自动控制原理 II：线性系统分析与设计课内实验

系统模型部分

概述：本部分实验主要内容

- MATLAB简单介绍
- 线性系统各类数学模型的表示（重点）
- 线性系统各种数学模型时间的相互转化（重点）
- 子系统的连接合并

概述：重点&预备知识

- 重点
  - ❑ 线性系统各类数学模型的表示与相互转化

- 预备知识
  - ❑ MATLAB基本知识：界面、常用操作、等
  - ❑ 线性系统理论知识：传递函数（阵）、状态空间表达式、坐标变换、等

课内实验：模型部分

MATLAB使用简介

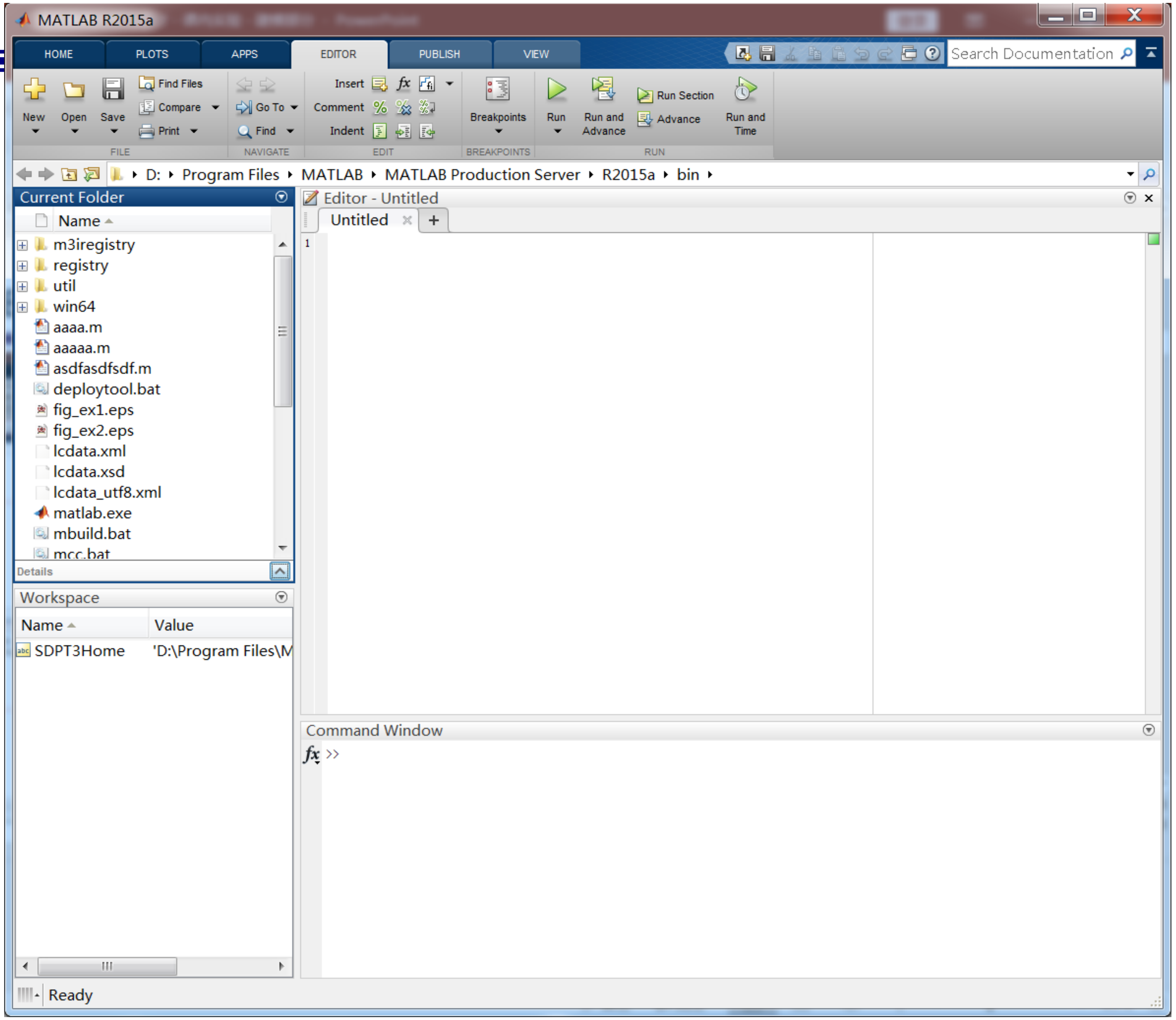
❑ 各部分功能

➢ 在哪里写代码？

➢ 如何进行调试？

➢ 在哪里看结果？

➢ 相关文件保存在哪里？



Current Folder:

m3iregistry
registry
util
win64
aaaa.m
aaaaa.m
asdfasdfsdf.m
deploytool.bat
fig_ex1.eps
fig_ex2.eps
lcdata.xml
lcdata.xsd
lcdata_utf8.xml
matlab.exe
mbuild.bat
mcc.bat

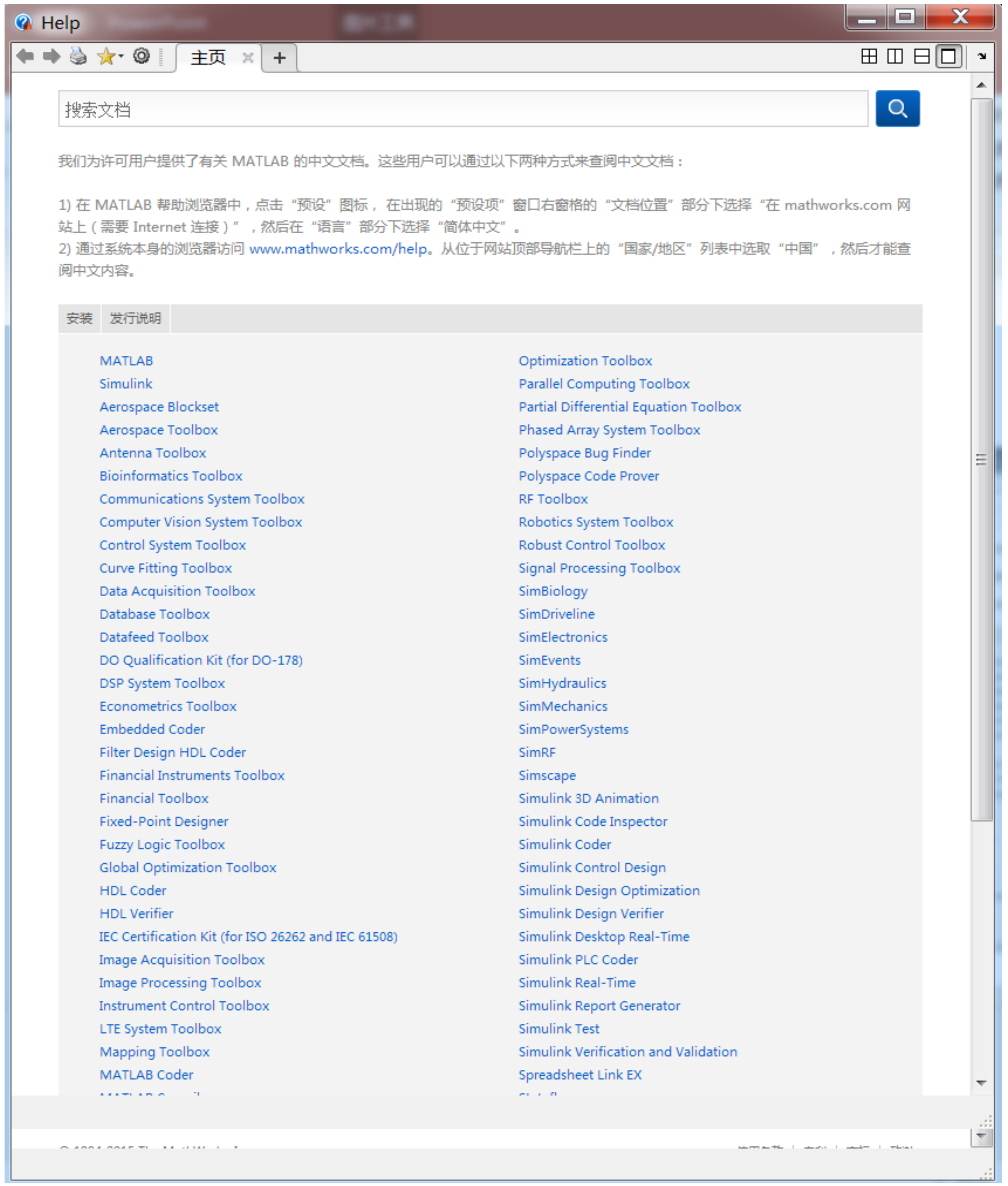
Workspace:

Name	Value
SDPT3Home	'D:\Program Files\M

```
fx >>
```

MATLAB使用简介

❑ 学习工具



我们为许可用户提供了有关 MATLAB 的中文文档。这些用户可以通过以下两种方式来查阅中文文档：

1) 在 MATLAB 帮助浏览器中，点击 “预设” 图标， 在出现的 “预设项” 窗口右窗格的 “文档位置” 部分下选择 “在 mathworks.com 网站上（需要 Internet 连接）” ，然后在 “语言” 部分下选择 “简体中文” 。
2) 通过系统本身的浏览器访问 www.mathworks.com/help。从位于网站顶部导航栏上的 “国家/地区” 列表中选取 “中国” ，然后才能查阅中文内容。

安装 发行说明

- MATLAB
- Simulink
- Aerospace Blockset
- Aerospace Toolbox
- Antenna Toolbox
- Bioinformatics Toolbox
- Communications System Toolbox
- Computer Vision System Toolbox
- Control System Toolbox
- Curve Fitting Toolbox
- Data Acquisition Toolbox
- Database Toolbox
- Datafeed Toolbox
- DO Qualification Kit (for DO-178)
- DSP System Toolbox
- Econometrics Toolbox
- Embedded Coder
- Filter Design HDL Coder
- Financial Instruments Toolbox
- Financial Toolbox
- Fixed-Point Designer
- Fuzzy Logic Toolbox
- Global Optimization Toolbox
- HDL Coder
- HDL Verifier
- IEC Certification Kit (for ISO 26262 and IEC 61508)
- Image Acquisition Toolbox
- Image Processing Toolbox
- Instrument Control Toolbox
- LTE System Toolbox
- Mapping Toolbox
- MATLAB Coder

- Optimization Toolbox
- Parallel Computing Toolbox
- Partial Differential Equation Toolbox
- Phased Array System Toolbox
- Polyspace Bug Finder
- Polyspace Code Prover
- RF Toolbox
- Robotics System Toolbox
- Robust Control Toolbox
- Signal Processing Toolbox
- SimBiology
- SimDriveline
- SimElectronics
- SimEvents
- SimHydraulics
- SimMechanics
- SimPowerSystems
- SimRF
- Simscape
- Simulink 3D Animation
- Simulink Code Inspector
- Simulink Coder
- Simulink Control Design
- Simulink Design Optimization
- Simulink Design Verifier
- Simulink Desktop Real-Time
- Simulink PLC Coder
- Simulink Real-Time
- Simulink Report Generator
- Simulink Test
- Simulink Verification and Validation
- Spreadsheet Link EX

课内实验：模型部分

线性系统数学模型表示

机理、辨识

高阶微分方程模型

传递函数模型

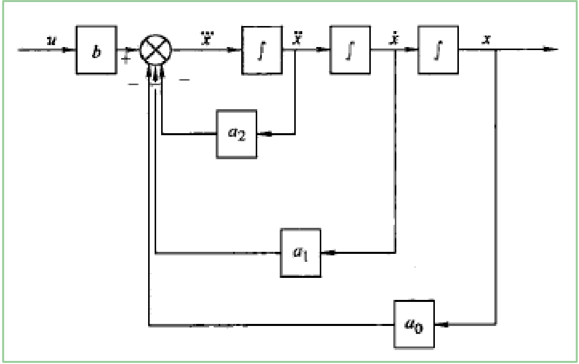
$$W(s)=\frac{b_m s^m + b_{m-1}s^{m-1}+\cdots+b_1 s+b_0}{s^n+a_{n-1}s^{n-1}+\cdots+a_1 s+a_0}$$

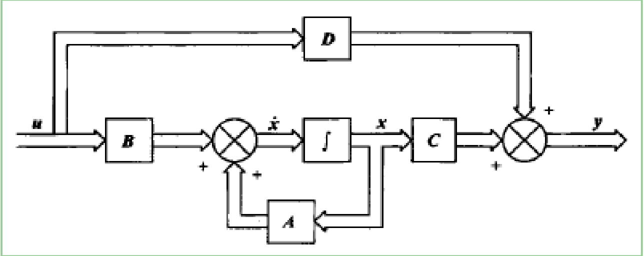
$$W(s)=k\frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)}$$

状态空间模型

$$\begin{cases}\dot{\boldsymbol{x}}(t)=\boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{B}\boldsymbol{u}(t)\\ \boldsymbol{y}(t)=\boldsymbol{C}\boldsymbol{x}(t)+\boldsymbol{D}\boldsymbol{u}(t)\end{cases}$$

结构图模型





线性系统数学模型表示

$$W(s)=\frac{b_m s^m + b_{m-1}s^{m-1}+\cdots+b_1 s+b_0}{s^n+a_{n-1}s^{n-1}+\cdots+a_1 s+a_0}$$

tf()函数

功能：建立系统的传递函数模型

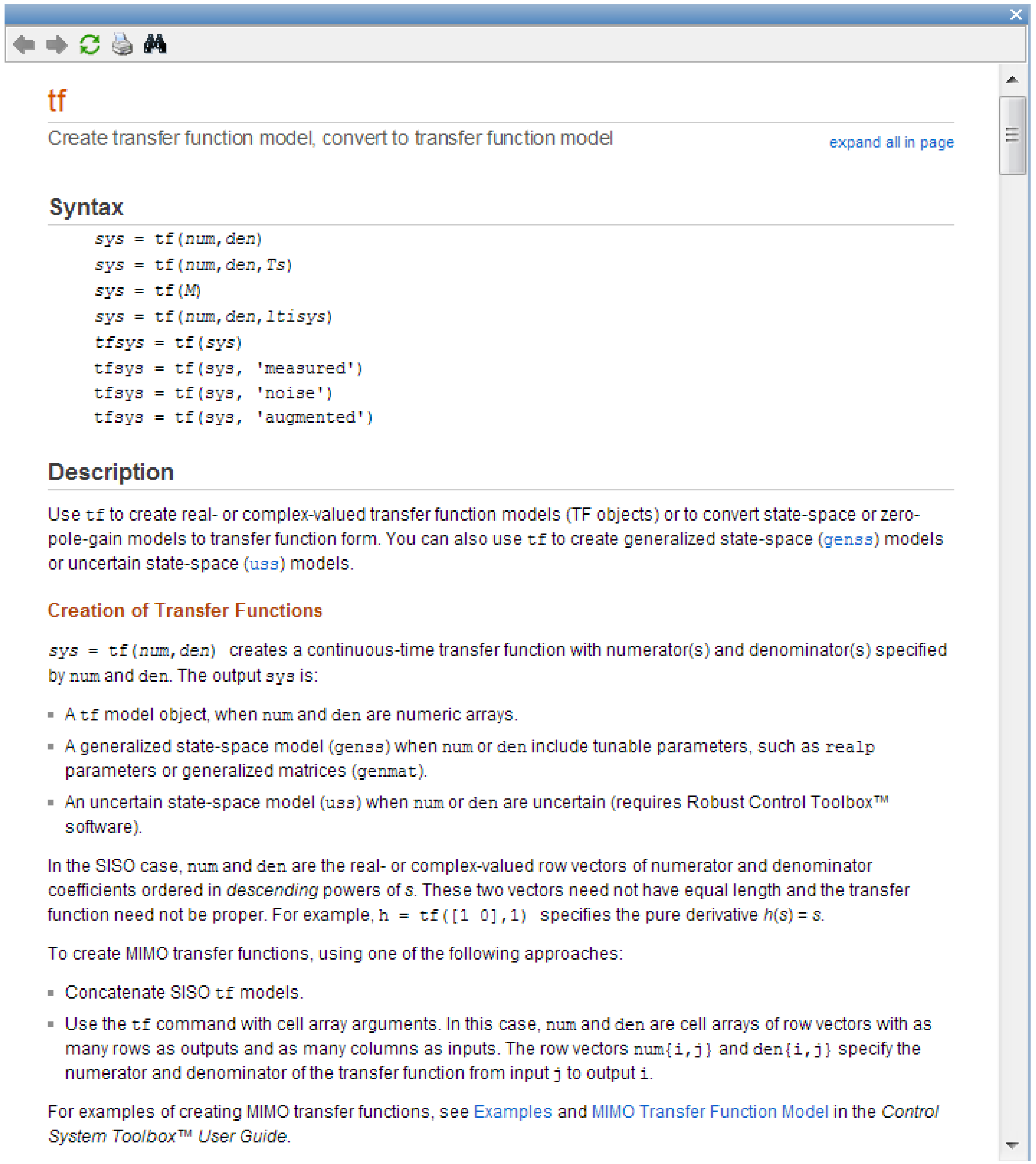
格式：

sys = tf(num,den)

其中，

$$num=[b_m, b_{m-1}, \cdots, b_1, b_0]$$

$$den=[1, a_{n-1}, a_{n-2}, \cdots, a_1, a_0]$$



tf

Create transfer function model, convert to transfer function model

expand all in page

Syntax

```
sys = tf(num,den)
sys = tf(num,den,Ts)
sys = tf(M)
sys = tf(num,den,ltisys)
tfsys = tf(sys)
tfsys = tf(sys, 'measured')
tfsys = tf(sys, 'noise')
tfsys = tf(sys, 'augmented')
```

Description

Use `tf` to create real- or complex-valued transfer function models (TF objects) or to convert state-space or zero-pole-gain models to transfer function form. You can also use `tf` to create generalized state-space (`genss`) models or uncertain state-space (`uss`) models.

Creation of Transfer Functions

`sys = tf(num,den)` creates a continuous-time transfer function with numerator(s) and denominator(s) specified by `num` and `den`. The output `sys` is:

- A `tf` model object, when `num` and `den` are numeric arrays.
- A generalized state-space model (`genss`) when `num` or `den` include tunable parameters, such as `realp` parameters or generalized matrices (`genmat`).
- An uncertain state-space model (`uss`) when `num` or `den` are uncertain (requires Robust Control Toolbox™ software).

In the SISO case, `num` and `den` are the real- or complex-valued row vectors of numerator and denominator coefficients ordered in *descending* powers of *s*. These two vectors need not have equal length and the transfer function need not be proper. For example, `h = tf([1 0],1)` specifies the pure derivative *h*(*s*) = *s*.

