The University Of Sheffield.

State-space models 10
models from a difference equation

J A Rossiter

Slides by Anthony Rossiter

Introduction

The previous videos focussed on continuous time models.

Next consideration is given to discrete time models.

It is shown that the modelling processes are almost identical and hence some effort is used to show analogies between the two.

Consequently, details are covered relatively quickly.

The University Of Sheffield.

Discrete system

It is assumed that viewers are familiar with z-transforms and discrete models.

Without loss of generality (one can always use zero coefficients), take the numerator and denominator orders to be equal.

$$y_k + a_1 y_{k-1} + \cdots + a_n y_{k-n} = b_1 u_{k-1} + \cdots + b_n u_{k-n}$$

$$(1 + a_1 z^{-1} + \cdots + a_n z^{-n}) y(z) = (b_1 z^{-1} + \cdots + b_n z^{-n}) u(z)$$

$$(z^n + a_1 z^{n-1} + \cdots + a_n) y(z) = (b_1 z^{n-1} + \cdots + b_n) u(z)$$

The aim here is to look at state space model equivalents.

Discrete state space model

In an analogous fashion to continuous time, the key principle is to use a 1st order matrix equation to represent a high order model.

The number of states matches the model order.

$$x_k + a_1 x_{k-1} = b_1 u_{k-1}$$

First order model.

$$x_{k+1} = Ax_k + Bu_k$$

Discrete state space model

$$zX(z) = AX(z) + BU(z)$$

Using transforms

Find a control canonical form for the following (see video 6)

$$Y(s) = \frac{6s^2 + 4s + 2}{s^4 - s^3 + 2s^2 + s + 3} U(s)$$

$$\left.\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned}\right\} \equiv sX = Ax + Bu \quad \Rightarrow \quad Y(s) = C(sI - A)^{-1} BU(s)$$

$$\frac{d}{dt}[x] = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & -1 & -2 & 1 \end{bmatrix}}_{A} x + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{B} u; \quad y = \underbrace{\begin{bmatrix} 2 & 4 & 6 & 0 \end{bmatrix}}_{C} z$$

Using analogies

We could use analogies between transfer functions to show that.

$$Y(z) = \frac{6z^2 + 4z + 2}{z^4 - z^3 + 2z^2 + z + 3} U(z)$$

$$\left. \begin{array}{c} x_{k+1} = Ax_k + Bu_k \\ y = Cx \end{array} \right\} \equiv \quad zX = AX + BU \quad \Rightarrow \quad Y(z) = C(zI - A)^{-1} BU(z)$$

$$X(k+1) = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -3 & -1 & -2 & 1 \end{bmatrix}}_{A} X(k) + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{B} u; \quad y = \underbrace{\begin{bmatrix} 2 & 4 & 6 & 0 \end{bmatrix}}_{C} Z(k)$$

DEVELOPMENT FROM FIRST PRINCIPLES

State space model for a generic 2nd order difference equation

Create two 1st order difference equations by creating a new state.

$$x(k+1)+a_1x(k)+a_2x(k-1)=bu(k)$$

$$x_1(k)=x(k-1)$$

$$x(k+1)+a_1x(k)+a_2x_1(k)=bu(k)$$

$$\begin{bmatrix} x(k+1) \\ x_1(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} -a_1 & -a_2 \\ 1 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x(k) \\ x_1(k) \end{bmatrix}}_{z} + \underbrace{\begin{bmatrix} b \\ 0 \end{bmatrix}}_{B} u(k)$$

High order discrete state space model

Create n equations to form a state space model with n states.

