STATE-SPACE CONTROL: Conversion between state-space and transfer functions forms

Here we focus on hand calculations. Of course, Matlab makes it easier!

In the previous section we introduced state-space models

$$\dot{\mathbf{x}} = A\,\mathbf{x} + B\,\mathbf{u}$$

$$\mathbf{y} = C\,\mathbf{x} + D\,\mathbf{u}$$

Example – show this as a block diagram, with A, B, C, D as matrix multiplication blocks

We can determine the transfer function by taking Laplace transforms

$$s\boldsymbol{X}(s) = A\,\mathbf{X}(s) + B\,\boldsymbol{U}(s)$$

$$\mathbf{Y}(s) = C\,\boldsymbol{X}(s) + D\,\boldsymbol{U}(s)$$

Note – the Laplace transform is a linear mapping and works for vectors and matrices, the same way it works for scalar variables.

Hence

$$(sI - A)\boldsymbol{X}(s) = B\,\boldsymbol{U}(s)$$

$$\boldsymbol{X}(s) = (sI - A)^{-1}B\,\boldsymbol{U}(s)$$

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

So $\mathbf{Y}(s) = G(s)\,\boldsymbol{U}(s)$ with

$$G(s) = C\,(sI - A)^{-1}B\; + D$$

Problem – check the matrix dimensions of this equation for a system with 2 inputs, 2 outputs and 10 state variables. What is the matrix dimension of G?

- What does G represent? Draw a block diagram to show how the different elements of G connect inputs to outputs
- Draw another block diagram, for a SISO system, showing how the four matrices (A, B, C, D) connect the input to the output. What is the role of D?

The above equations show how to convert from state-space format to transfer function form [*ss2tf* in Matlab]. Mostly we restrict attention to SISO systems when using this function.

The equation

$$G(s) = C\,(sI - A)^{-1}B\ + D$$

allows us to connect *transfer function poles* to the properties of matrix A. Take the example

$$G(s) = \frac{s+3}{(s+1)(s+2)}$$

The poles are s = -1, -2 which are also the points where $G(s) \to \infty$.

In state-space form,

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

so, the poles are values of s with zero determinant: these are also the *eigenvalues* of A:

$$\det(sI - A) = 0$$

Example – for the following state-space system, find (i) the transfer function $G(s) = Y(s)/U(s)$, (ii) the s-plane poles and (iii) the eigenvalues of the A-matrix:

$$\begin{aligned}
&\dot{x}_1 = -x_1 + x_2 \\
&\dot{x}_2 = -2x_2 + 4x_3 \\
&\dot{x}_3 = -3x_3 + u \\
&y = x_1 + x_2
\end{aligned}$$

We need to <u>do this example by hand</u> to fully see how it works.

Matlab can do the same calculation very easily using the command *ss2tf*.

In the opposite direction, we can convert from a transfer function to state space, but there is a ‘catch’, as we will see.

Example – Find *the state space equivalent of the transfer function

$$G(s) = \frac{1}{s^3 + s^2 + 1}$$

Again, we should do the example by hand. To solve the problem, we need to define system states. If y is the output, we can define

$$x_1 = y, x_2 = \dot{y}, x_3 = \ddot{y}$$

where we need 3 states to represent the third order system.

We obtain

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & -1 \end{pmatrix}, B = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, C = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}, D = (0)$$

***** there are many different ways to create a state-space model to match a given (input-output) transfer function.

The corresponding Matlab function is *tf2ss*.

Note that

$$tf2ss: G(s) \rightarrow ABCD$$
$$ss2tf: ABCD \rightarrow G(s)$$

will always returns the original result. However

$$ss2tf: ABCD \rightarrow G(s)$$
$$tf2ss: G(s) \rightarrow A'B'C'D'$$

may only create an 'equivalent' state space system. The equivalence (or otherwise) of different state-space systems is something we will pick up on later in the module.

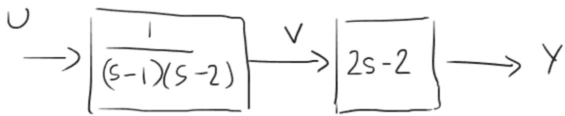
Problem – Find (by hand) the transfer function of the state-space system

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 2 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, C = \begin{pmatrix} 1 & 1 \end{pmatrix}, D = (0)$$

ANS.

$$G(s) = \frac{2s - 2}{(s-1)(s-2)}$$

Example – find (by hand) a state-space form for the above transfer function. The s-term in the numerator can cause a slight problem. One way to approach it is via this block diagram



From the first block we can obtain the state equations as before, defining $x_1 = v, x_2 = \dot{v}$:

$$V = \frac{1}{(s-1)(s-2)}U$$

so

$$(s^2 - 3s + 2)V = U$$

$$\ddot{v} - 3\dot{v} + 2v = u$$

so

$$\dot{x}_1 = x_2, \qquad \dot{x}_2 = -2x_1 + 3x_2 + u$$

and hence

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

The second block tells us $y = -2v + 2\dot{v} = -2x_1 + 2x_2$, so

$$C = \begin{pmatrix} -2 & 2 \end{pmatrix}, D = (0)$$

Problem – check (by hand or using Matlab) that this state-space system has the expected transfer function.

Summary (Sessions 1 and 2)

We have introduced state-space as a way to represent engineering systems as first-order differential equations. This normally starts by modelling a system using differential equations and introducing a complete set of state variables $x_1, x_2, \ldots, x_n$. This is then an n^{th} order system. Mostly we deal with linear systems for controller design.

For linear systems, state space and transfer functions can represent the same dynamics, but state-space looks 'inside the box' – it doesn't only compare outputs to inputs.

We can easily find the transfer function for any given SISO state-space system.

MIMO systems have a transfer function matrix (again, easily found)

Finding state-space equations from a transfer function has no unique method, and no unique solution