To create MIMO transfer functions, using one of the following approaches:

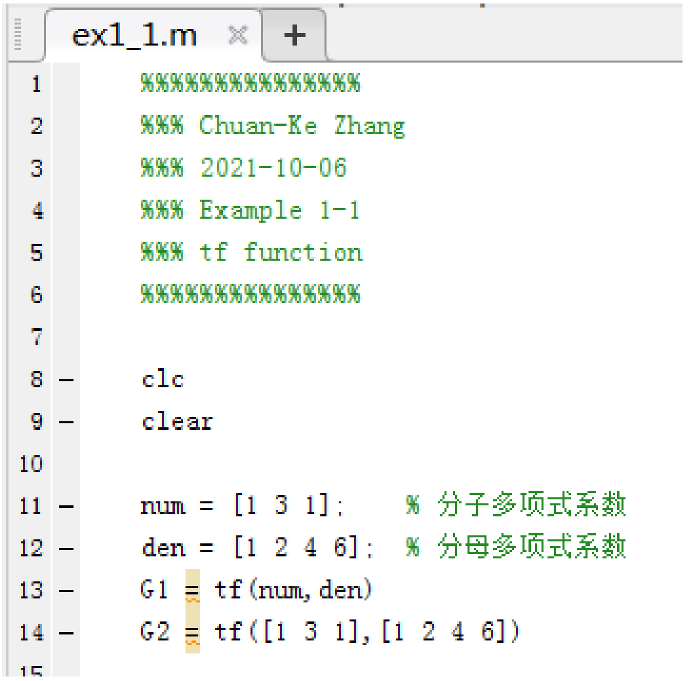
- Concatenate SISO `tf` models.
- Use the `tf` command with cell array arguments. In this case, `num` and `den` are cell arrays of row vectors with as many rows as outputs and as many columns as inputs. The row vectors `num{i,j}` and `den{i,j}` specify the numerator and denominator of the transfer function from input `j` to output `i`.

For examples of creating MIMO transfer functions, see Examples and MIMO Transfer Function Model in the *Control System Toolbox™ User Guide*.

线性系统数学模型表示

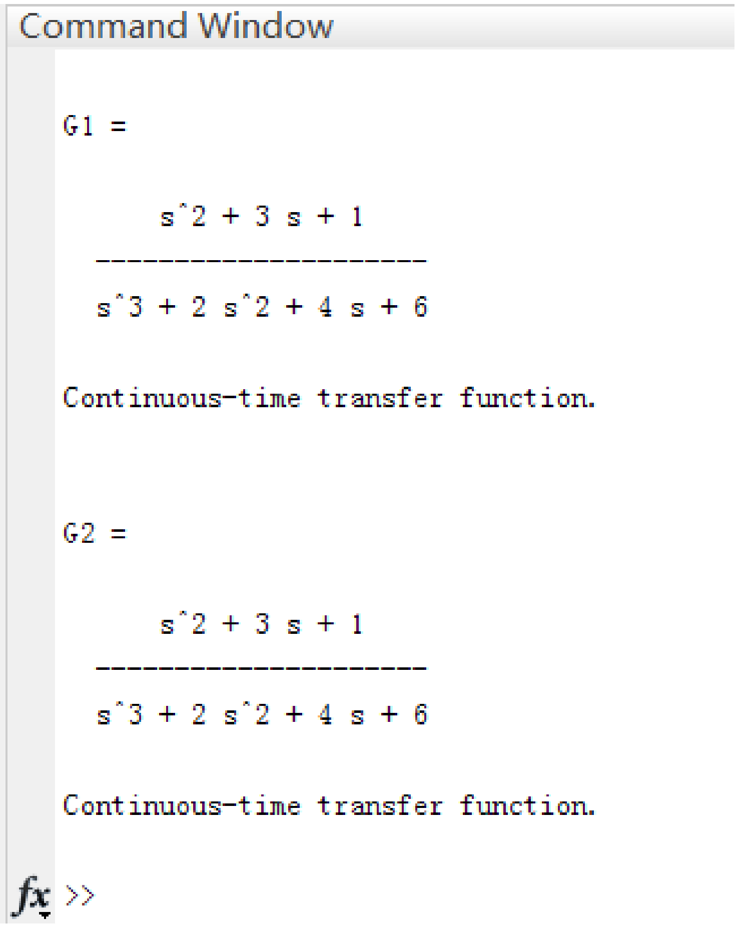
例1-1：试用Matlab描述如下系统模型

$$W(s)=\frac{s^2+3s+1}{s^3+2s^2+4s+6}$$



ex1_1.m

```
%%%%%%%%%%%%%%%%%
%%% Chuan-Ke Zhang
%%% 2021-10-06
%%% Example 1-1
%%% tf function
%%%%%%%%%%%%%%%%%

clc
clear

num = [1 3 1];     % 分子多项式系数
den = [1 2 4 6];   % 分母多项式系数
G1 = tf(num, den)
G2 = tf([1 3 1],[1 2 4 6])
```



Command Window

```
G1 =

       s^2 + 3 s + 1
  ---------------------
  s^3 + 2 s^2 + 4 s + 6

Continuous-time transfer function.


G2 =

       s^2 + 3 s + 1
  ---------------------
  s^3 + 2 s^2 + 4 s + 6

Continuous-time transfer function.

>>
```

线性系统数学模型表示

$$W(s)=k\frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)}$$

zpk()函数

功能：建立系统的传递函数模型

格式：

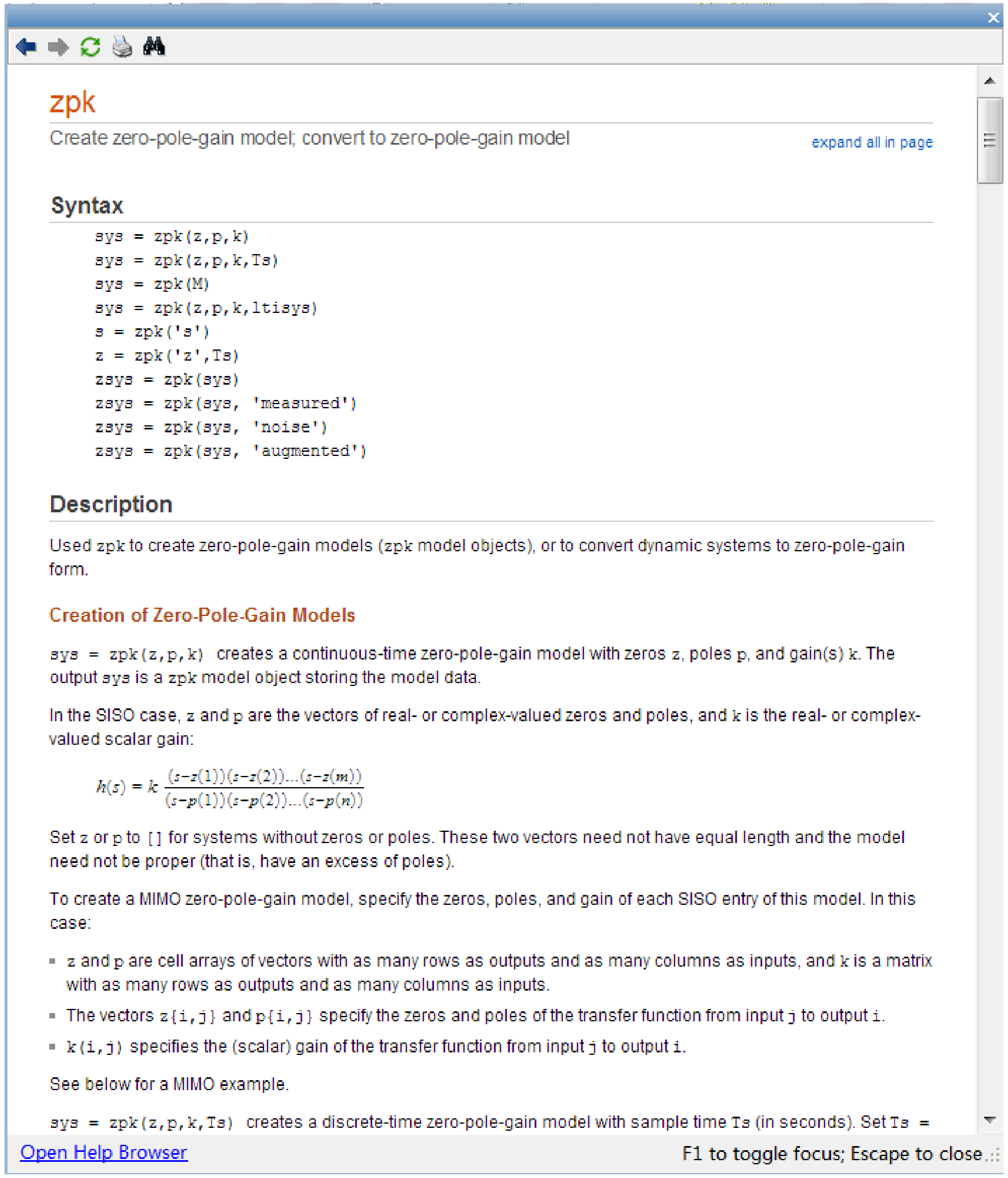
sys = zpk(z,p,k)

其中，

$$\boldsymbol{z}=[z_1,z_2,\cdots,z_m]$$

$$\boldsymbol{p}=[p_1,p_2,\cdots,p_n]$$

$$k=k$$



zpk

Create zero-pole-gain model; convert to zero-pole-gain model

expand all in page

Syntax

```
sys = zpk(z,p,k)
sys = zpk(z,p,k,Ts)
sys = zpk(M)
sys = zpk(z,p,k,ltisys)
s = zpk('s')
z = zpk('z',Ts)
zsys = zpk(sys)
zsys = zpk(sys, 'measured')
zsys = zpk(sys, 'noise')
zsys = zpk(sys, 'augmented')
```

Description

Used zpk to create zero-pole-gain models (zpk model objects), or to convert dynamic systems to zero-pole-gain form.

Creation of Zero-Pole-Gain Models

`sys = zpk(z,p,k)` creates a continuous-time zero-pole-gain model with zeros z, poles p, and gain(s) k. The output sys is a zpk model object storing the model data.

In the SISO case, z and p are the vectors of real- or complex-valued zeros and poles, and k is the real- or complex-valued scalar gain:

$$h(s)=k\frac{(s-z(1))(s-z(2))\ldots(s-z(m))}{(s-p(1))(s-p(2))\ldots(s-p(n))}$$

Set z or p to [] for systems without zeros or poles. These two vectors need not have equal length and the model need not be proper (that is, have an excess of poles).

To create a MIMO zero-pole-gain model, specify the zeros, poles, and gain of each SISO entry of this model. In this case:

- z and p are cell arrays of vectors with as many rows as outputs and as many columns as inputs, and k is a matrix with as many rows as outputs and as many columns as inputs.
- The vectors z{i,j} and p{i,j} specify the zeros and poles of the transfer function from input j to output i.
- k(i,j) specifies the (scalar) gain of the transfer function from input j to output i.

See below for a MIMO example.

`sys = zpk(z,p,k,Ts)` creates a discrete-time zero-pole-gain model with sample time Ts (in seconds). Set Ts =

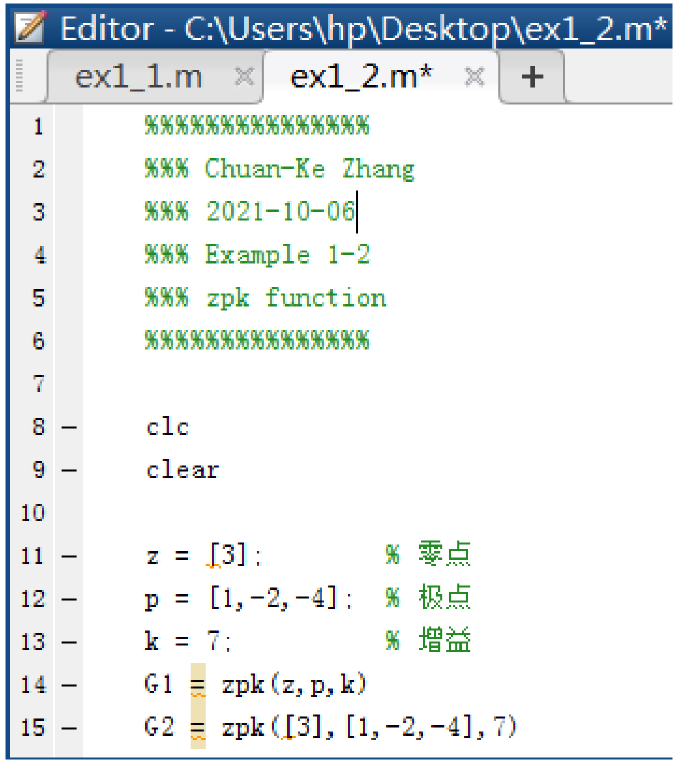
Open Help Browser

F1 to toggle focus; Escape to close

线性系统数学模型表示

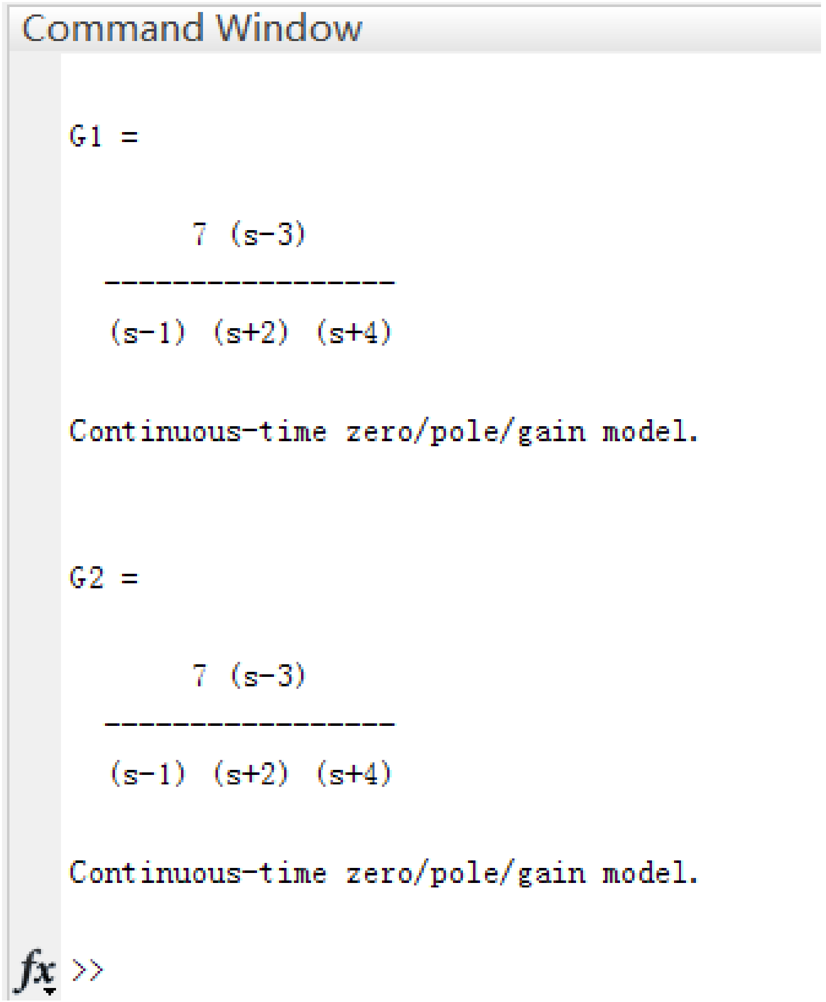
例1-2：试用Matlab描述如下系统模型

$$W(s)=\frac{7(s-3)}{(s-1)(s+2)(s+4)}$$


Editor - C:\Users\hp\Desktop\ex1_2.m*

ex1_1.m ex1_2.m*

```
%%%%%%%%%%%%%%%%
%%% Chuan-Ke Zhang
%%% 2021-10-06
%%% Example 1-2
%%% zpk function
%%%%%%%%%%%%%%%%

clc
clear

z = [3];          % 零点
p = [1,-2,-4];    % 极点
k = 7;            % 增益
G1 = zpk(z,p,k)
G2 = zpk([3],[1,-2,-4],7)
```


Command Window

```
G1 =

       7 (s-3)
  -----------------
  (s-1) (s+2) (s+4)

Continuous-time zero/pole/gain model.


G2 =

       7 (s-3)
  -----------------
  (s-1) (s+2) (s+4)

Continuous-time zero/pole/gain model.

fx >>
```

线性系统数学模型表示

$$\begin{cases}\dot{\boldsymbol{x}}(t)=\boldsymbol{Ax}(t)+\boldsymbol{Bu}(t)\\ \boldsymbol{y}(t)=\boldsymbol{Cx}(t)+\boldsymbol{Du}(t)\end{cases}$$

ss()函数

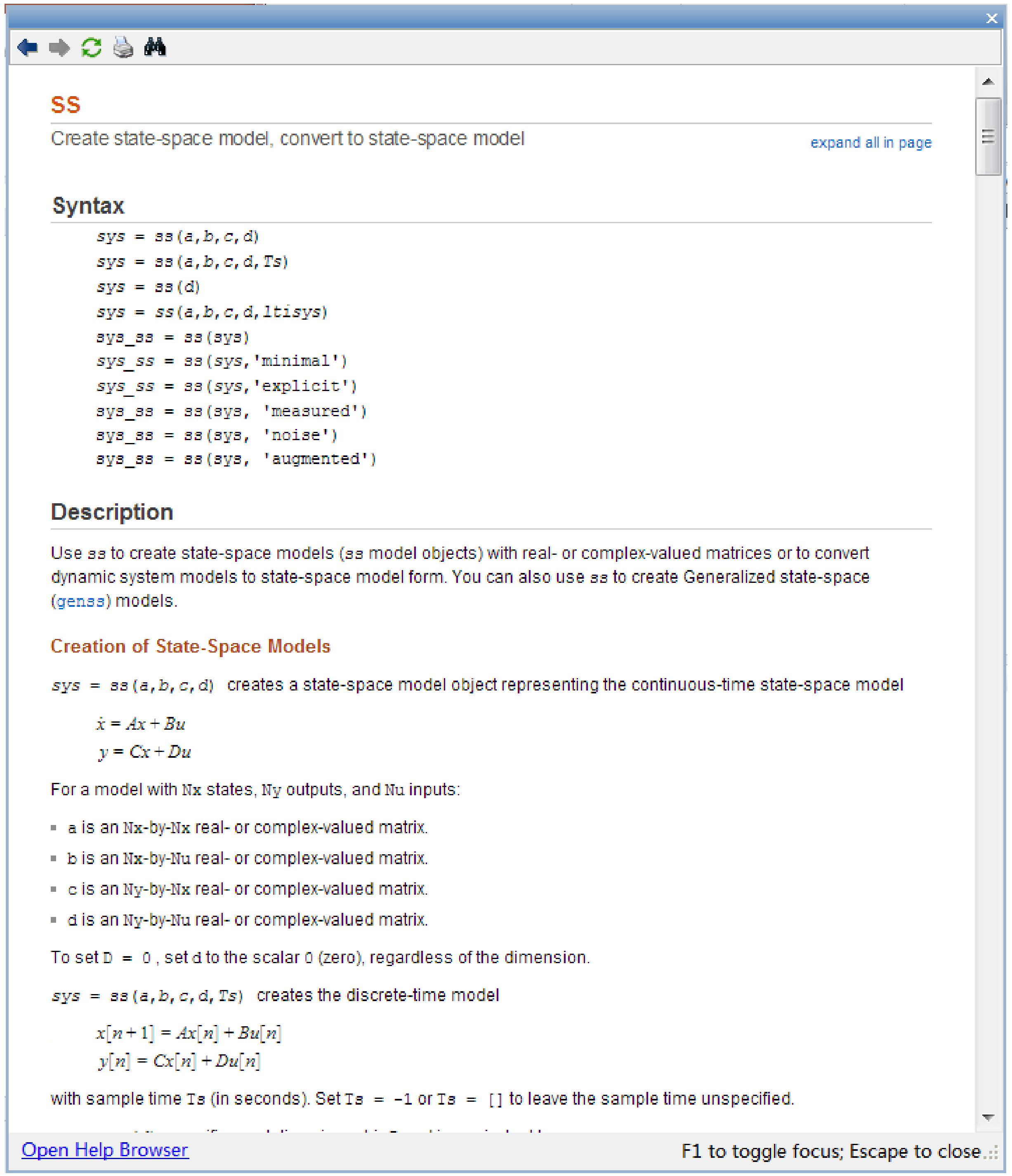
功能：建立系统的状态空间模型

格式：

sys = ss(A,B,C,D)

其中,

$$\boldsymbol{A}=[a_{11},a_{12},\cdots,a_{1n};a_{21},a_{22},\cdots,a_{2n};\cdots;a_{n1},a_{n2},\cdots,a_{nn}]$$
$$\boldsymbol{B}=[b_{11},b_{12},\cdots,b_{1n};b_{21},b_{22},\cdots,b_{2n};\cdots;b_{n1},b_{n2},\cdots,b_{nr}]$$
$$\boldsymbol{C}=[c_{11},c_{12},\cdots,c_{1n};c_{21},c_{22},\cdots,c_{2n};\cdots;c_{m1},c_{m2},\cdots,c_{mn}]$$
$$\boldsymbol{D}=[d_{11},d_{12},\cdots,d_{1r};d_{21},d_{22},\cdots,d_{2r};\cdots;d_{m1},d_{m2},\cdots,d_{mr}]$$



ss

Create state-space model, convert to state-space model

expand all in page

Syntax

```
sys = ss(a,b,c,d)
sys = ss(a,b,c,d,Ts)
sys = ss(d)
sys = ss(a,b,c,d,ltisys)
sys_ss = ss(sys)
sys_ss = ss(sys,'minimal')
sys_ss = ss(sys,'explicit')
sys_ss = ss(sys, 'measured')
sys_ss = ss(sys, 'noise')
sys_ss = ss(sys, 'augmented')
```

Description

Use ss to create state-space models (ss model objects) with real- or complex-valued matrices or to convert dynamic system models to state-space model form. You can also use ss to create Generalized state-space (genss) models.

