

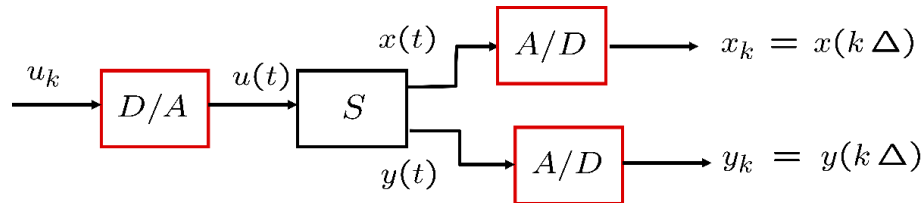
Hands-On: Step-Invariant Transform

Table of Contents

Introduction.....	1
First Scenario: an LTI System State-Space Description with Symbolic Parameters.....	2
Using MATLAB and the Symbolic Math Toolbox.....	2
Computation of the Discrete-Time Description Matrix A.....	3
Remark.....	4
An Alternative Use for the Command subs.....	4
Computing the Matrix B of the Discrete-Time Description.....	5
Computing the Matrix C and D of the Discrete-Time Description.....	6
First Scenario: Final Remarks.....	6
Second Scenario: an LTI System State-Space Description with Numerical Matrices.....	7
Summary.....	8

Introduction

Consider the scheme



where the LTI dynamic system is described by means of state equations

$$\begin{cases} \dot{x}(t) = A_c x(t) + B_c u(t) \\ y(t) = C_c x(t) + D_c u(t) \end{cases}$$

Both state variables and outputs are sampled by means of an **ideal sampler**, whereas the inputs to the system are converted from discrete- to continuous-time using a **ZOH**.

The discrete-time state equations for the dynamic system considered are

$$\begin{cases} x((k+1)\Delta) = A_d x(k\Delta) + B_d u(k\Delta) \\ y(k\Delta) = C_d x(k\Delta) + D_d u(k\Delta) \end{cases}$$

where the formulas

$$A_d = e^{A_c \Delta} \quad C_d = C_c$$

$$B_d = \int_0^{\Delta} e^{A_c r} B_c dr \quad D_d = D_c$$

describe the so called **step-invariant transform**.

In this live script, we illustrate how to use Matlab to apply the step-invariant transform.

What you will learn:

- How to apply the step-invariant transform, when the continuous-time system has a state space description using some symbolic parameters (so you need some symbolic variables in MATLAB to describe such an LTI system);
- How to apply the step-invariant transform when the continuous-time LTI system state equations have been used to create an `LTI system` object in MATLAB.

First Scenario: an LTI System State-Space Description with Symbolic Parameters

Consider

$$\begin{cases} \dot{x} &= \begin{bmatrix} 0 & 1 \\ 0 & -a \end{bmatrix} x + \begin{bmatrix} 0 \\ K \end{bmatrix} u \\ y &= [1 \quad 0] x \end{cases} \quad a \in \mathbf{R}, K \in \mathbf{R} \quad (1)$$

and let's determine the discrete-time description, by sampling with ZOH and ideal samplers.

Applying the step-invariant transform

$$A_d = e^{A_c \Delta} = e^{A_c t} \Big|_{t=\Delta} = \mathcal{L}^{-1} \{ (sI - A_c)^{-1} \} \Big|_{t=\Delta} = \begin{bmatrix} 1 & \frac{1}{a}(1 - e^{-a\Delta}) \\ 0 & e^{-a\Delta} \end{bmatrix} \quad (2)$$

$$B_d = \int_0^{\Delta} e^{A_c r} B_c dr = \begin{bmatrix} \int_0^{\Delta} \frac{K}{a}(1 - e^{-ar}) dr \\ \int_0^{\Delta} K e^{-ar} dr \end{bmatrix}$$

$$C_d = C_c \quad D_d = D_c$$

Using MATLAB and the Symbolic Math Toolbox

```
clear
```

```
close all
clc

sympref('HeavisideAtOrigin', 1); % setting the preferences
                                % for the Symbolic Math Toolbox
sympref('FloatingPointOutput', 0);
sympref % show the actual settings
```

```
ans = struct with fields:
    FourierParameters: [1    -1]
    HeavisideAtOrigin: 1
    AbbreviateOutput: 1
    TypesetOutput: 1
    FloatingPointOutput: 0
    PolynomialDisplayStyle: 'default'
    MatrixWithSquareBrackets: 0
```