$$u(k) = x(k+1) + a_1 x(k) + \cdots + a_{n-1} x(k-n+2) + a_n x(k-n+1)$$

$$x(k-1) = x_1(k); \quad x(k-2) = x_2(k); \quad x(k-n+1) = x_{n-1}(k);$$

$$x_1(k+1) = x(k);$$
$$x_2(k+1) = x(k-1) = x_1(k);$$
$$x_{n-1}(k+1) = x(k-n+1) = x_{n-2}(k);$$

$$u(k) = x(k+1) + a_1 x(k) + \cdots + a_{n-1} x_{n-2}(k) + a_n x_{n-1}(k)$$

Discrete state space model

$$u(k) = x(k+1) + a_1 x(k) + \cdots + a_{n-1} x_{n-2}(k) + a_n x_{n-1}(k)$$

$$x_1(k+1) = x(k);$$
$$x_2(k+1) = x_1(k);$$
$$x_{n-1}(k+1) = x_{n-2}(k);$$

$$\begin{bmatrix} x(k+1) \\ x_1(k+1) \\ \vdots \\ x_{n-1}(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} -a_1 & -a_2 & \cdots & -a_n \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x(k) \\ x_1(k) \\ \vdots \\ x_{n-1}(k) \end{bmatrix}}_{X} + \underbrace{\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{B} u(k)$$

Remark

The elements in the state vector X are all delayed versions of the underlying state 'x'.

$$\begin{bmatrix} x(k) \\ x_1(k) \\ \vdots \\ x_{n-1}(k) \end{bmatrix} = \underbrace{\begin{bmatrix} x(k) \\ x(k-1) \\ \vdots \\ x(k-n+1) \end{bmatrix}}_{X}$$

Delay of one sample
$x(k-1)=e_2^T X$

Delay of n-1 samples
$x(k-n+1)=e_n^T X$

$$e_1^T = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}; \quad e_2^T = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \end{bmatrix}$$

e_i is terminology for the standard basis set

Extension for high order numerator

The previous slide showed that:

$$Y(z) = \frac{bz^{n-1}}{z^n + a_1 z^{n-1} + \cdots + a_1 z + a_0} U(z) \Rightarrow \begin{Bmatrix} X(k+1) = AX(k) + BU(k) \\ C = \begin{bmatrix} b & 0 & \cdots & 0 \end{bmatrix} = b e_1^T \end{Bmatrix}$$

However, we also note that given the definition of the states (delayed versions of the first state):

$$C = b_i e_i^T \quad \Rightarrow \quad Y(z) \frac{b_i z^{n-i}}{z^n + a_1 z^{n-1} + \cdots + a_1 z + a_0} U(z)$$

$$C = \sum_i b_i e_i^T \quad \Rightarrow \quad Y(z) \frac{\sum_i b_i z^{n-i}}{z^n + a_1 z^{n-1} + \cdots + a_1 z + a_0} U(z)$$

EXAMPLE

$$Y(z) = \frac{2z^2 - 3z + 1}{z^4 + 0.2z^3 + 0.8z^2 + z + 0.1} U(z)$$

$$\begin{bmatrix} x(k+1) \\ x_1(k+1) \\ \vdots \\ x_{n-1}(k+1) \end{bmatrix} = \underbrace{\begin{bmatrix} -0.2 & -0.8 & -1 & -0.1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x(k) \\ x_1(k) \\ \vdots \\ x_{n-1}(k) \end{bmatrix}}_{Z(k)} + \underbrace{\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{B} u(k)$$

$$Y(k) = \begin{bmatrix} 0 & 2 & -3 & 1 \end{bmatrix} Z(k)$$

REMARKS

We will not repeat state space to transfer function for discrete systems as this is identical to video 5 with the only change being the use of ‘z’ instead of ‘s’.

We will not repeat discussion of canonical forms as again this is identical to videos 6, 7.