Creation of State-Space Models

`sys = ss(a,b,c,d)` creates a state-space model object representing the continuous-time state-space model

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

For a model with Nx states, Ny outputs, and Nu inputs:

- a is an Nx-by-Nx real- or complex-valued matrix.
- b is an Nx-by-Nu real- or complex-valued matrix.
- c is an Ny-by-Nx real- or complex-valued matrix.
- d is an Ny-by-Nu real- or complex-valued matrix.

To set `D = 0` , set d to the scalar 0 (zero), regardless of the dimension.

`sys = ss(a,b,c,d,Ts)` creates the discrete-time model

$$x[n+1]=Ax[n]+Bu[n]$$
$$y[n]=Cx[n]+Du[n]$$

with sample time Ts (in seconds). Set `Ts = -1` or `Ts = []` to leave the sample time unspecified.

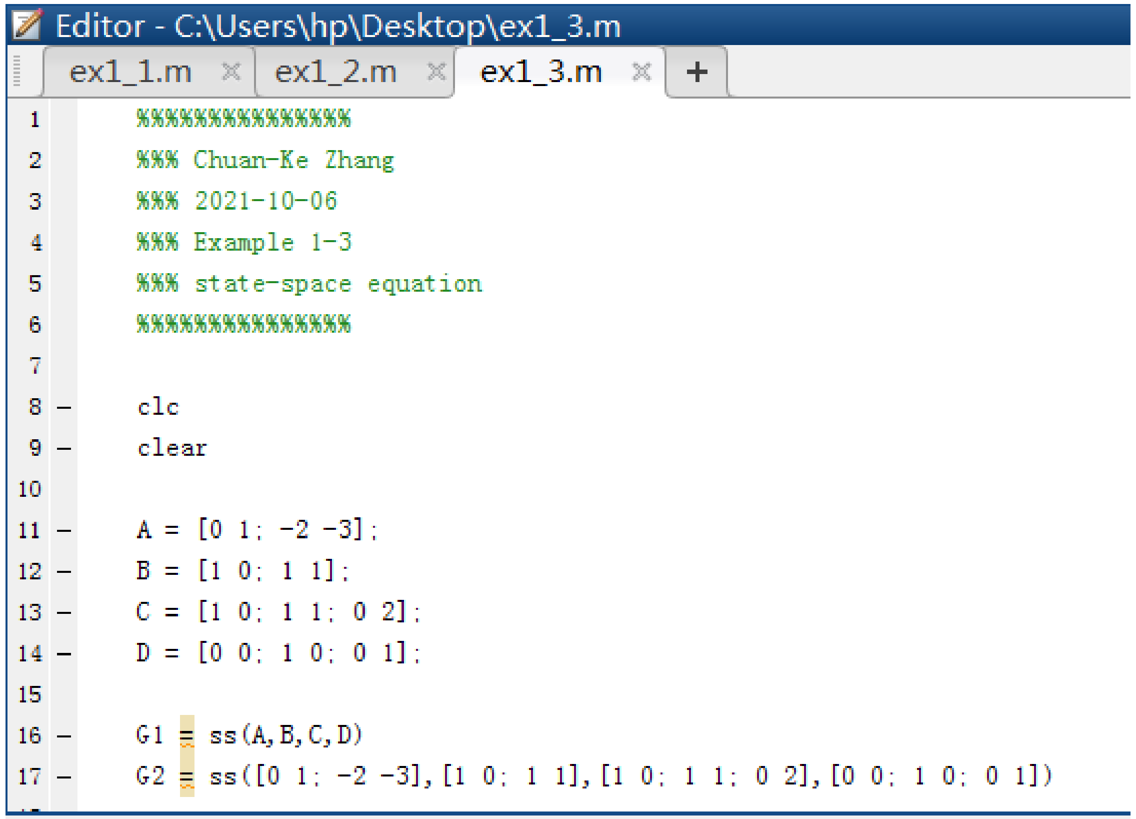
Open Help Browser

F1 to toggle focus; Escape to close

线性系统数学模型表示

例1-3：试用Matlab描述如下系统模型

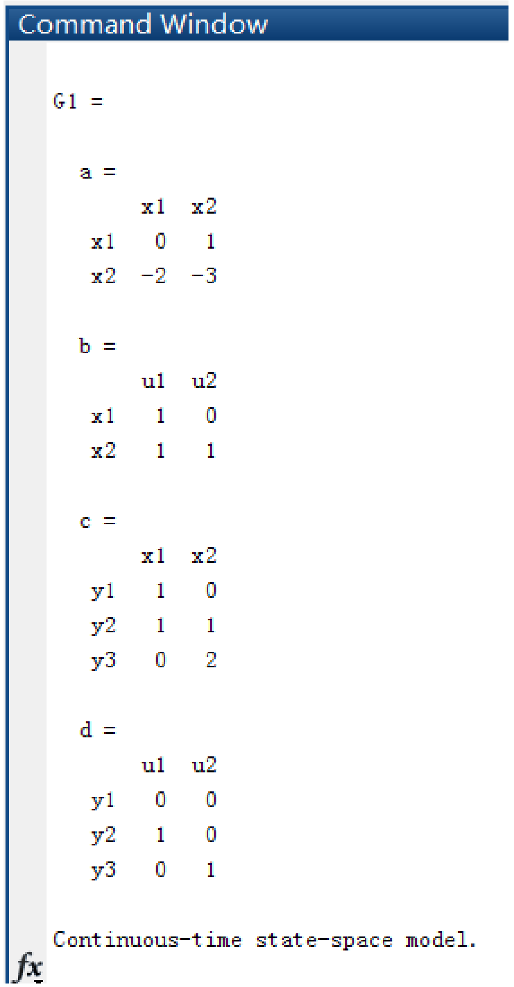
$$\begin{cases}\dot{x}=\begin{bmatrix}0 & 1\\ -2 & -3\end{bmatrix}x+\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}u\\ y=\begin{bmatrix}1 & 0\\ 1 & 1\\ 0 & 2\end{bmatrix}x+\begin{bmatrix}0 & 0\\ 1 & 0\\ 0 & 1\end{bmatrix}u\end{cases}$$



Editor - C:\Users\hp\Desktop\ex1_3.m

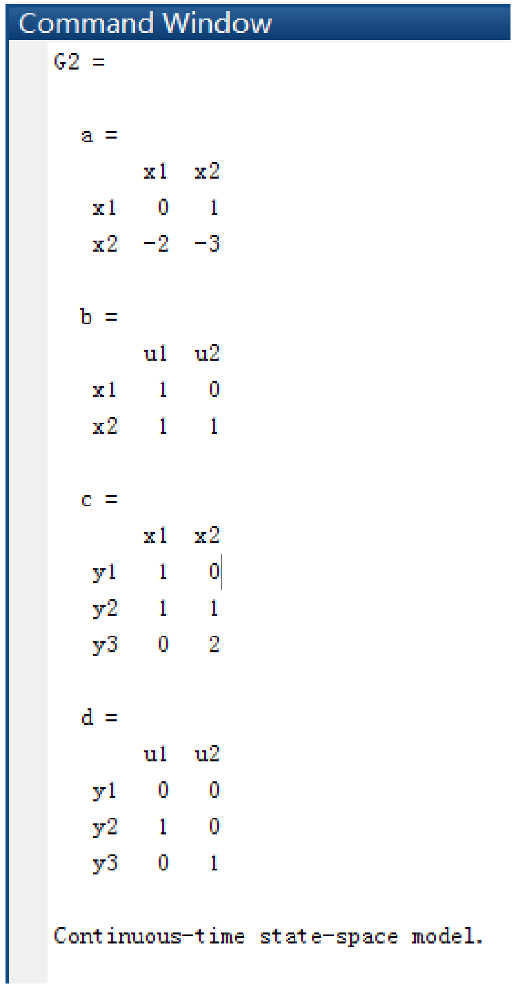
ex1_1.m ex1_2.m ex1_3.m

```
%%%%%%%%%%%%%%%%%
%%% Chuan-Ke Zhang
%%% 2021-10-06
%%% Example 1-3
%%% state-space equation
%%%%%%%%%%%%%%%%%

clc
clear

A = [0 1; -2 -3];
B = [1 0; 1 1];
C = [1 0; 1 1; 0 2];
D = [0 0; 1 0; 0 1];

G1 = ss(A,B,C,D)
G2 = ss([0 1; -2 -3],[1 0; 1 1],[1 0; 1 1; 0 2],[0 0; 1 0; 0 1])
```



Command Window

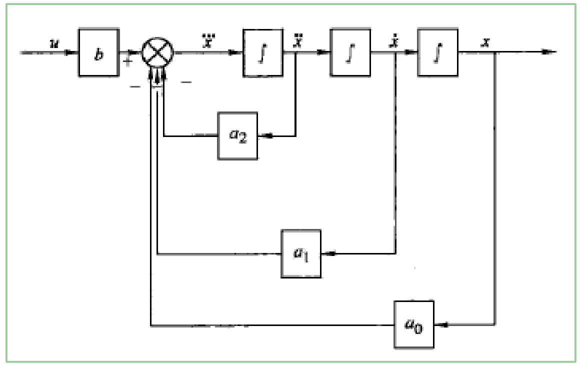
```
G1 =

  a = 
       x1  x2
   x1   0   1
   x2  -2  -3

  b = 
       u1  u2
   x1   1   0
   x2   1   1

  c = 
       x1  x2
   y1   1   0
   y2   1   1
   y3   0   2

  d = 
       u1  u2
   y1   0   0
   y2   1   0
   y3   0   1

Continuous-time state-space model.
```

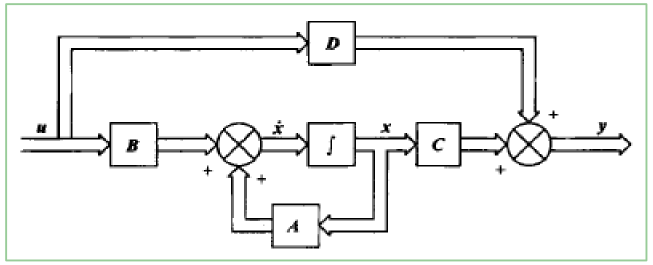


Command Window

```
G2 =

  a = 
       x1  x2
   x1   0   1
   x2  -2  -3

  b = 
       u1  u2
   x1   1   0
   x2   1   1

  c = 
       x1  x2
   y1   1   0
   y2   1   1
   y3   0   2

  d = 
       u1  u2
   y1   0   0
   y2   1   0
   y3   0   1

Continuous-time state-space model.
```

线性系统数学模型表示



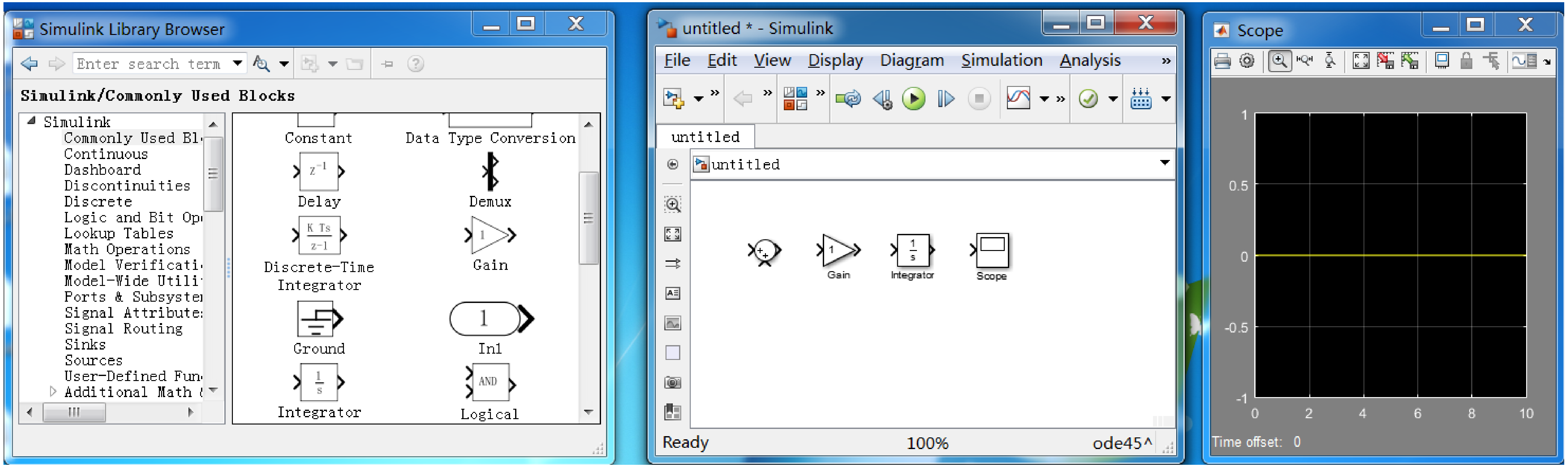


Simulink 模块

功能：

✓ 搭建系统结构框图模型

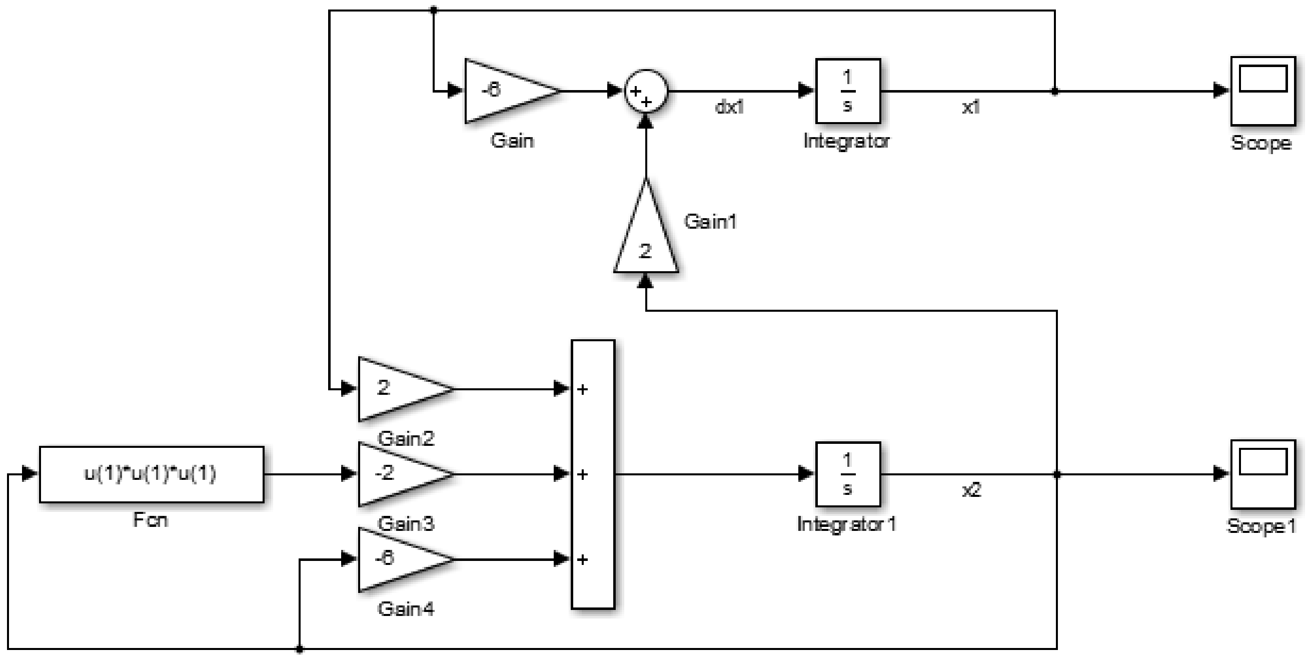
✓ 运行系统获取响应曲线



线性系统数学模型表示

例1-4：试用Simulink描述如下系统模型

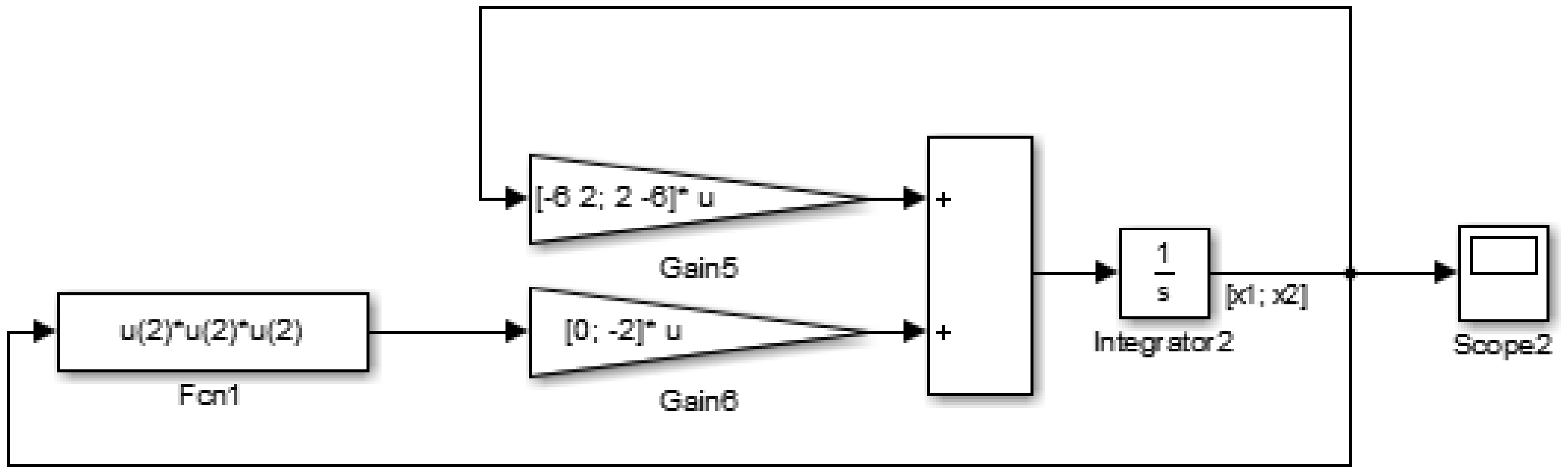
$$\begin{cases} \dot{x}_1 = -6x_1 + 2x_2 \\ \dot{x}_2 = 2x_1 - 6x_2 - 2x_2^3 \end{cases}$$