Declare the parameters a and K as symbolic variables

```
syms a K
```

Then assign the system matrices

```
Ac = [0 , 1; ...
      0 , -a]
```

```
Ac =

$$\begin{pmatrix} 0 & 1 \\ 0 & -a \end{pmatrix}$$

```

```
Bc = [0; ...
      K]
```

```
Bc =

$$\begin{pmatrix} 0 \\ K \end{pmatrix}$$

```

```
Cc = sym([1 , 0]), Dc = sym(0)
```

```
Cc = (1 0)
Dc = 0
```

Declare the symbolic variables s , t , and the sampling time Δ . We need these symbolic variables to compute the symbolic expressions of the discrete-time representation for the LTI system, according to the step-invariant transform, reported in [Eq. \(2\)](#).

```
syms s t Delta
```

Computation of the Discrete-Time Description Matrix A

Compute the matrix $A_d = e^{A_c \Delta}$, according to [Eq. \(2\)](#).

In order to evaluate the matrix exponential, start computing the Laplace transform of it:

$$sIA = s \cdot \text{eye}(\text{size}(Ac)) - Ac$$

$$sIA =$$

$$\begin{pmatrix} s & -1 \\ 0 & a + s \end{pmatrix}$$

$$sIA_{\text{inv}} = \text{inv}(sIA)$$

$$sIA_{\text{inv}} =$$

$$\begin{pmatrix} \frac{1}{s} & \frac{1}{s(a+s)} \\ 0 & \frac{1}{a+s} \end{pmatrix}$$

Now, determine the matrix exponential, by means of the inverse Laplace transform:

$$e^{A_c t} = \mathcal{L}^{-1}\{(sI - A_c)^{-1}\}$$

$$\text{expAc}_t = \text{ilaplace}(sIA_{\text{inv}}, s, t)$$

$$\text{expAc}_t =$$

$$\begin{pmatrix} 1 & \frac{1}{a} - \frac{e^{-at}}{a} \\ 0 & e^{-at} \end{pmatrix}$$

Finally, evaluate the symbolic expression $e^{A_c t}$ at $t = \Delta$

$$Ad = \text{simplify}(\text{subs}(\text{expAc}_t, t, \text{Delta}))$$

$$Ad =$$

$$\begin{pmatrix} 1 & -\frac{e^{-\Delta a} - 1}{a} \\ 0 & e^{-\Delta a} \end{pmatrix}$$

We found the expression of the matrix A_d . Please compare with the theoretical expression in the [Eq.\(2\)](#).

Remark

For details on the command `simplify`, please use the `help` or the `doc` commands.

An Alternative Use for the Command `subs`

What if we would assign a numerical value to the parameter a , and/or to the sampling time Δ ?

Easy! We can use the command `subs`.

Example: assign the values $a = 10$

```
AdEX1 = subs(Ad,a,10)
```

$$\text{AdEX1} = \begin{pmatrix} 1 & \frac{1}{10} - \frac{e^{-10\Delta}}{10} \\ 0 & e^{-10\Delta} \end{pmatrix}$$

Now suppose we want assign the values $a = 4.75$, and as sampling time $\Delta = 10^{-3}$ s

```
AdEX2 = subs(Ad,{a,Delta},{4.75, 1e-3})
```

$$\text{AdEX2} = \begin{pmatrix} 1 & \frac{4}{19} - \frac{4e^{-\frac{19}{4000}}}{19} \\ 0 & e^{-\frac{19}{4000}} \end{pmatrix}$$

Computing the Matrix B of the Discrete-Time Description

We should evaluate

$$B_d = \int_0^{\Delta} e^{A_c r} B_c dr$$

```
Bd = int(expAc_t*Bc, t, 0, Delta)
```

$$\text{Bd} = \begin{pmatrix} \frac{K(e^{-\Delta a} + \Delta a - 1)}{a^2} \\ -\frac{K(e^{-\Delta a} - 1)}{a} \end{pmatrix}$$