Use of MATLAB

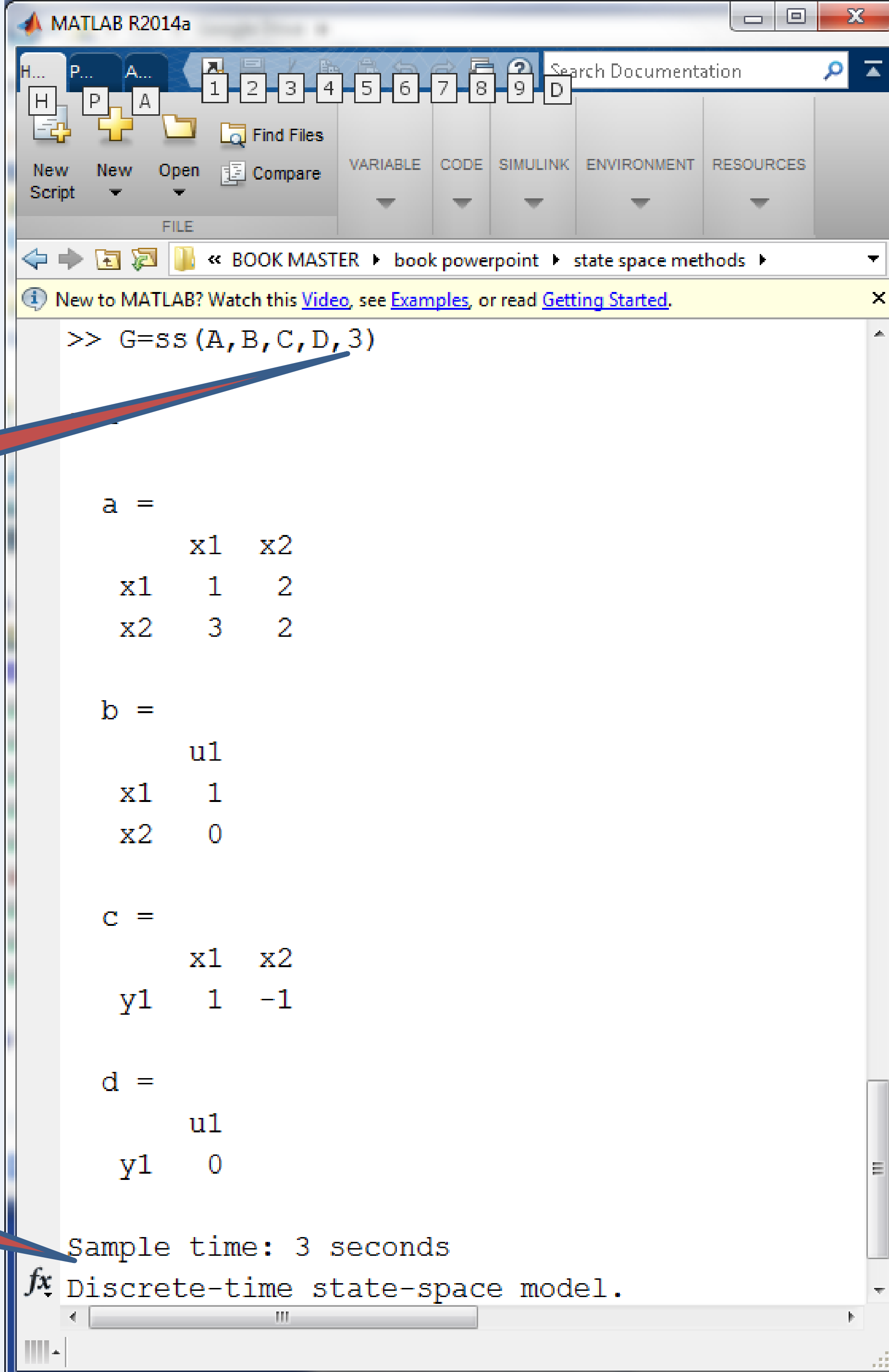
The resource on use of MATLAB carries across almost entirely with just one minor change – ensure that the models are defined as being discrete where this is necessary.

1. When using ss.m, add the sampling time and MATLAB will automatically make this discrete.
2. When using tf2ss.m, ensure the coefficients are done as powers of ‘z’ as in the examples earlier in this resource. MATLAB assumes the maximum power from the length of the vector.