线性系统数学模型表示

例1-4：试用Simulink描述如下系统模型

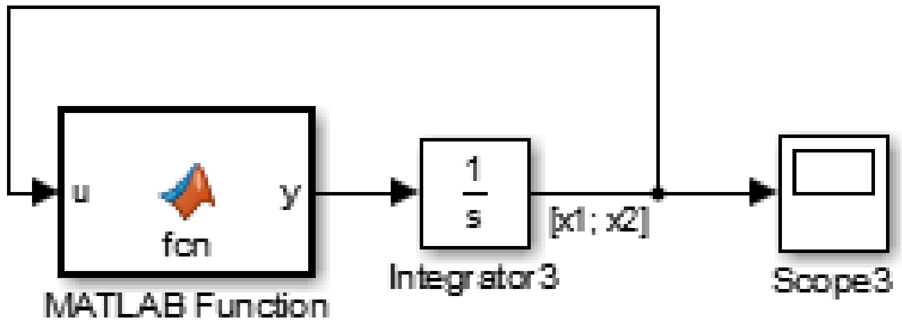
$$\begin{cases} \dot{x}_1 = -6x_1 + 2x_2 \\ \dot{x}_2 = 2x_1 - 6x_2 - 2x_2^3 \end{cases} \Rightarrow \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -6 & 2 \\ 2 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -2 \end{bmatrix} x_2^3$$

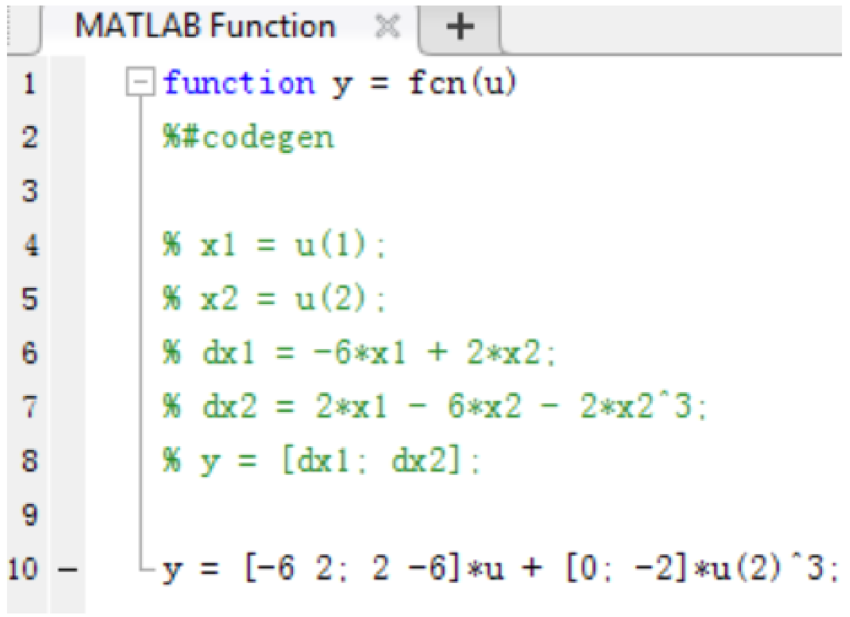


线性系统数学模型表示

例1-4：试用Simulink描述如下系统模型

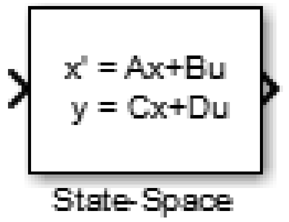
$$\begin{cases} \dot{x}_1 = -6x_1 + 2x_2 \\ \dot{x}_2 = 2x_1 - 6x_2 - 2x_2^3 \end{cases} \Rightarrow \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -6 & 2 \\ 2 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -2 \end{bmatrix} x_2^3$$





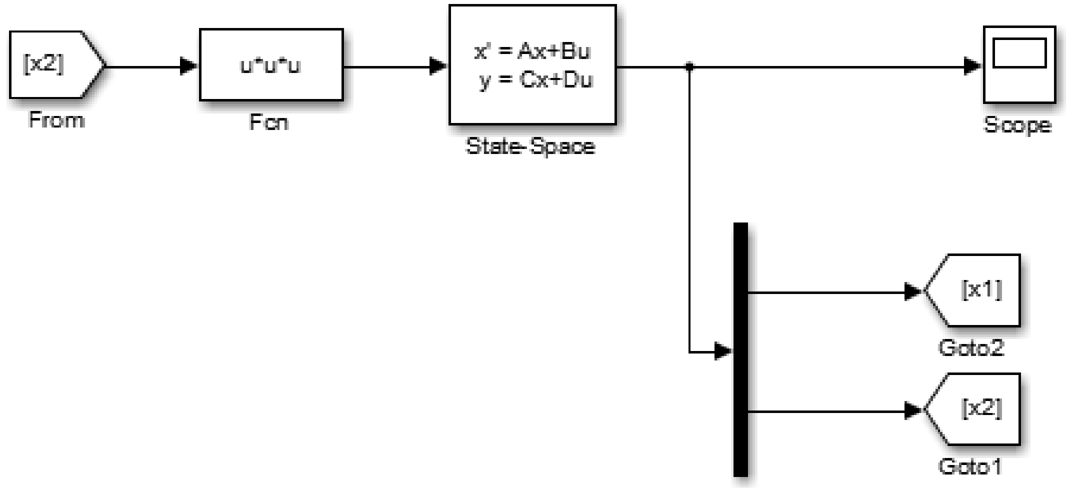
```
function y = fcn(u)
%#codegen

% x1 = u(1);
% x2 = u(2);
% dx1 = -6*x1 + 2*x2;
% dx2 = 2*x1 - 6*x2 - 2*x2^3;
% y = [dx1; dx2];

y = [-6 2; 2 -6]*u + [0; -2]*u(2)^3;
```

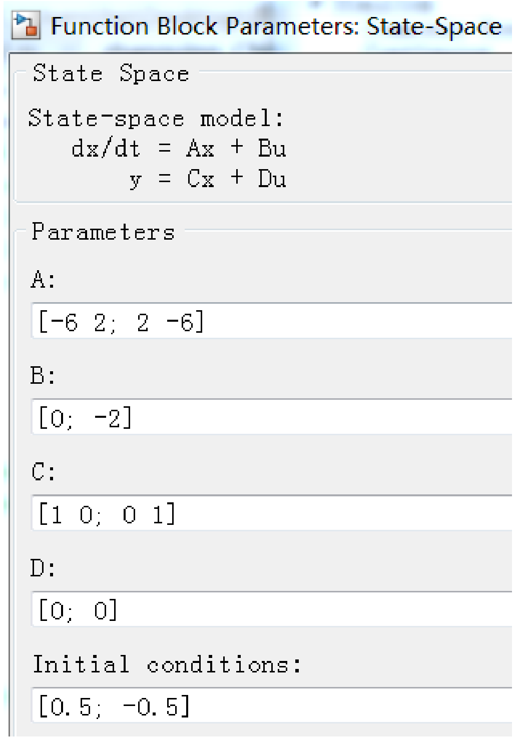
线性系统数学模型表示



例1-4：试用Simulink描述如下系统模型

$$\begin{cases}\dot{x}_1 = -6x_1 + 2x_2 \\ \dot{x}_2 = 2x_1 - 6x_2 - 2x_2^3\end{cases} \Rightarrow \begin{bmatrix}\dot{x}_1 \\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}-6 & 2 \\ 2 & -6\end{bmatrix}\begin{bmatrix}x_1 \\ x_2\end{bmatrix} + \begin{bmatrix}0 \\ -2\end{bmatrix} x_2^3$$





Function Block Parameters: State-Space

State Space

```
State-space model:
   dx/dt = Ax + Bu
       y = Cx + Du
```

Parameters

A:

[-6 2; 2 -6]

B:

[0; -2]

C:

[1 0; 0 1]

D:

[0; 0]

Initial conditions:

[0.5; -0.5]

课内实验：模型部分

状态空间模型与其它模型的相互转化

传递函数模型

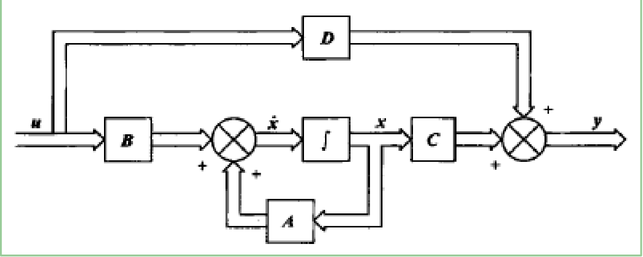
状态空间模型

结构图模型

$$W(s)=\frac{b_m s^m+b_{m-1}s^{m-1}+\cdots+b_1 s+b_0}{s^n+a_{n-1}s^{n-1}+\cdots+a_1 s+a_0}$$

$$W(s)=k\frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)}$$

$$\begin{cases}\dot{\boldsymbol{x}}(t)=\boldsymbol{A}\boldsymbol{x}(t)+\boldsymbol{B}\boldsymbol{u}(t)\\ \boldsymbol{y}(t)=\boldsymbol{C}\boldsymbol{x}(t)+\boldsymbol{D}\boldsymbol{u}(t)\end{cases}$$



状态空间模型与其它模型的相互转化

$$W(s)=\frac{b_m s^m+b_{m-1}s^{m-1}+\cdots+b_1 s+b_0}{s^n+a_{n-1}s^{n-1}+\cdots+a_1 s+a_0}$$

$$W(s)=k\frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)}$$

$$\Rightarrow \begin{cases}\dot{x}(t)=\boldsymbol{A}x(t)+\boldsymbol{B}u(t)\\ y(t)=\boldsymbol{C}x(t)+\boldsymbol{D}u(t)\end{cases}$$

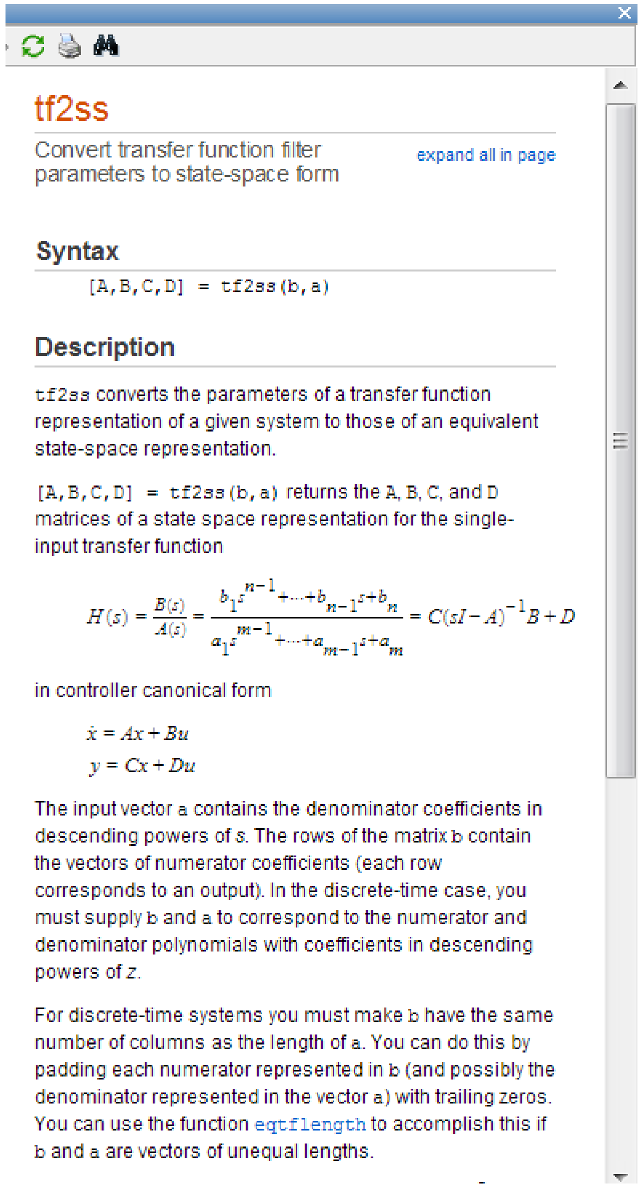
tf2ss()函数、zp2ss() 函数

功能：传函模型转化为状态空间模型

格式：

[A,B,C,D] = tf2ss(num,den)

[A,B,C,D] = zp2ss(num,den)



tf2ss

Convert transfer function filter parameters to state-space form

expand all in page

Syntax

```
[A,B,C,D] = tf2ss(b,a)
```

Description

`tf2ss` converts the parameters of a transfer function representation of a given system to those of an equivalent state-space representation.

`[A,B,C,D] = tf2ss(b,a)` returns the A, B, C, and D matrices of a state space representation for the single-input transfer function

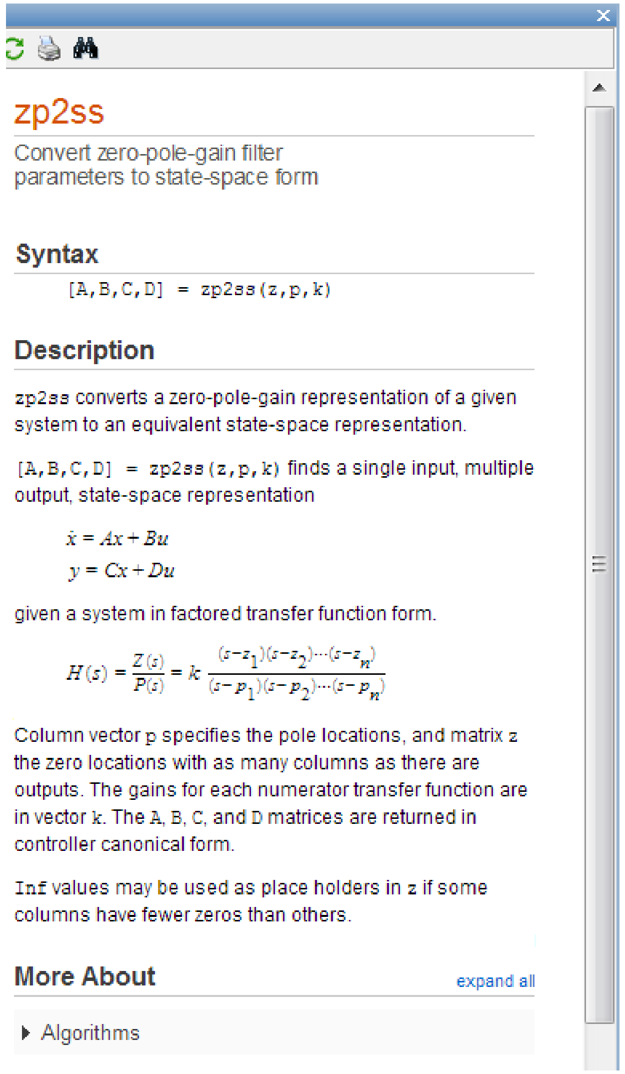
$$H(s)=\frac{B(s)}{A(s)}=\frac{b_1 s^{n-1}+\cdots+b_{n-1}s+b_n}{a_1 s^{m-1}+\cdots+a_{m-1}s+a_m}=C(sI-A)^{-1}B+D$$

in controller canonical form

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

The input vector `a` contains the denominator coefficients in descending powers of *s*. The rows of the matrix `b` contain the vectors of numerator coefficients (each row corresponds to an output). In the discrete-time case, you must supply `b` and `a` to correspond to the numerator and denominator polynomials with coefficients in descending powers of *z*.

For discrete-time systems you must make `b` have the same number of columns as the length of `a`. You can do this by padding each numerator represented in `b` (and possibly the denominator represented in the vector `a`) with trailing zeros. You can use the function `eqtflength` to accomplish this if `b` and `a` are vectors of unequal lengths.



zp2ss

Convert zero-pole-gain filter parameters to state-space form

Syntax

```
[A,B,C,D] = zp2ss(z,p,k)
```

Description

`zp2ss` converts a zero-pole-gain representation of a given system to an equivalent state-space representation.

`[A,B,C,D] = zp2ss(z,p,k)` finds a single input, multiple output, state-space representation

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

given a system in factored transfer function form.

$$H(s)=\frac{Z(s)}{P(s)}=k\frac{(s-z_1)(s-z_2)\cdots(s-z_n)}{(s-p_1)(s-p_2)\cdots(s-p_n)}$$

Column vector `p` specifies the pole locations, and matrix `z` the zero locations with as many columns as there are outputs. The gains for each numerator transfer function are in vector `k`. The `A`, `B`, `C`, and `D` matrices are returned in controller canonical form.

`Inf` values may be used as place holders in `z` if some columns have fewer zeros than others.