Then we can use the command `subs` to assign numerical values to the parameters and to the sampling period.
For example

```
BdEX2 = subs(Bd,{a,K,Delta},{4.75,-2,1e-3})
```

$$\text{BdEX2} = \begin{pmatrix} \frac{3981}{45125} - \frac{32e^{-\frac{19}{4000}}}{361} \\ \frac{8e^{-\frac{19}{4000}}}{19} - \frac{8}{19} \end{pmatrix}$$

Computing the Matrix C and D of the Discrete-Time Description

Easily, according to [Eq. \(2\)](#)

$$C_d = C_c \quad D_d = D_c$$

Thus

```
Cd = Cc
```

```
cd = (1 0)
```

```
Dd = Dc
```

```
Dd = 0
```

First Scenario: Final Remaks

Please note: the resulting matrices A_d , B_d , C_d , and D_d are symbolic variables, so you can not create an LTI object using such matrices. In order to create an LTI object, you have to convert the symbolic matrices in numeric ones:

```
Adr = double(AdEX2)
```

```
Adr = 2x2
    1.0000    0.0010
         0    0.9953
```

```
Bdr = double(BdEX2)
```

```
Bdr = 2x1
   -0.0000
   -0.0020
```

```
Cdr = double(Cd)
```

```
Cdr = 1x2
     1     0
```

```
Ddr = double(Dd)
```

```
Ddr = 0
```

Finally

```
sysD_EX2 = ss(Adr, Bdr, Cdr, Ddr, 1e-3)
```

```
sysD_EX2 =
```

```
A =
      x1      x2
x1      1  0.0009976
x2      0  0.9953
```

```

B =
           u1
x1  -9.984e-07
x2  -0.001995

```

```

C =
      x1  x2
y1    1    0

```

```

D =
      u1
y1    0

```

Sample time: 0.001 seconds
Discrete-time state-space model.
Model Properties

Second Scenario: an LTI System State-Space Description with Numerical Matrices

Let's consider the continuous-time LTI system

$$\begin{cases} \dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} \cdot x(t) + \begin{bmatrix} 0 \\ 10 \end{bmatrix} \cdot u(t) \\ y(t) = [1 \ 0] \cdot x(t) \end{cases}$$

Please note: it is the same example as in [Eq. \(1\)](#), with parameters values $a = 1$, $K = 10$.

```

clear
close all
clc
sympref('default'); % resetting the Symbolic Math Toolbox preferences

% assigning the numeric matrices Ac, Bc, Cc, Dc
Ac = [0, 1;...
      0, -1];
Bc = [0; 10];
Cc = [1, 0]; Dc = 0;
sysC = ss(Ac, Bc, Cc, Dc)

```

sysC =

```

A =
      x1  x2
x1    0    1
x2    0   -1

```

```

B =
      u1
x1    0
x2   10

```

```

C =
      x1  x2

```

```

    y1    1    0

D =
      u1
    y1    0

Continuous-time state-space model.
Model Properties

```

In this scenario, the matrices A_c , B_c , C_c , and D_c are numeric matrices, so we can compute the step-invariant transform easily using the Control System Toolbox command `c2d`. Chosen as sampling period the value $\Delta = 10^{-1}$ s, the discrete-time description is computed simply by typing

```

Delta = 1e-1; % assigning the sampling period
sysD_ExN = c2d(sysC, Delta)

```

```

sysD_ExN =

A =
      x1      x2
    x1      1  0.09516
    x2      0  0.9048

B =
      u1
    x1  0.04837
    x2  0.9516

C =
      x1  x2
    y1   1   0

D =
      u1
    y1   0

Sample time: 0.1 seconds
Discrete-time state-space model.
Model Properties

```

Summary

In this live script, we illustrated how to use Matlab to apply the step-invariant transform in order to obtain a discrete-time description of a continuous-time LTI system, fed by inputs converted from discrete- to continuous-time using a ZOH and sampling both state variables and outputs through an ideal sampler.

Using this live script you have learnt:

- How to apply the step-invariant transform, when the continuous-time system has a state space description using some symbolic parameters;

- How to apply the step-invariant transform when the continuous-time LTI system state equations have been used to create an `LTI_system` object in MATLAB.