 0 & -b_2 \\ 0 & 0 & 1 & -b_3 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} U$$

$$Y = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + [0]U$$

The block-diagram version looks like the following:

$$sX_1 = -b_0X_4 + c_0U$$

etc.



Observer Canonical Form

The nice thing about observer canonical form is you can determine all four states by measuring the output, Y, and its derivatives.

Cascade Form:

Assume the transfer function factors as

$$Y = \left(\frac{c_0}{(s+p_1)(s+p_2)(s+p_3)(s+p_4)} \right) U$$

Treat this as four systems cascaded:

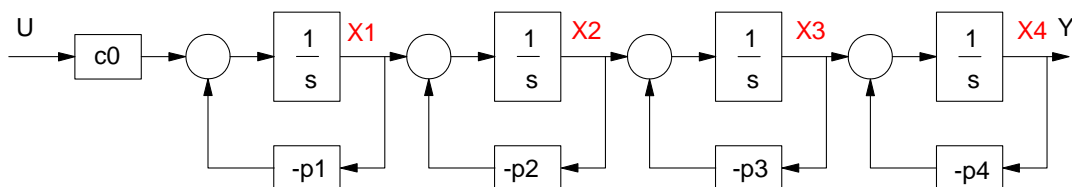
$$X_1 = \left(\frac{c_0}{s+p_1} \right) U$$

$$X_2 = \left(\frac{1}{s+p_2} \right) X_1$$

$$X_3 = \left(\frac{1}{s+p_3} \right) X_2$$

$$X_4 = \left(\frac{1}{s+p_4} \right) X_3$$

The block diagram model looks like the following:



The state-space model is:

$$s \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} -p_1 & 1 & 0 & 0 \\ 0 & -p_2 & 1 & 0 \\ 0 & 0 & -p_3 & 1 \\ 0 & 0 & 0 & -p_4 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + \begin{bmatrix} c_0 \\ 0 \\ 0 \\ 0 \end{bmatrix} U$$

$$Y = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + [0]U$$

The nice thing about cascade form is it has very good numerical properties. You can also determine the poles of the system by inspection: they're the values on the diagonal.

Jordan Form:

Assume the transfer function can be expressed using partial fractions as

$$Y = \left(\left(\frac{a_1}{s+p_1} \right) + \left(\frac{a_2}{s+p_2} \right) + \left(\frac{a_3}{s+p_3} \right) + \left(\frac{a_4}{s+p_4} \right) \right) U$$

Treat this as four separate systems:

$$X_1 = \left(\frac{c_1}{s+p_1} \right) U$$

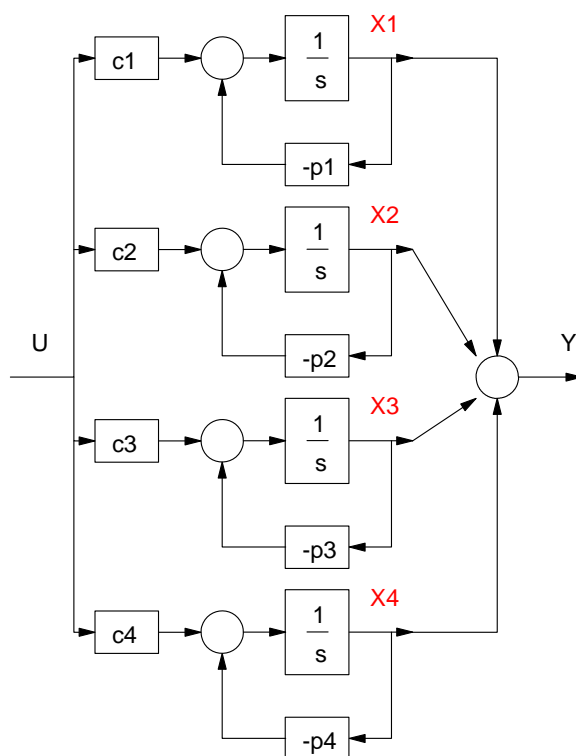
$$X_2 = \left(\frac{c_2}{s+p_2} \right) U$$

$$X_3 = \left(\frac{c_3}{s+p_3} \right) U$$

$$X_4 = \left(\frac{c_4}{s+p_4} \right) U$$

Y is then the sum of these four.

In block diagram form, Jordan form looks like the following:



In state-space, this looks like:

$$s \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} = \begin{bmatrix} -p_1 & 0 & 0 & 0 \\ 0 & -p_2 & 0 & 0 \\ 0 & 0 & -p_3 & 0 \\ 0 & 0 & 0 & -p_4 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} U$$

$$Y = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + [0]U$$