More About

expand all

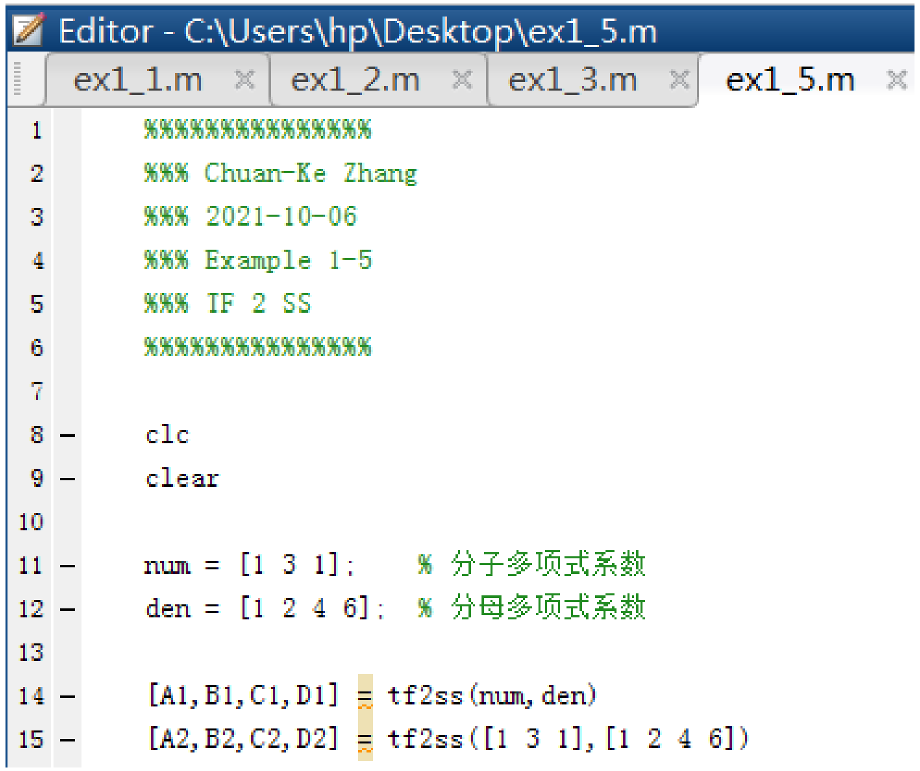
▸ Algorithms

状态空间模型与其它模型的相互转化

例1-5：将以下传递函数模型变为状态空间模型

$$W(s)=\frac{s^2+3s+1}{s^3+2s^2+4s+6}$$

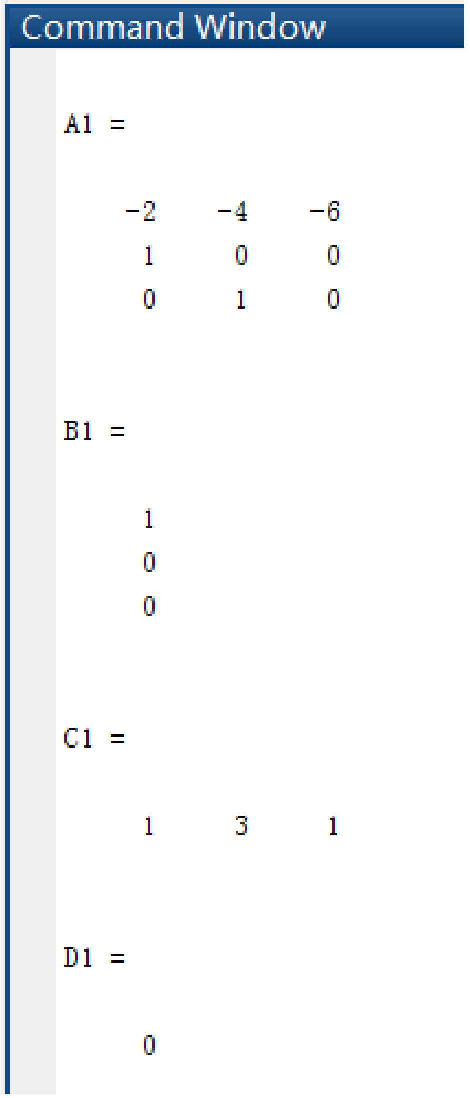
$$\begin{cases}\dot{x}=\begin{bmatrix}-2 & -4 & -6\\ 1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}x+\begin{bmatrix}1\\0\\0\end{bmatrix}u\\ y=\begin{bmatrix}1 & 3 & 1\end{bmatrix}x\end{cases}$$



Editor - C:\Users\hp\Desktop\ex1_5.m

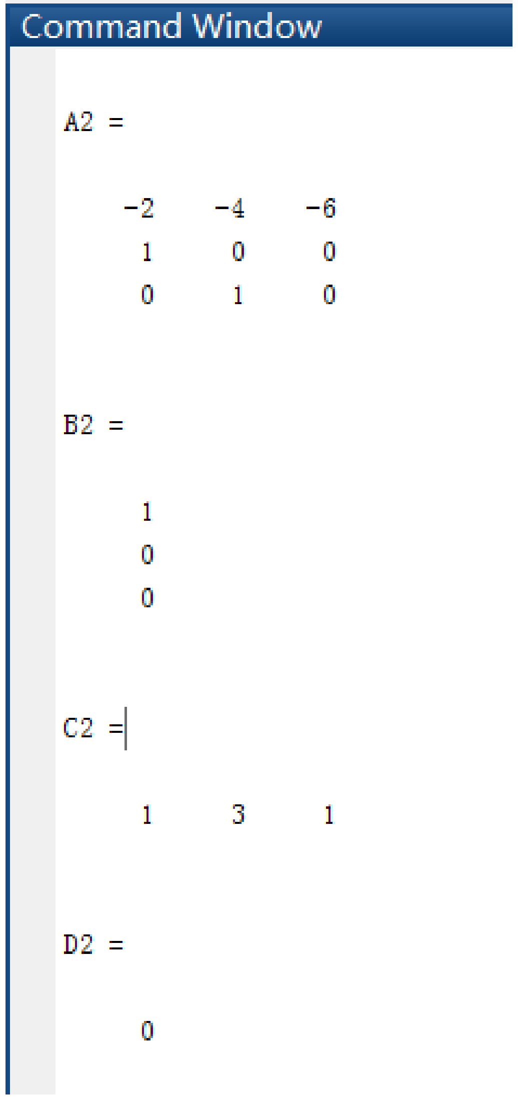
ex1_1.m ex1_2.m ex1_3.m ex1_5.m

```
%%%%%%%%%%%%%%%%%
%%% Chuan-Ke Zhang
%%% 2021-10-06
%%% Example 1-5
%%% TF 2 SS
%%%%%%%%%%%%%%%%%

clc
clear

num = [1 3 1];    % 分子多项式系数
den = [1 2 4 6];  % 分母多项式系数

[A1,B1,C1,D1] = tf2ss(num,den)
[A2,B2,C2,D2] = tf2ss([1 3 1],[1 2 4 6])
```



Command Window

```
A1 =

    -2    -4    -6
     1     0     0
     0     1     0


B1 =

     1
     0
     0


C1 =

     1     3     1


D1 =

     0
```



Command Window

```
A2 =

    -2    -4    -6
     1     0     0
     0     1     0


B2 =

     1
     0
     0


C2 =

     1     3     1


D2 =

     0
```

状态空间模型与其它模型的相互转化

$$\begin{cases}\dot{\boldsymbol{x}}(t)=\boldsymbol{Ax}(t)+\boldsymbol{Bu}(t)\\ \boldsymbol{y}(t)=\boldsymbol{Cx}(t)+\boldsymbol{Du}(t)\end{cases} \Rightarrow$$

$$W(s)=\frac{b_m s^m+b_{m-1}s^{m-1}+\cdots+b_1 s+b_0}{s^n+a_{n-1}s^{n-1}+\cdots+a_1 s+a_0}$$

$$W(s)=k\frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)}$$

ss2tf()函数、ss2zp() 函数

功能：状态空间模型转化为传函模型

格式：

SISO:

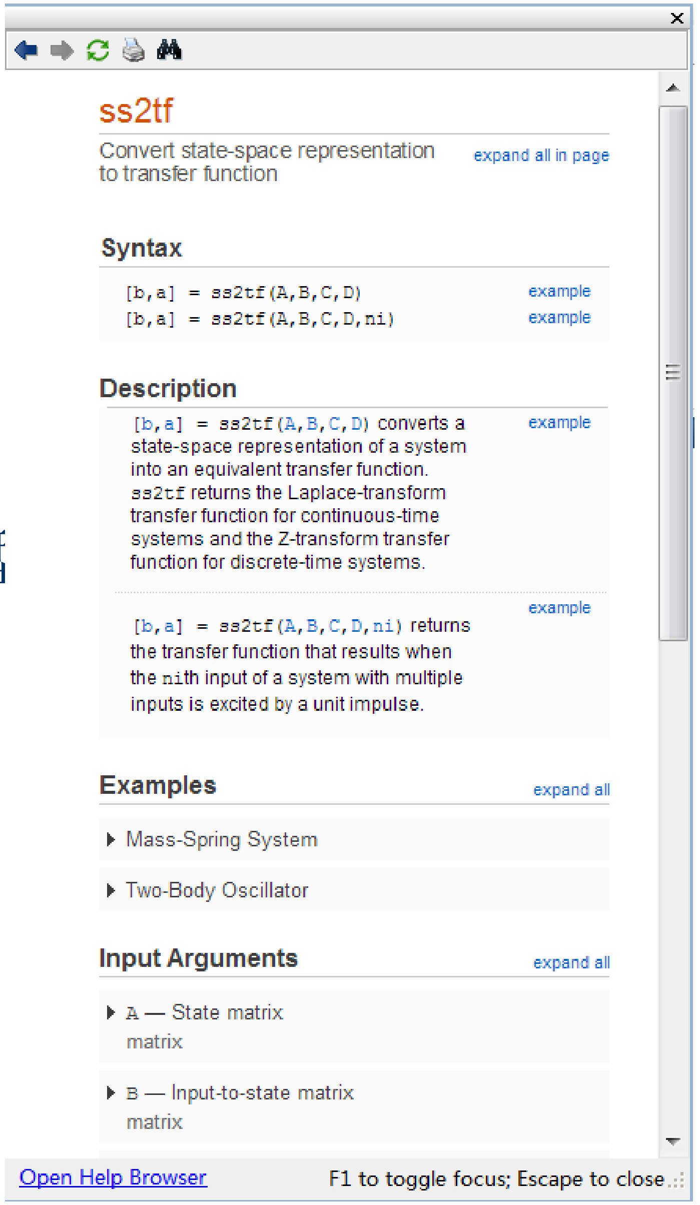
[num,den] = ss2tf(A,B,C,D)

[z,p,k] = ss2zp(A,B,C,D)

MIMO:

[num,den] = ss2tf(A,B,C,D,iu)

[z,p,k] = ss2zp(A,B,C,D,iu)



ss2tf

Convert state-space representation to transfer function — expand all in page

Syntax

```
[b,a] = ss2tf(A,B,C,D)        example
[b,a] = ss2tf(A,B,C,D,ni)     example
```

Description

`[b,a] = ss2tf(A,B,C,D)` converts a state-space representation of a system into an equivalent transfer function. `ss2tf` returns the Laplace-transform transfer function for continuous-time systems and the Z-transform transfer function for discrete-time systems. — example

`[b,a] = ss2tf(A,B,C,D,ni)` returns the transfer function that results when the `ni`th input of a system with multiple inputs is excited by a unit impulse. — example

Examples — expand all

▸ Mass-Spring System

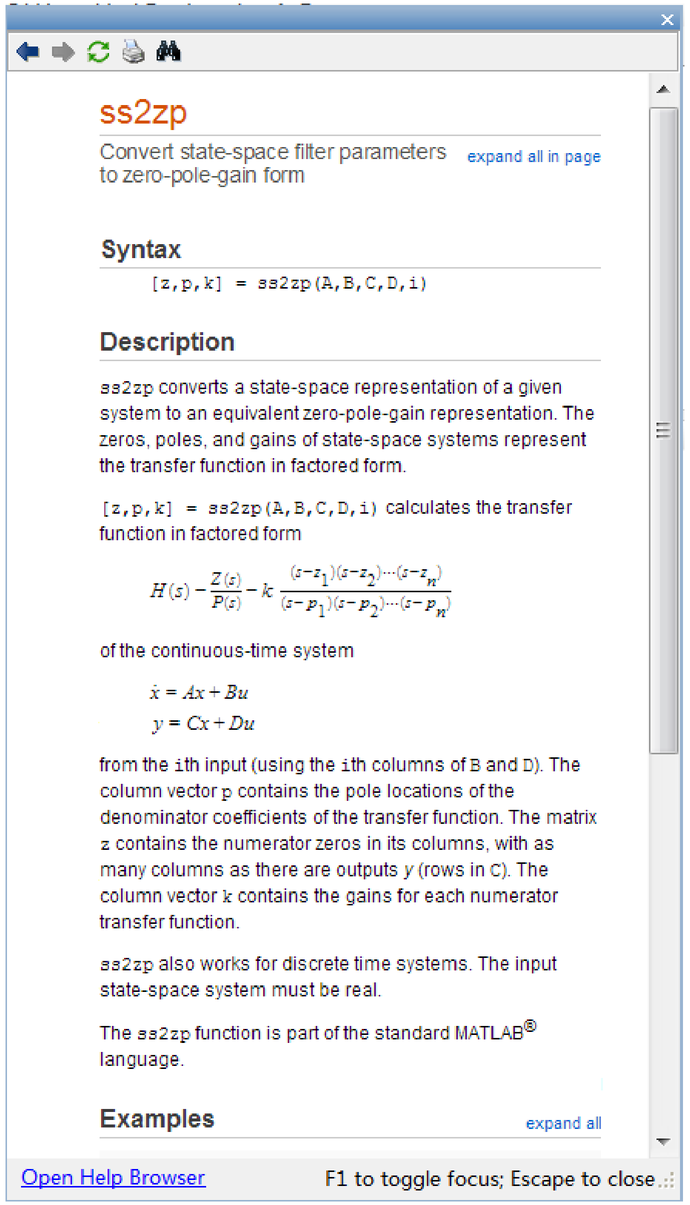
▸ Two-Body Oscillator

Input Arguments — expand all

▸ `A` — State matrix
matrix

▸ `B` — Input-to-state matrix
matrix

Open Help Browser — F1 to toggle focus; Escape to close



ss2zp

Convert state-space filter parameters to zero-pole-gain form — expand all in page

Syntax

```
[z,p,k] = ss2zp(A,B,C,D,i)
```

Description

`ss2zp` converts a state-space representation of a given system to an equivalent zero-pole-gain representation. The zeros, poles, and gains of state-space systems represent the transfer function in factored form.

`[z,p,k] = ss2zp(A,B,C,D,i)` calculates the transfer function in factored form

$$H(s)-\frac{Z(s)}{P(s)}-k\frac{(s-z_1)(s-z_2)\cdots(s-z_n)}{(s-p_1)(s-p_2)\cdots(s-p_n)}$$

of the continuous-time system

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

from the `i`th input (using the `i`th columns of `B` and `D`). The column vector `p` contains the pole locations of the denominator coefficients of the transfer function. The matrix `z` contains the numerator zeros in its columns, with as many columns as there are outputs *y* (rows in `C`). The column vector `k` contains the gains for each numerator transfer function.

`ss2zp` also works for discrete time systems. The input state-space system must be real.

The `ss2zp` function is part of the standard MATLAB® language.

Examples — expand all

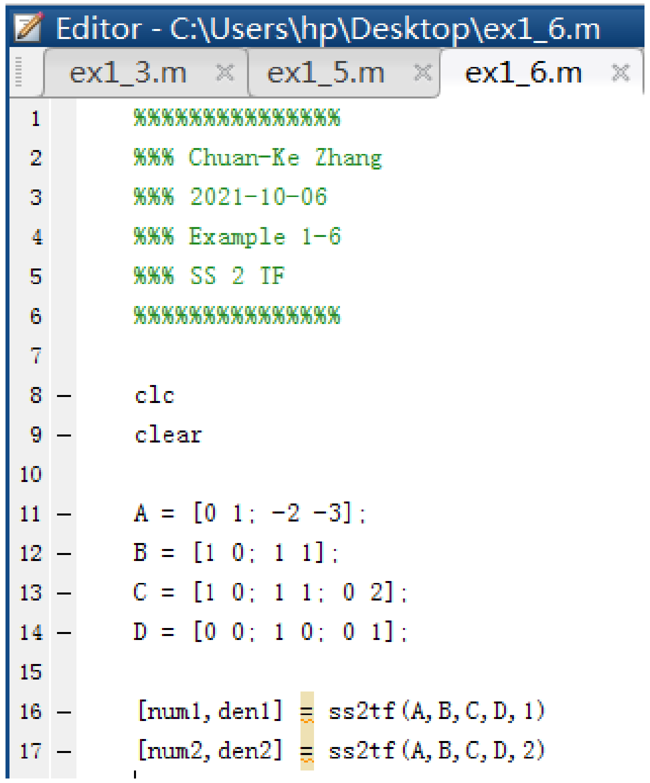
Open Help Browser — F1 to toggle focus; Escape to close

状态空间模型与其它模型的相互转化

例1-6：求以下状态空间模型对应的传递函数阵

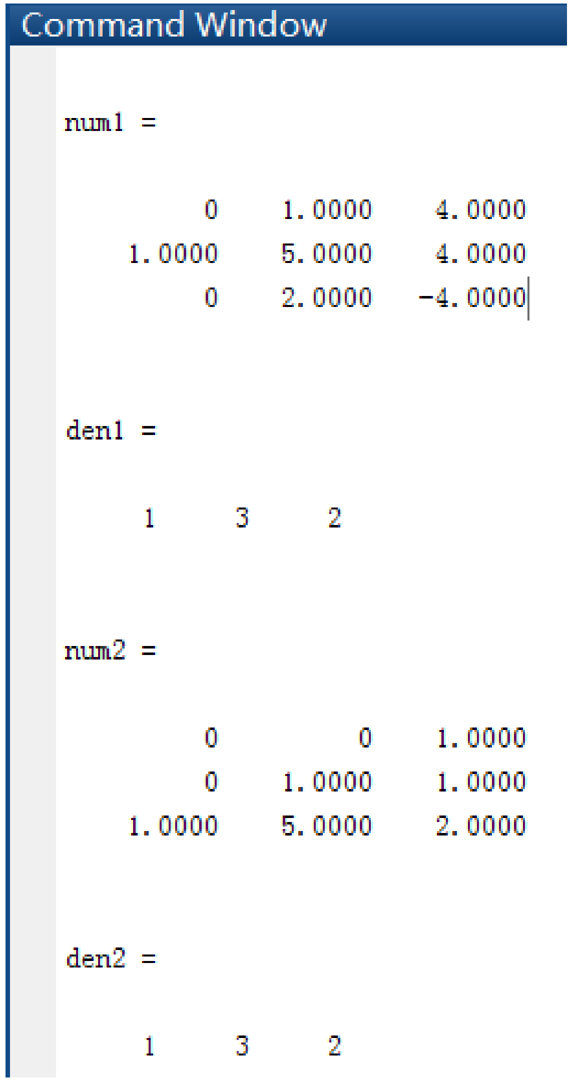
$$\begin{cases}\dot{x}=\begin{bmatrix}0 & 1\\ -2 & -3\end{bmatrix}x+\begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}u\\ y=\begin{bmatrix}1 & 0\\ 1 & 1\\ 0 & 2\end{bmatrix}x+\begin{bmatrix}0 & 0\\ 1 & 0\\ 0 & 1\end{bmatrix}u\end{cases}$$

$$W(s)=\frac{\begin{bmatrix}s+4 & 1\\ s^2+5s+4 & s+1\\ 2s-4 & s^2+5s+2\end{bmatrix}}{s^2+3s+2}$$


Editor - C:\Users\hp\Desktop\ex1_6.m

ex1_3.m ex1_5.m ex1_6.m

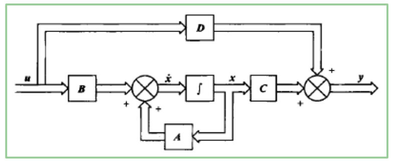
```
%%%%%%%%%%%%%%%%%%
%%% Chuan-Ke Zhang
%%% 2021-10-06
%%% Example 1-6
%%% SS 2 TF
%%%%%%%%%%%%%%%%%%

clc
clear

A = [0 1; -2 -3];
B = [1 0; 1 1];
C = [1 0; 1 1; 0 2];
D = [0 0; 1 0; 0 1];

[num1,den1] = ss2tf(A,B,C,D,1)
[num2,den2] = ss2tf(A,B,C,D,2)
```