The University Of Sheffield.

ss.m

Sample time

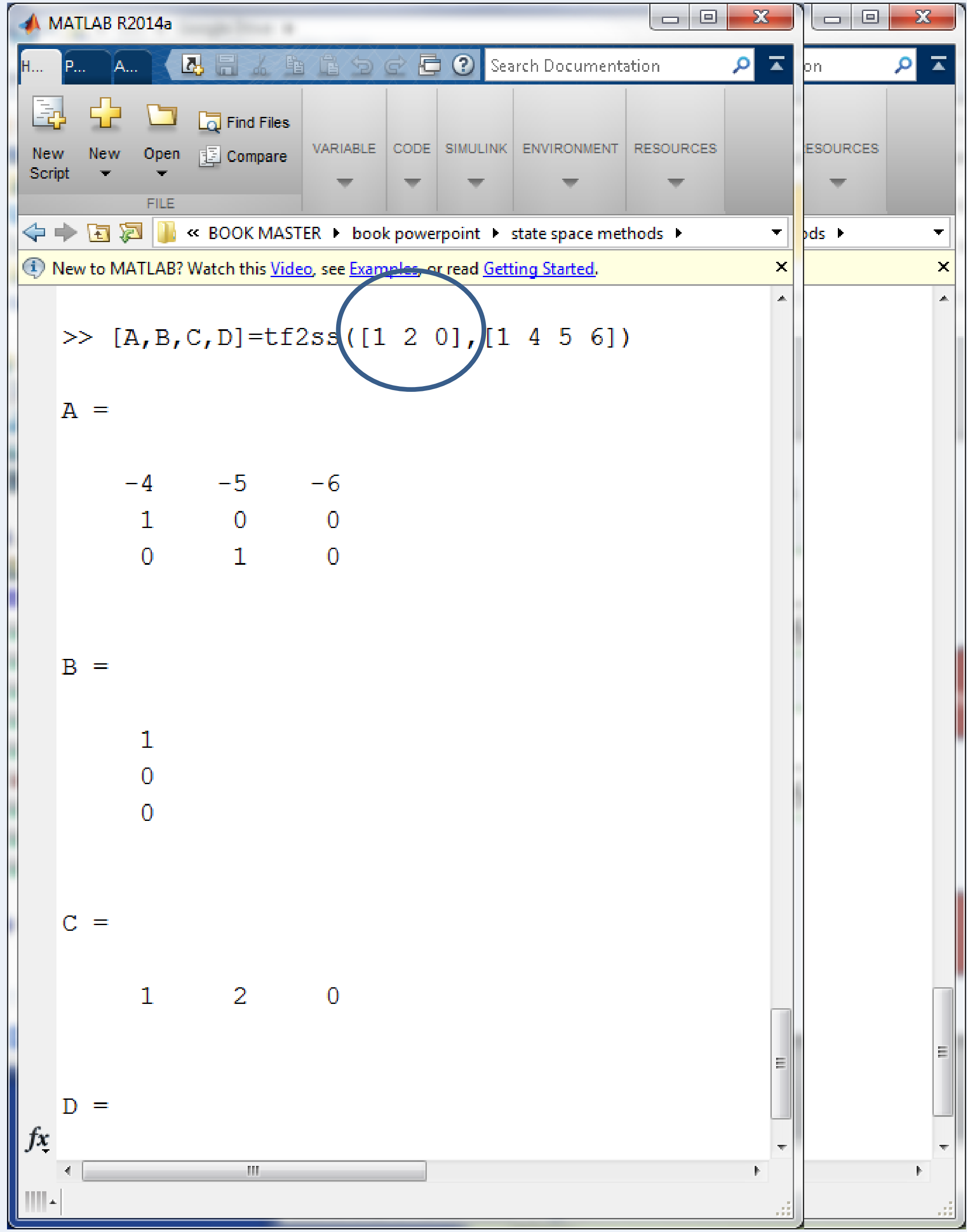


Sample time

tf2ss

$$\frac{1z+2}{z^3+4z^2+5z+6}$$

$$\frac{z^2+2z}{z^3+4z^2+5z+6}$$



Slides by Anthony Rossiter

Summary

Given a quick illustration of state space models for discrete systems.

Shown that the conversion from transfer function to state-space and vice-versa are equivalent to the mechanisms used for continuous time systems.

$$x(k+1) = Ax(k) + Bu(k)$$
$$(zI - A)X(z) = BU(z)$$

$$\dot{x} = Ax(t) + Bu(t)$$
$$(sI - A)X(s) = BU(s)$$

Anthony Rossiter
Department of Automatic Control and Systems Engineering
University of Sheffield
www.shef.ac.uk/acse