



# State-space models 5

## transfer functions from state space

J A Rossiter

# Introduction

The previous videos introduced state space models. It is useful to consider to what extent these can be represented by an equivalent transfer function model.

This video introduces a standard process for doing this.

# Background

A general state space model includes two vector equations.

1. One vector equation defines the system dynamics through first order derivatives (using A,B matrices).
2. The second vector equation defines the outputs  $y$  through matrices C,D. This equation has no dynamics.

$$\dot{x} = Ax + Bu; \quad y = Cx + Du$$

# Laplace Transforms

It is expected that viewers know the basics of Laplace and therefore recognise expressions such as:

$$L[x(t)] = X(s) \quad \Rightarrow \quad L\left[\frac{dx}{dt}\right] = sX(s) - x(0)$$

Here we are dealing with transfer function models and so initial conditions will be ignored.

Apply this definition to the state space model.

# Laplace transforms of model

Take Laplace of every term. Use capitals for Laplace and lower case for time domain.

$$\dot{x} = Ax + Bu; \quad y = Cx + Du$$

$$sX = AX + BU; \quad Y = CX + DU$$

Rearrange to solve for  $X(s)$  and  $Y(s)$

$$(sI - A)X = BU \quad \Rightarrow \quad \begin{aligned} X &= (sI - A)^{-1} BU \\ Y &= [C(sI - A)^{-1} B + D]U \end{aligned}$$

# Example 1

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x \end{bmatrix} = \underbrace{\begin{bmatrix} -b & -c \\ 1 & 0 \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1 \\ x \end{bmatrix}}_z + \underbrace{\begin{bmatrix} k \\ 0 \end{bmatrix}}_B f$$

$$G = (sI - A)^{-1} B = \frac{\begin{bmatrix} s & -c \\ 1 & s+b \end{bmatrix} \begin{bmatrix} k \\ 0 \end{bmatrix}}{(s+b)s+c} = \frac{\begin{bmatrix} ks \\ k \end{bmatrix}}{s^2 + bs + c}$$

# Example 2 (mass-spring-damper)

$$\begin{bmatrix} \frac{dv}{dt} \\ \frac{dx}{dt} \end{bmatrix} = \underbrace{\begin{bmatrix} -\hat{B} & -k \\ M & 0 \end{bmatrix}}_A \underbrace{\begin{bmatrix} v \\ x \end{bmatrix}}_z + \underbrace{\begin{bmatrix} 1 \\ M \\ 0 \end{bmatrix}}_B f$$

$$\begin{bmatrix} v \\ x \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_C \underbrace{\begin{bmatrix} v \\ x \end{bmatrix}}_z = Cz$$

$$G = C(sI - A)^{-1} B = \frac{\begin{bmatrix} s & -\frac{k}{M} \\ 1 & s + \frac{\hat{B}}{M} \end{bmatrix} \begin{bmatrix} 1 \\ M \\ 0 \end{bmatrix}}{\left(s + \frac{\hat{B}}{M}\right)s + \frac{k}{M}} = \frac{\begin{bmatrix} \frac{s}{M} \\ 1 \\ M \end{bmatrix}}{s^2 + \frac{\hat{B}}{M}s + \frac{k}{M}}$$

# REMARK

Finding transfer functions by hand is not recommended in general as it is rather tedious. Instead, it is recommended that students use a computer.

Some examples with MATLAB are given next.

**NOTE: MATLAB will only do one input at a time**



# MATLAB example 1

```

1 - A=[1 3;-2 5]
2 - B=[2;-1]
3 - C=[1 1]
4 - D=0
5 - [n,d]=ss2tf(A,B,C,D,1)
    
```

```

n =
    0    1.0000   -16.0000

d =
    1    -6    11
    
```

$$G = \frac{s - 16}{s^2 - 6s + 11}$$

# MATLAB example 2

```

Editor - C:\Users\uos\Documents\BOOK MASTER\book pow
EDITOR PUBLISH VIEW
FILE EDIT NAVIGATE Breakpoints Run Run and Advance
BREAKPOINTS RUN
statespace1.m
6
7 - A=[1 3 2;0 -2 5;1 -0.3
8 - B=[2;-1;0]
9 - C=[1 0 5]
10 - D=0
11 - [n,d]=ss2tf(A,B,C,D,1)
12
script
    
```

```

MATLAB R2014a
H... P... A...
New Script New Open Find Files Compare
FILE VARIABLE CODE SIMULINK ENVIRONMENT RESOURCES
<< Documents >> BOOK MASTER >> book powerpoint >> state space methods >>
New to MATLAB? Watch this Video, see Examples, or read Getting Started.
n =
      0      2.0000      8.5000      5.1000
d =
      1.0000     -1.0000     -4.5000    -16.5000
    
```

$$G = \frac{2s^2 + 8.5s + 5.1}{s^3 - s^2 - 4.5s - 16.5}$$

## REMARK 2

The denominator of the transfer function has a clear analogy to the eigenvalues of  $A$ , hence it is clear that the eigenvalues of  $A$  must be the system poles.

The denominator of a matrix inverse is the determinant.

$$Y = [C(sI - A)^{-1}B + D]U \quad \Rightarrow \quad \text{poles} \equiv |sI - A| = 0$$

$$\text{eigenvalues} \equiv |\lambda I - A| = 0$$

# Summary

Illustrated that it is straightforward in principle to find a transfer function model from a state-space model.

This is not a pen and paper exercise in general due to the requirement of a matrix inverse – software tools are recommended.

$$\dot{x} = Ax + Bu; \quad y = Cx + Du$$

$$(sI - A)X = BU \quad \Rightarrow \quad \begin{aligned} X &= (sI - A)^{-1} BU \\ Y &= [C(sI - A)^{-1} B + D]U \end{aligned}$$



Anthony Rossiter  
Department of Automatic Control and  
Systems Engineering  
University of Sheffield  
[www.shef.ac.uk/acse](http://www.shef.ac.uk/acse)

© 2016 University of Sheffield

This work is licensed under the Creative Commons Attribution 2.0 UK: England & Wales Licence. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/2.0/uk/> or send a letter to: Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.



It should be noted that some of the materials contained within this resource are subject to third party rights and any copyright notices must remain with these materials in the event of reuse or repurposing.

If there are third party images within the resource please do not remove or alter any of the copyright notices or website details shown below the image.

*(Please list details of the third party rights contained within this work.*

*If you include your institutions logo on the cover please include reference to the fact that it is a trade mark and all copyright in that image is reserved.)*