Command Window

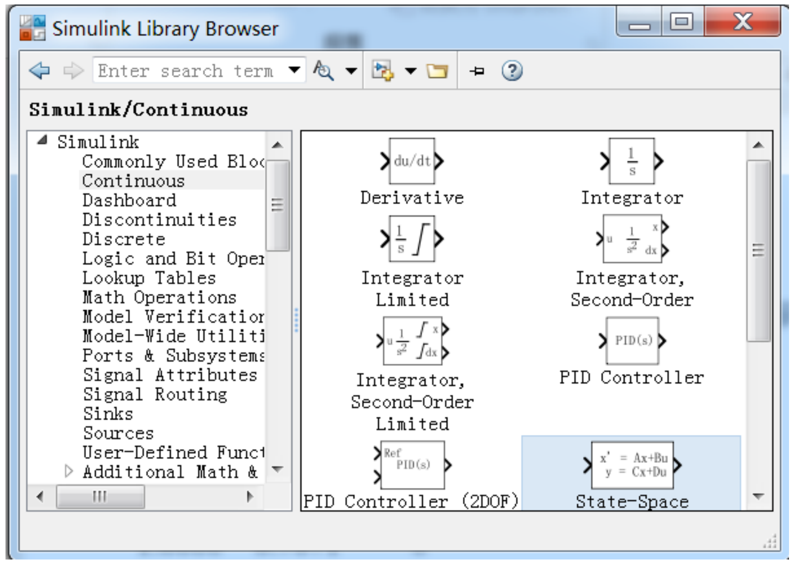
```
num1 =

         0    1.0000    4.0000
    1.0000    5.0000    4.0000
         0    2.0000   -4.0000


den1 =

     1     3     2


num2 =

         0         0    1.0000
         0    1.0000    1.0000
    1.0000    5.0000    2.0000


den2 =

     1     3     2
```

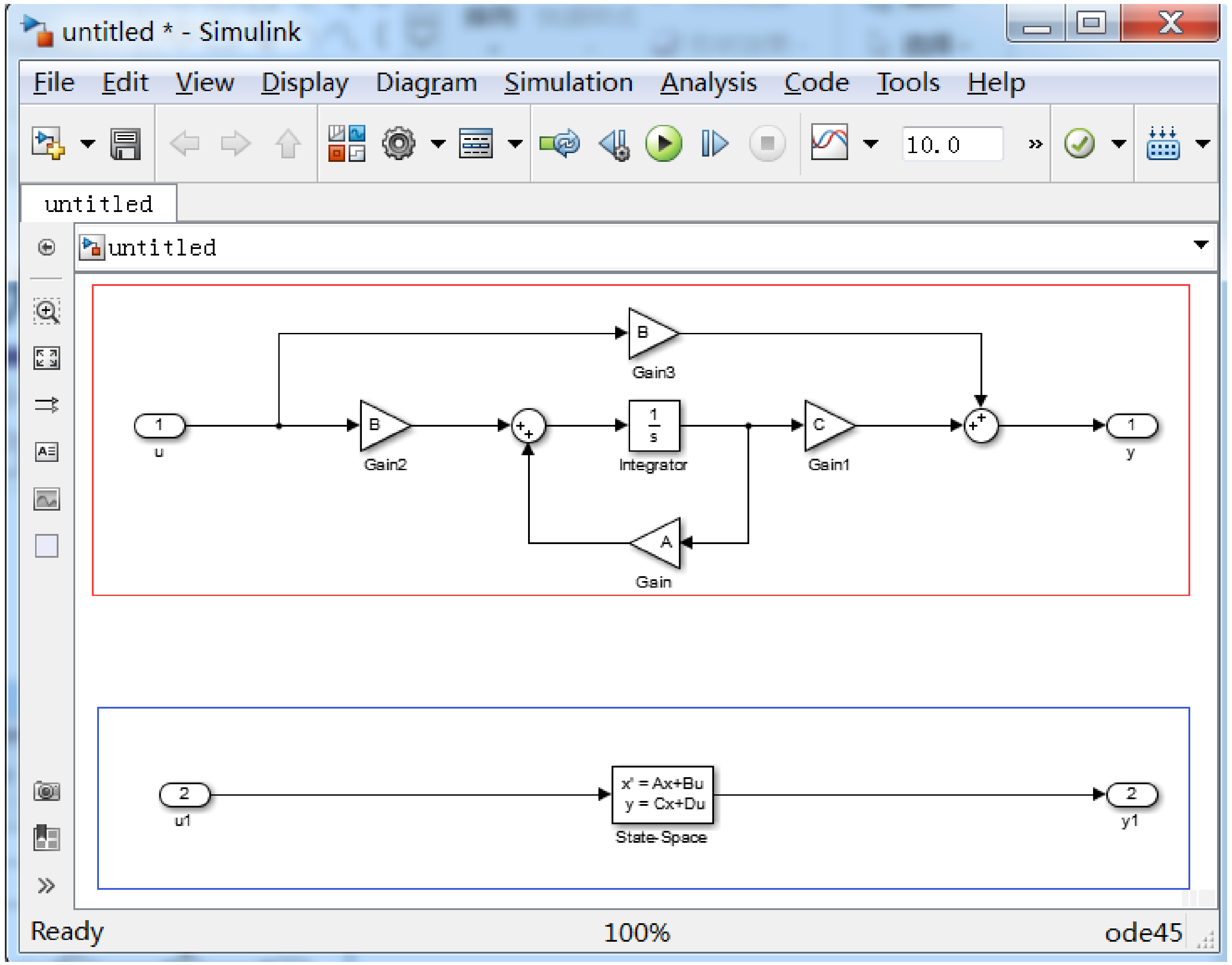
状态空间模型与其它模型的相互转化

$$\begin{cases}\dot{x}(t)=Ax(t)+Bu(t)\\ y(t)=Cx(t)+Du(t)\end{cases}$$



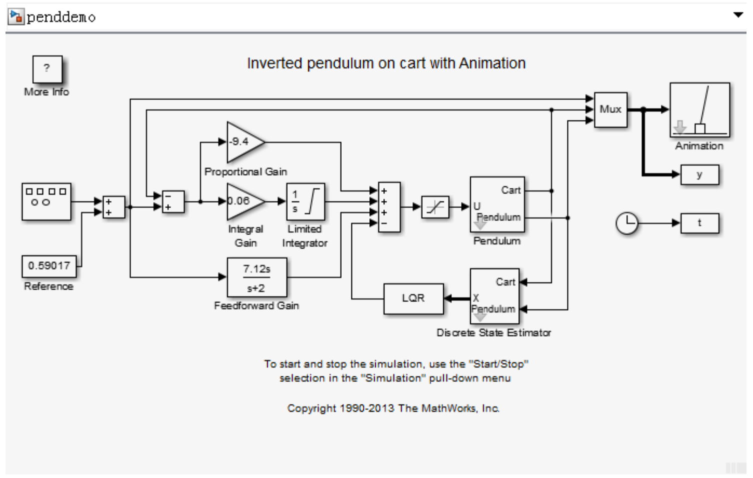
Simulink（重点）





例1-4：Pages 15-18

状态空间模型与其它模型的相互转化


penddemo
Inverted pendulum on cart with Animation
More Info
Mux
Animation
Proportional Gain
Integral Gain
Limited Integrator
Cart Pendulum
y
t
0.59017
Reference
Feedforward Gain
LQR
Discrete State Estimator
To start and stop the simulation, use the "Start/Stop" selection in the "Simulation" pull-down menu
Copyright 1990-2013 The MathWorks, Inc.

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

simulink 线性 工具 百度一下

网页 资讯 视频 图片 知道 文库 贴吧 地图 采购 更多

百度为您找到相关结果约11,400,000个 搜索工具

利用MATLAB 自带的工具提取系统线性化总模型 - 百度经验

2014年8月5日 利用MATLAB 自带的工具提取系统线性化总模型,利用MATLABimulik自带的liearaal yi工具提取系统线性化总模型,将复杂系统线性化,更容易分析其时域和频域的关系了

百度经验 百度快照

Simulink下的线性系统分析工具在哪找到? – MATLAB中文论坛

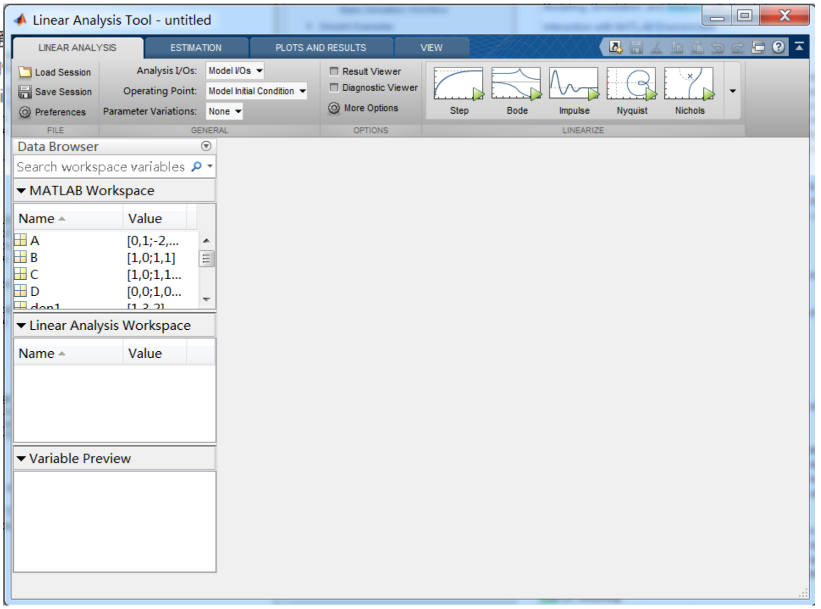
2012年2月12日 书上说"选中Tools|Linear System 菜单项，则打开线性系统分析工具，用户选择其中的Get Linear Model 菜单...

www.ilovematlab.cn/thread-1619... 百度快照

simulink之线性系统分析 - 知乎

2018年9月13日 在利用matlab 进行控制系统设计时,适 uist、rlocus、nichols、step等指令来分析线性系统的 中,可以进行相同分析,且更加直观,性能更强。 Analysi

知乎 百度快照



Simulink：Linear Analysis Tool（自学内容）

功能：从结构图获取线性状态空间模型

课内实验：模型部分

状态空间模型与状态空间模型的转化

$$\begin{tikzcd}
 & \boxed{\begin{cases} \dot{z}(t) = T^{-1}\boldsymbol{A}Tz(t) + T^{-1}\boldsymbol{Bu}(t) \\ \boldsymbol{y}(t) = \boldsymbol{C}Tz(t) + \boldsymbol{Du}(t) \end{cases}} \arrow[d, "\boxed{\bar{T} = T^{-1}}"] \\
\boxed{\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{Ax}(t) + \boldsymbol{Bu}(t) \\ \boldsymbol{y}(t) = \boldsymbol{Cx}(t) + \boldsymbol{Du}(t) \end{cases}} \arrow[ur, "\text{教材：}\boxed{x = Tz}"] \arrow[dr, "\text{MATLAB：}\boxed{\bar{z} = \bar{T}x}"'] & \\
 & \boxed{\begin{cases} \dot{\bar{z}}(t) = \bar{T}\boldsymbol{A}\bar{T}^{-1}\bar{z}(t) + \bar{T}\boldsymbol{Bu}(t) \\ \boldsymbol{y}(t) = \boldsymbol{C}\bar{T}^{-1}\bar{z}(t) + \boldsymbol{Du}(t) \end{cases}}
\end{tikzcd}$$

实现状态空间模型的规范化，以便于系统分析与设计

- ✓ 对角线标准型、约旦标准型
- ✓ 能控标准型、能观标准型（下次实验）
- ✓ 能控结构分解、能观结构分解（下次实验）

状态空间模型与状态空间模型的转化

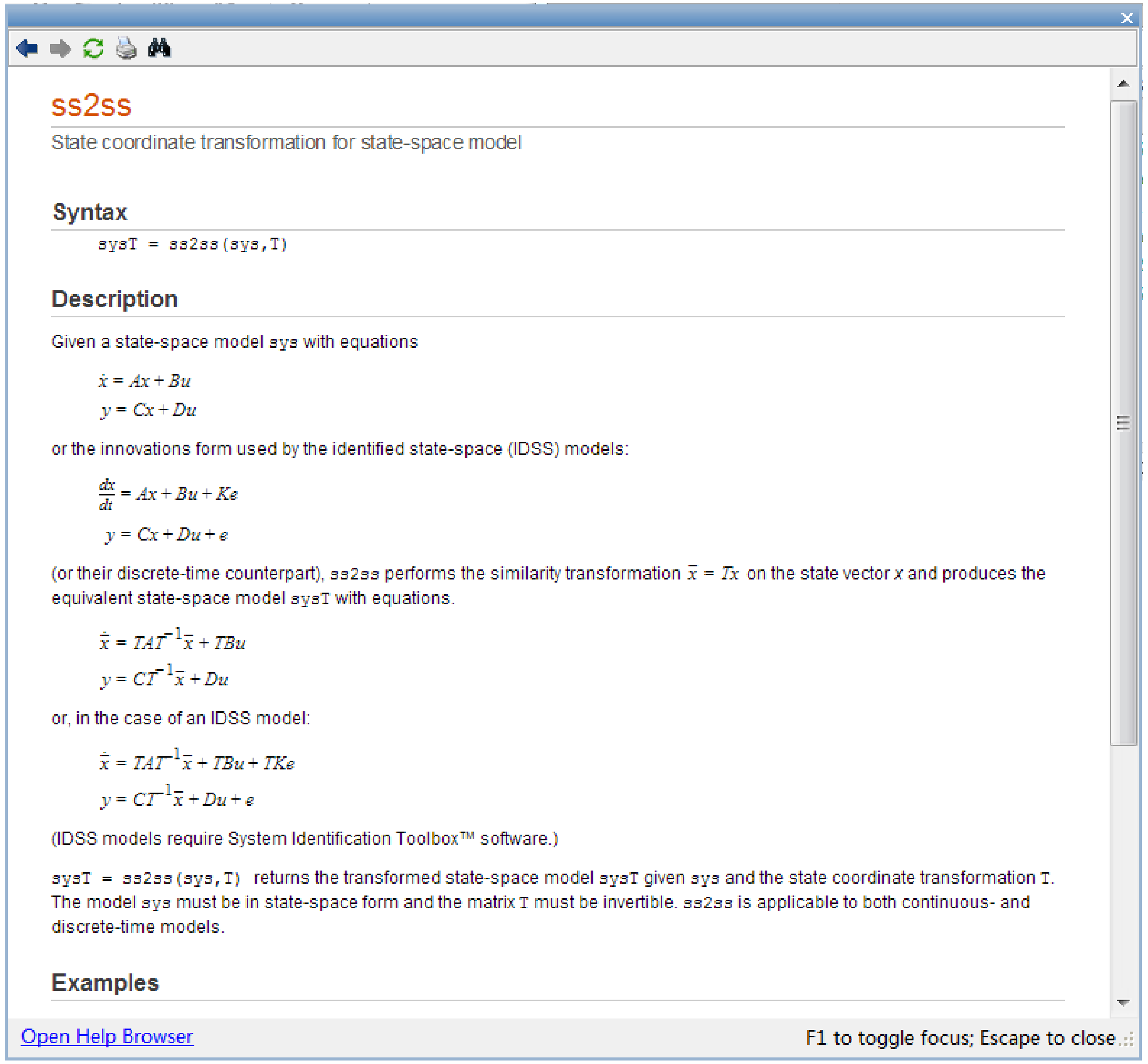
$$\begin{cases}\dot{x}(t) = \boldsymbol{A}x(t) + \boldsymbol{B}u(t) \\ y(t) = \boldsymbol{C}x(t) + \boldsymbol{D}u(t)\end{cases} \Rightarrow \begin{cases}\dot{\bar{z}}(t) = \bar{T}\boldsymbol{A}\bar{T}^{-1}\bar{z}(t) + \bar{T}\boldsymbol{B}u(t) \\ y(t) = \boldsymbol{C}\bar{T}^{-1}\bar{z}(t) + \boldsymbol{D}u(t)\end{cases}$$

ss2ss()函数

功能：实现状态空间模型的线性变换
格式：

sys_new = ss2ss(sys_original,T)
[An,Bn,Cn,Dn] = ss2ss(A,B,C,D,T)

其中，T为非奇异变换矩阵



ss2ss
State coordinate transformation for state-space model

Syntax

```
sysT = ss2ss(sys,T)
```

Description

Given a state-space model `sys` with equations

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

or the innovations form used by the identified state-space (IDSS) models:

$$\frac{dx}{dt} = Ax + Bu + Ke$$
$$y = Cx + Du + e$$

(or their discrete-time counterpart), `ss2ss` performs the similarity transformation $\bar{x} = Tx$ on the state vector x and produces the equivalent state-space model `sysT` with equations.

$$\dot{\bar{x}} = TAT^{-1}\bar{x} + TBu$$
$$y = CT^{-1}\bar{x} + Du$$

or, in the case of an IDSS model:

$$\dot{\bar{x}} = TAT^{-1}\bar{x} + TBu + TKe$$
$$y = CT^{-1}\bar{x} + Du + e$$

(IDSS models require System Identification Toolbox™ software.)

`sysT = ss2ss(sys,T)` returns the transformed state-space model `sysT` given `sys` and the state coordinate transformation `T`. The model `sys` must be in state-space form and the matrix `T` must be invertible. `ss2ss` is applicable to both continuous- and discrete-time models.

Examples

Open Help Browser F1 to toggle focus; Escape to close

状态空间模型与状态空间模型的转化

$$\begin{cases}\dot{x}(t)=\boldsymbol{A}x(t)+\boldsymbol{B}u(t)\\ y(t)=\boldsymbol{C}x(t)+\boldsymbol{D}u(t)\end{cases} \Rightarrow \begin{cases}\dot{\bar{z}}(t)=\bar{T}\boldsymbol{A}\bar{T}^{-1}\bar{z}(t)+\bar{T}\boldsymbol{B}u(t)\\ y(t)=\boldsymbol{C}\bar{T}^{-1}\bar{z}(t)+\boldsymbol{D}u(t)\end{cases}$$

对角线标准型

要求：系统矩阵均为单根

步骤：先求变换矩阵T，再求标准型

Lambda = eig(A)

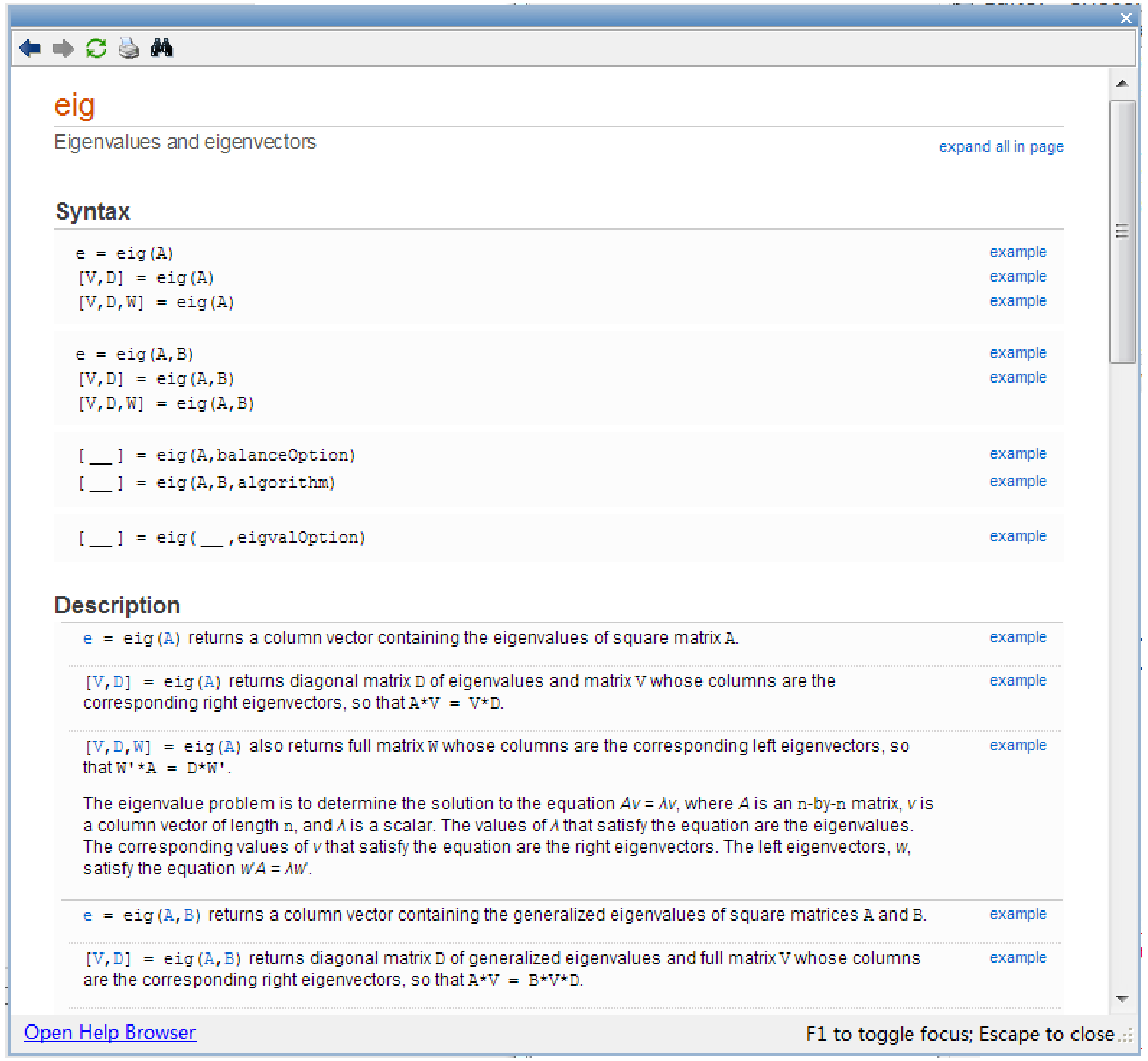
[V, L] = eig(A)

T = inv(V)

sys_new = ss2ss(sys_original,T)

[An,Bn,Cn,Dn] = ss2ss(A,B,C,D,T)

其中，Lambda为特征值，V为对应特征向量，即为所需变换阵T的逆



eig

Eigenvalues and eigenvectors

expand all in page

Syntax

```
e = eig(A)                              example
[V,D] = eig(A)                          example
[V,D,W] = eig(A)                        example

e = eig(A,B)                            example
[V,D] = eig(A,B)                        example
[V,D,W] = eig(A,B)

[ __ ] = eig(A,balanceOption)           example
[ __ ] = eig(A,B,algorithm)             example

[ __ ] = eig( __ ,eigvalOption)         example
```

Description

`e = eig(A)` returns a column vector containing the eigenvalues of square matrix `A`. example

`[V,D] = eig(A)` returns diagonal matrix `D` of eigenvalues and matrix `V` whose columns are the corresponding right eigenvectors, so that `A*V = V*D`. example

`[V,D,W] = eig(A)` also returns full matrix `W` whose columns are the corresponding left eigenvectors, so that `W'*A = D*W'`. example

The eigenvalue problem is to determine the solution to the equation $Av = \lambda v$, where A is an `n`-by-`n` matrix, v is a column vector of length `n`, and λ is a scalar. The values of λ that satisfy the equation are the eigenvalues. The corresponding values of v that satisfy the equation are the right eigenvectors. The left eigenvectors, w, satisfy the equation $w'A = \lambda w'$.

`e = eig(A,B)` returns a column vector containing the generalized eigenvalues of square matrices `A` and `B`. example

`[V,D] = eig(A,B)` returns diagonal matrix `D` of generalized eigenvalues and full matrix `V` whose columns are the corresponding right eigenvectors, so that `A*V = B*V*D`. example

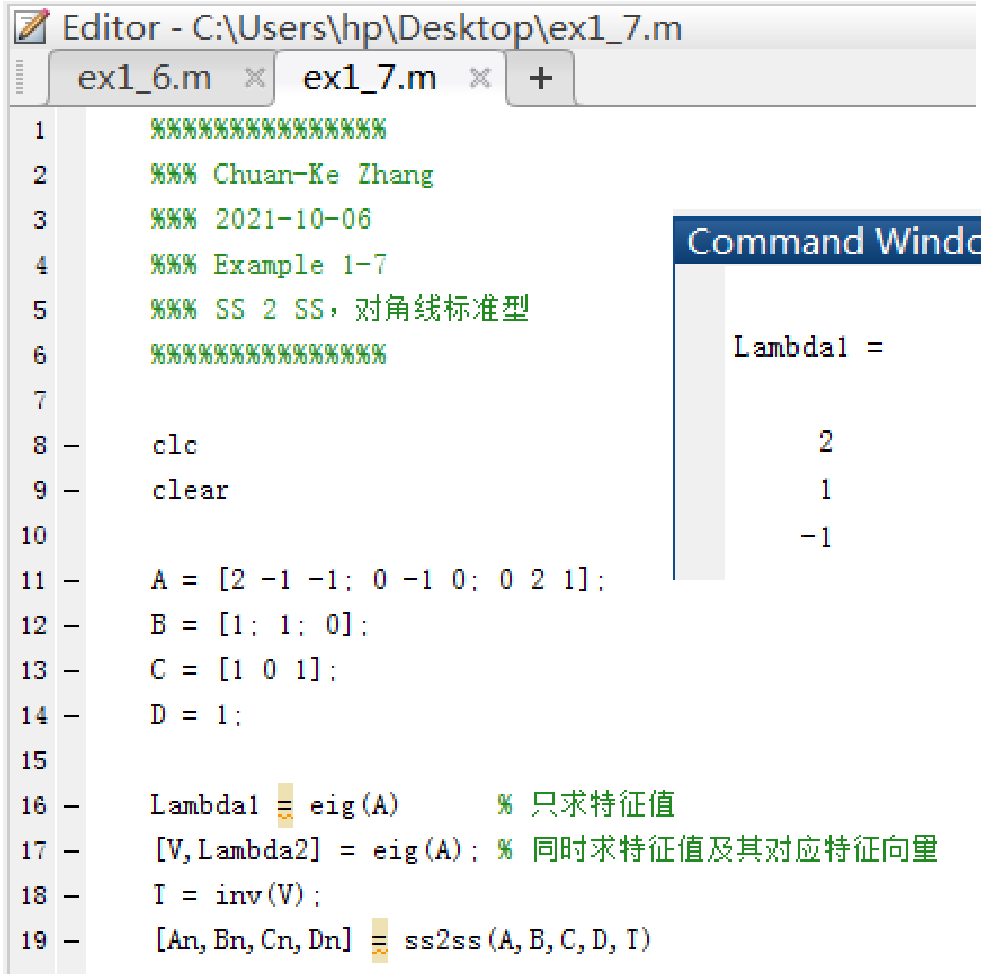
Open Help Browser

F1 to toggle focus; Escape to close

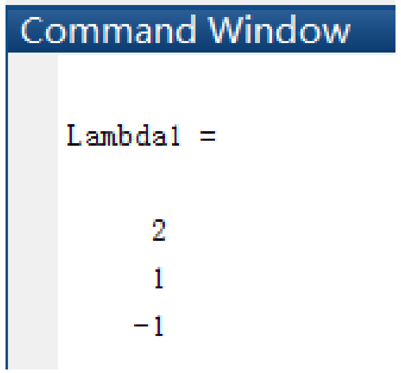
状态空间模型与状态空间模型的转化

例1-7：将以下状态空间模型化为对角线标准型

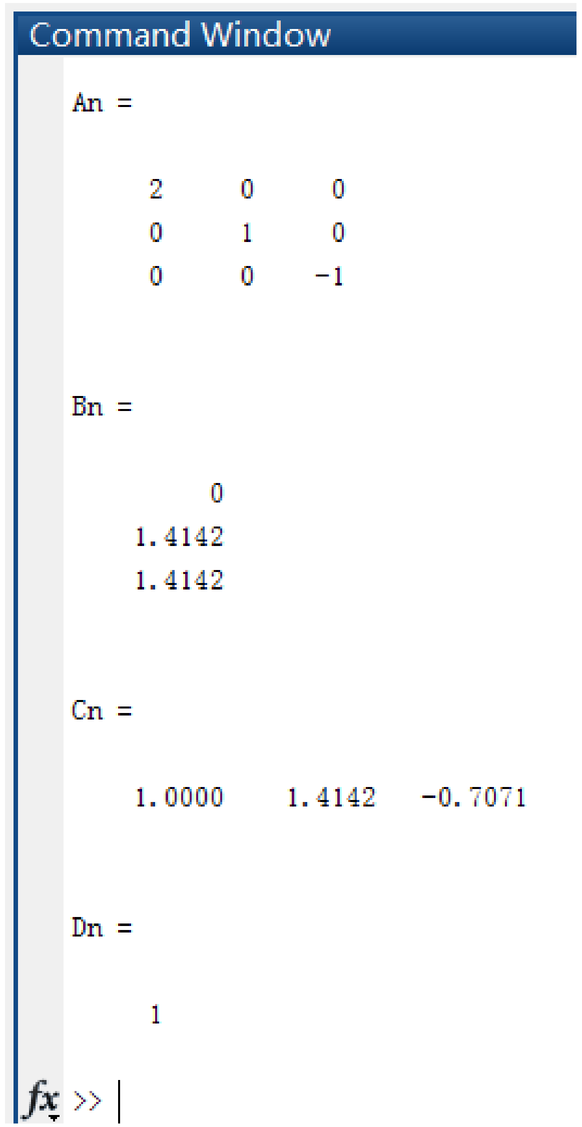
$$\begin{cases}\dot{x}=\begin{bmatrix}2 & -1 & -1\\ 0 & -1 & 0\\ 0 & 2 & 1\end{bmatrix}x+\begin{bmatrix}1\\ 1\\ 0\end{bmatrix}u\\ y=\begin{bmatrix}1 & 0 & 1\end{bmatrix}x+u\end{cases}$$



```
%%%%%%%%%%%%%%%%%%
%%% Chuan-Ke Zhang
%%% 2021-10-06
%%% Example 1-7
%%% SS 2 SS，对角线标准型
%%%%%%%%%%%%%%%%%%

clc
clear

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

Lambda1 = eig(A)       % 只求特征值
[V,Lambda2] = eig(A); % 同时求特征值及其对应特征向量
I = inv(V);
[An,Bn,Cn,Dn] = ss2ss(A,B,C,D,I)
```



```
Lambda1 =

     2
     1
    -1
```



```
An =

     2     0     0
     0     1     0
     0     0    -1


Bn =

         0
    1.4142
    1.4142


Cn =

    1.0000    1.4142   -0.7071


Dn =

     1

fx >>
```

$$\begin{cases}\dot{x}=\begin{bmatrix}2 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & -1\end{bmatrix}x+\begin{bmatrix}0\\ 1.4142\\ 1.4142\end{bmatrix}u\\ y=\begin{bmatrix}1 & 1.4142 & -0.7071\end{bmatrix}x+u\end{cases}$$

状态空间模型与状态空间模型的转化

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases} \Rightarrow \begin{cases} \dot{\bar{z}}(t) = \bar{T}A\bar{T}^{-1}\bar{z}(t) + \bar{T}Bu(t) \\ y(t) = C\bar{T}^{-1}\bar{z}(t) + Du(t) \end{cases}$$

约旦标准型

要求：系统矩阵存在重根

步骤：先求变换矩阵T， 再求标准型

J = jordan(A)

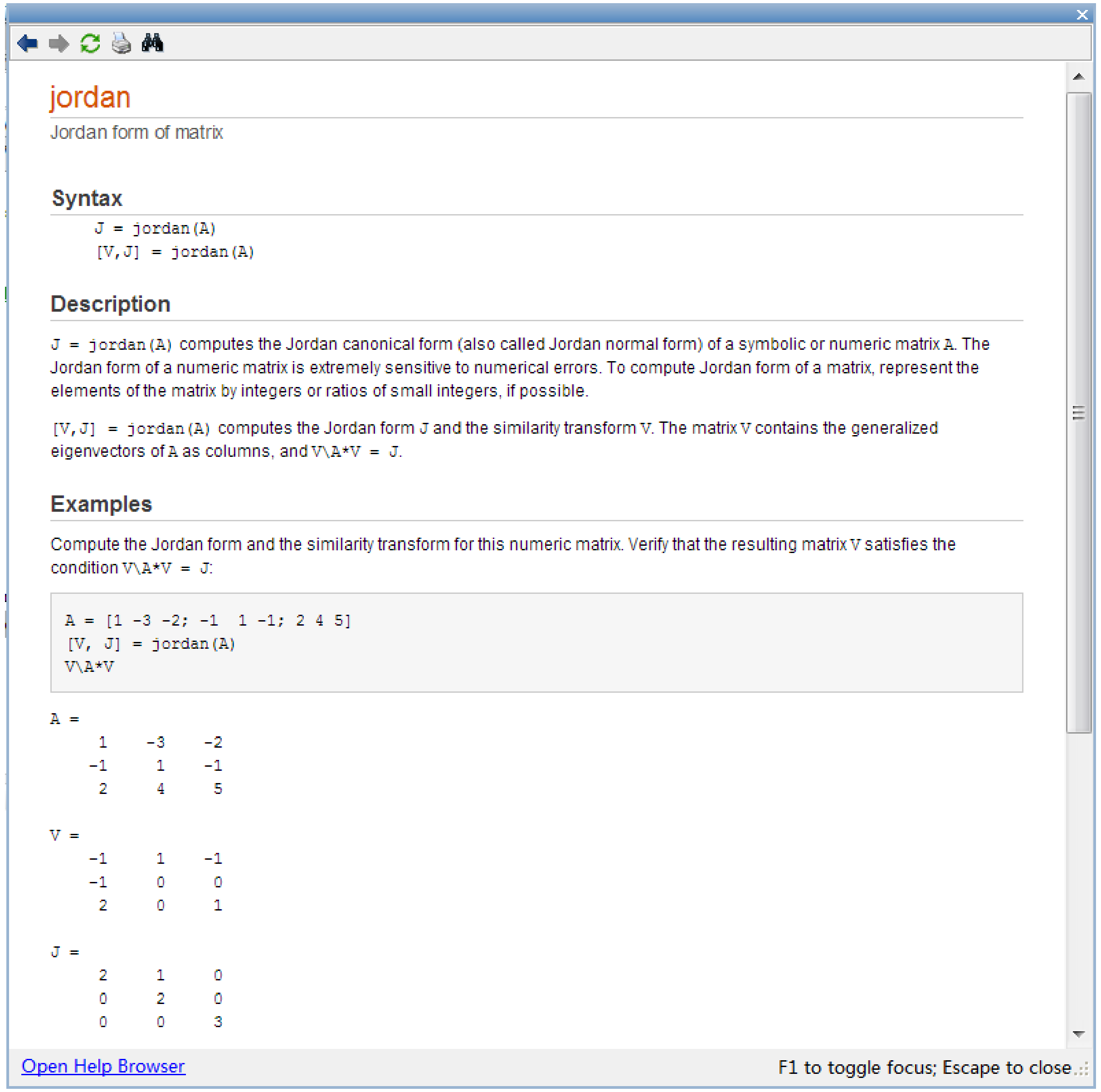
[V, J] = jordan(A)

T = inv(V)

sys_new = ss2ss(sys_original,T)

[An,Bn,Cn,Dn] = ss2ss(A,B,C,D,T)

其中，J为约旦矩阵，V为对应广义特征向量，即为所需变换阵T的逆



jordan

Jordan form of matrix

Syntax

```
J = jordan(A)
[V,J] = jordan(A)
```

Description

`J = jordan(A)` computes the Jordan canonical form (also called Jordan normal form) of a symbolic or numeric matrix `A`. The Jordan form of a numeric matrix is extremely sensitive to numerical errors. To compute Jordan form of a matrix, represent the elements of the matrix by integers or ratios of small integers, if possible.

`[V,J] = jordan(A)` computes the Jordan form `J` and the similarity transform `V`. The matrix `V` contains the generalized eigenvectors of `A` as columns, and `V\A*V = J`.

Examples

Compute the Jordan form and the similarity transform for this numeric matrix. Verify that the resulting matrix `V` satisfies the condition `V\A*V = J`:

```
A = [1 -3 -2; -1  1 -1; 2 4 5]
[V, J] = jordan(A)
V\A*V
```

```
A =

     1    -3    -2
    -1     1    -1
     2     4     5

V =

    -1     1    -1
    -1     0     0
     2     0     1

J =

     2     1     0
     0     2     0
     0     0     3
```

Open Help Browser

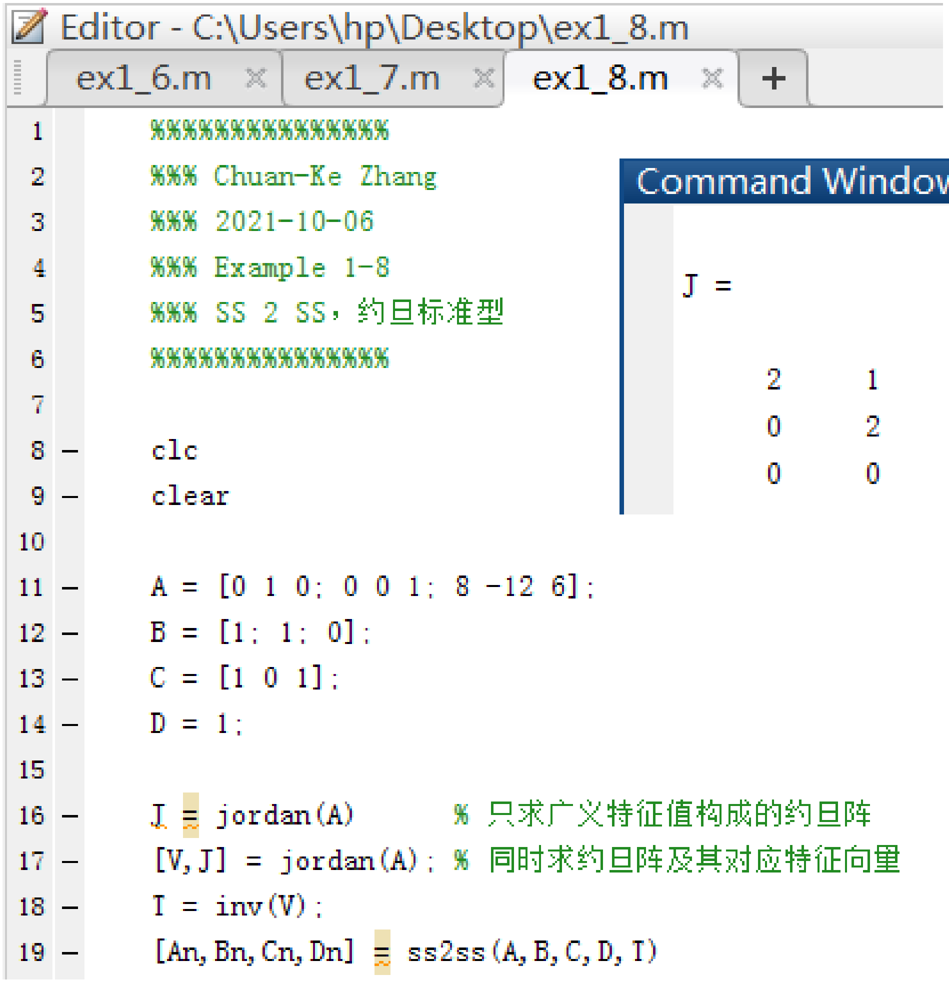
F1 to toggle focus; Escape to close

状态空间模型与状态空间模型的转化

例1-8：将以下状态空间模型化为对角线标准型

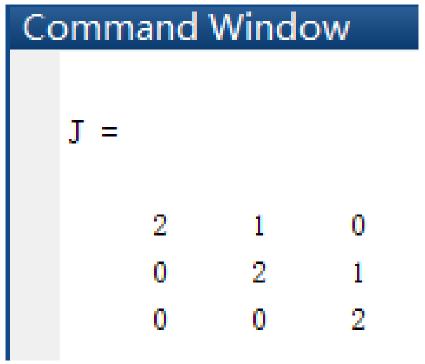
$$\begin{cases}\dot{x}=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 8 & -12 & 6\end{bmatrix}x+\begin{bmatrix}1\\ 1\\ 0\end{bmatrix}u\\ y=\begin{bmatrix}1 & 0 & 1\end{bmatrix}x+u\end{cases}$$

$$\begin{cases}\dot{x}=\begin{bmatrix}2 & 1 & 0\\ 0 & 2 & 1\\ 0 & 0 & 2\end{bmatrix}x+\begin{bmatrix}0.125\\ -0.25\\ 0\end{bmatrix}u\\ y=\begin{bmatrix}20 & 6 & 1\end{bmatrix}x+u\end{cases}$$

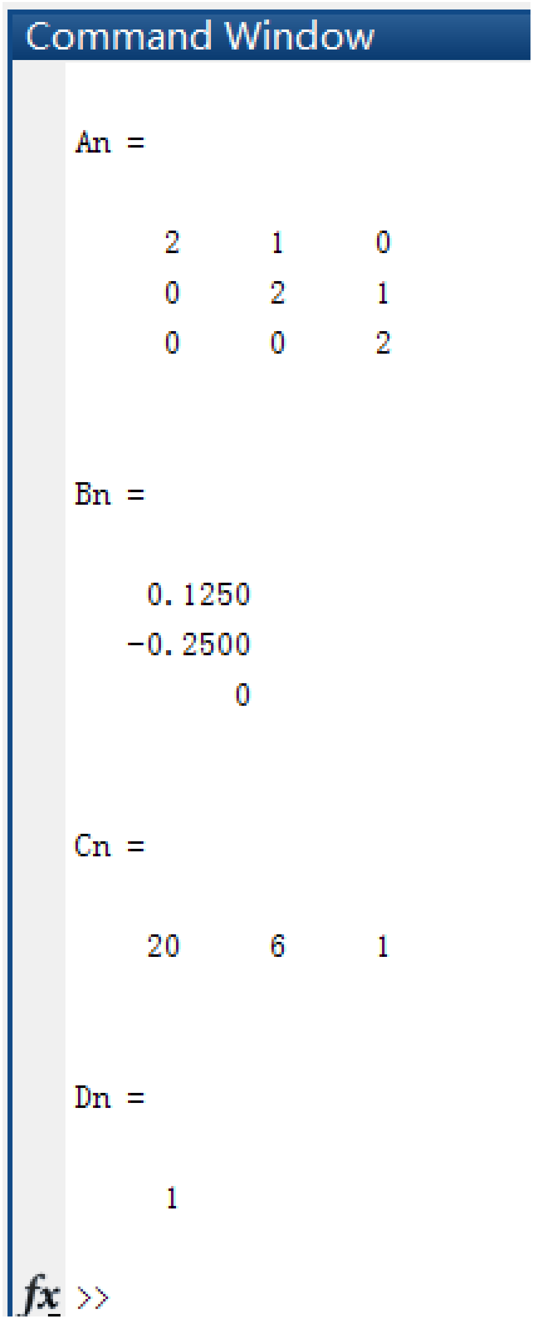

Editor - C:\Users\hp\Desktop\ex1_8.m

ex1_6.m ex1_7.m ex1_8.m

```
%%%%%%%%%%%%%%%%
%%% Chuan-Ke Zhang
%%% 2021-10-06
%%% Example 1-8
%%% SS 2 SS，约旦标准型
%%%%%%%%%%%%%%%%

clc
clear

A = [0 1 0; 0 0 1; 8 -12 6];
B = [1; 1; 0];
C = [1 0 1];
D = 1;

J = jordan(A)       % 只求广义特征值构成的约旦阵
[V,J] = jordan(A);  % 同时求约旦阵及其对应特征向量
T = inv(V);
[An,Bn,Cn,Dn] = ss2ss(A,B,C,D,T)
```


Command Window

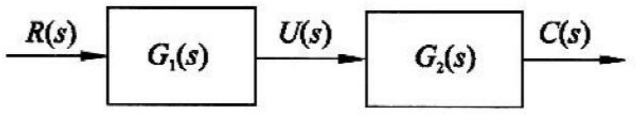
```
J =

     2     1     0
     0     2     1
     0     0     2
```


Command Window

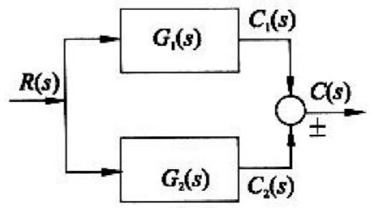
```
An =

     2     1     0
     0     2     1
     0     0     2


Bn =

    0.1250
   -0.2500
         0


Cn =

    20     6     1


Dn =

     1

>>
```

课内实验：模型部分

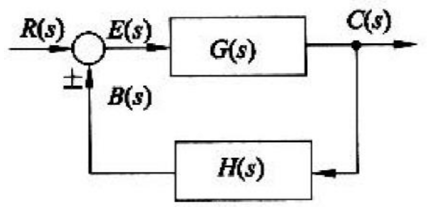
多子系统连接下的整体系统模型



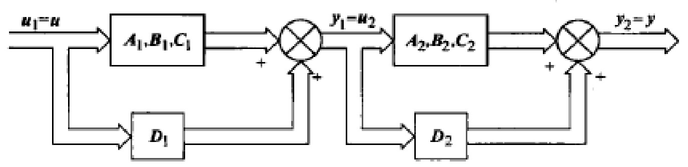
(a)

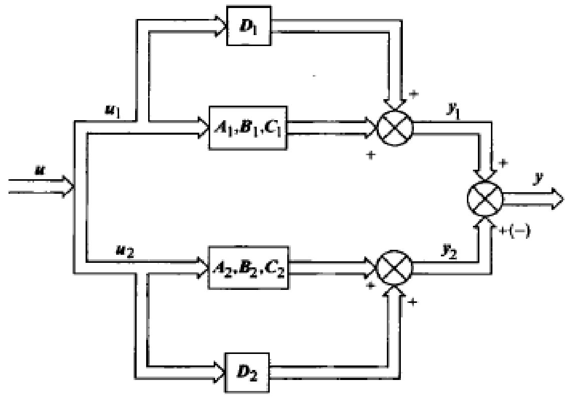


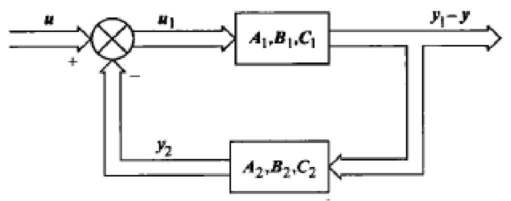
(a)



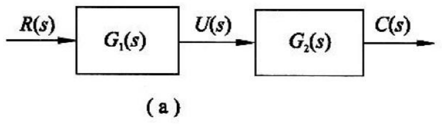
(a)

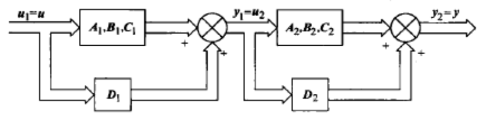






多子系统连接下的整体系统模型





series()函数

功能：实现两个子系统的串联

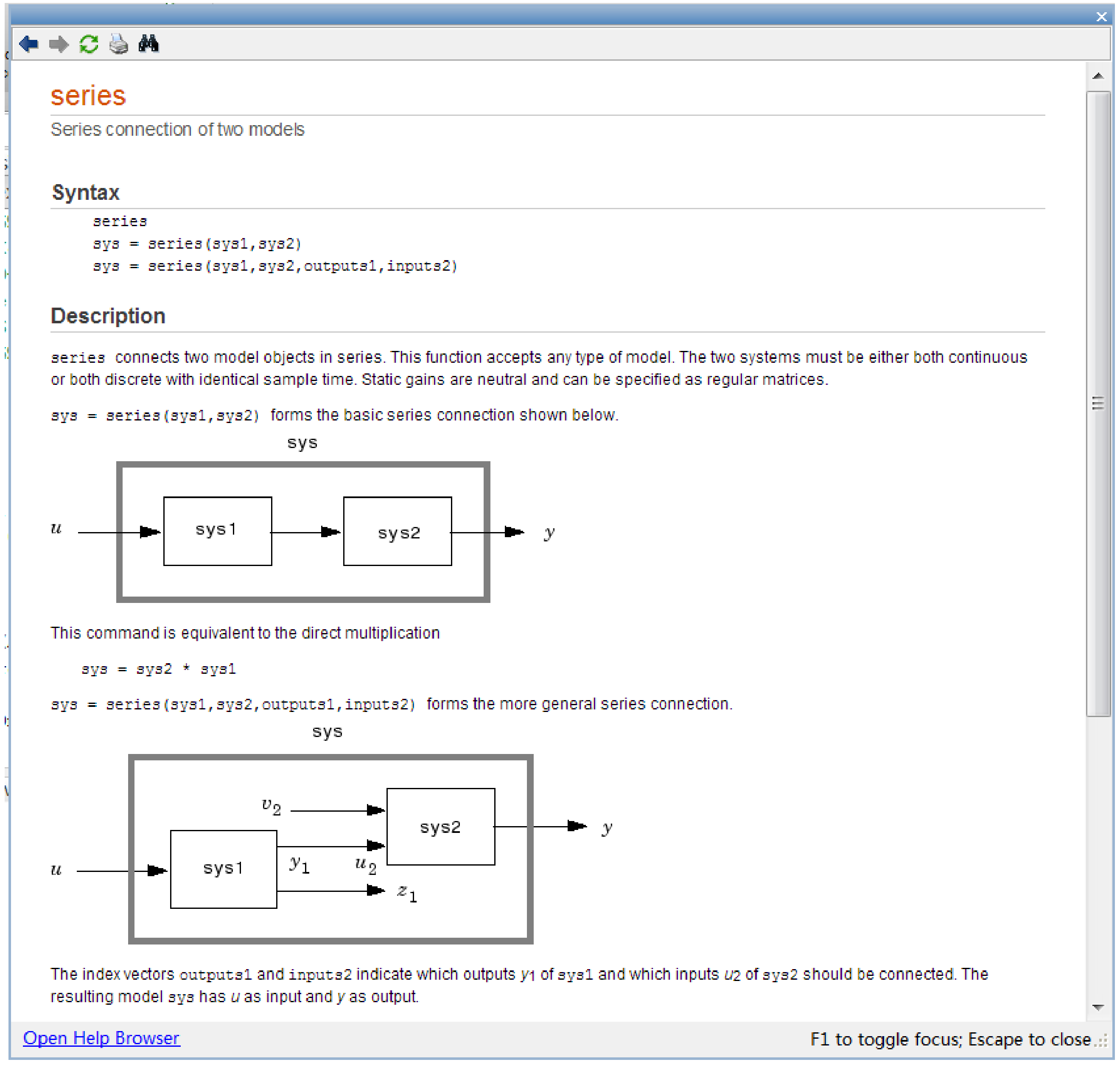
格式：

sys_new = series(sys1,sys2)

[A,B,C,D] = series(A1,B1,C1,D1, A2,B2,C2,D2)

[num,den] = series(num1,den1, num2,den2)

其中，sys1和sys2为同类表示子系统



series

Series connection of two models

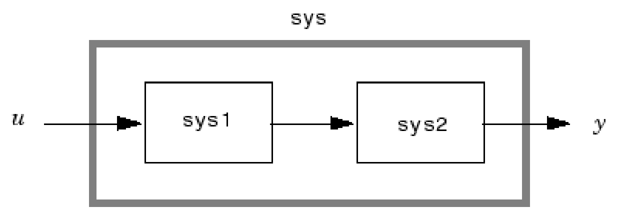
Syntax

```
series
sys = series(sys1,sys2)
sys = series(sys1,sys2,outputs1,inputs2)
```

Description

`series` connects two model objects in series. This function accepts any type of model. The two systems must be either both continuous or both discrete with identical sample time. Static gains are neutral and can be specified as regular matrices.

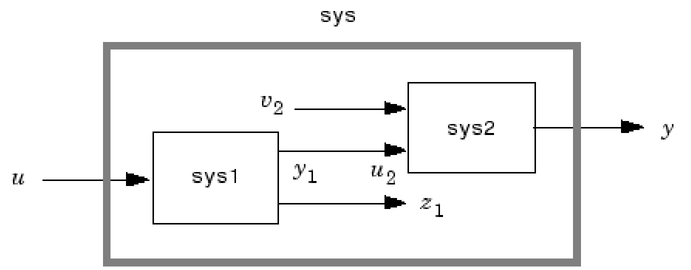
`sys = series(sys1,sys2)` forms the basic series connection shown below.



This command is equivalent to the direct multiplication

```
sys = sys2 * sys1
```

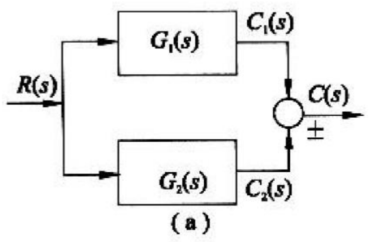
`sys = series(sys1,sys2,outputs1,inputs2)` forms the more general series connection.

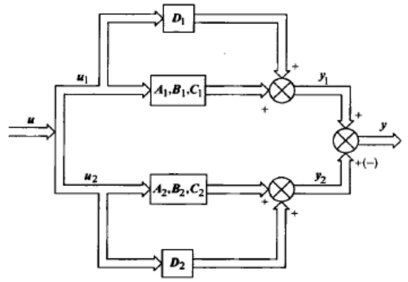


The index vectors `outputs1` and `inputs2` indicate which outputs y_1 of `sys1` and which inputs u_2 of `sys2` should be connected. The resulting model `sys` has u as input and y as output.

Open Help Browser F1 to toggle focus; Escape to close

多子系统连接下的整体系统模型



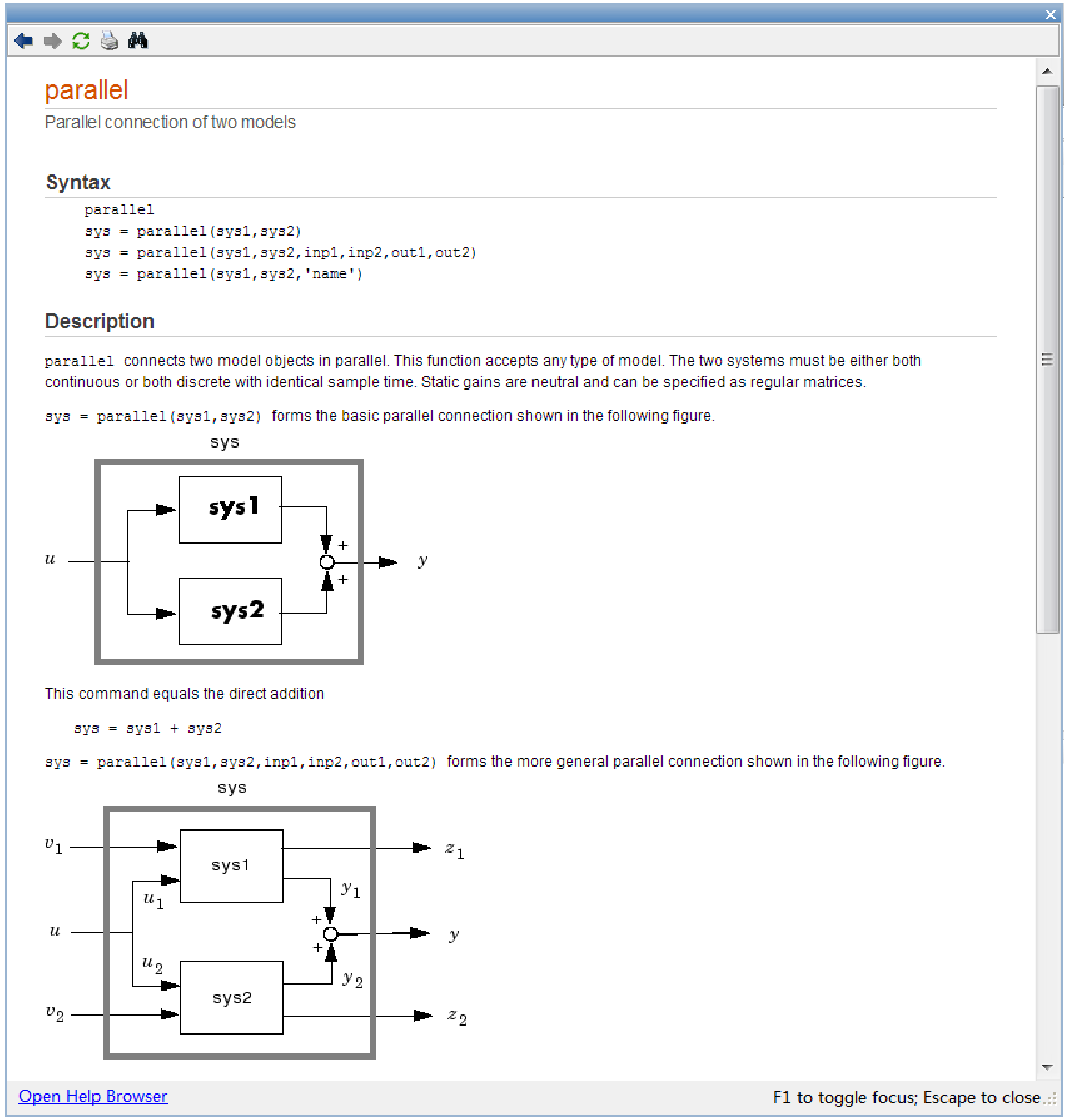


parallel()函数

功能：实现两个子系统的串联
格式：

sys_new = parallel (sys1,sys2)
[A,B,C,D] = parallel (A1,B1,C1,D1, A2,B2,C2,D2)
[num,den] = parallel (num1,den1, num2,den2)

其中，sys1和sys2为同类表示子系统



parallel

Parallel connection of two models

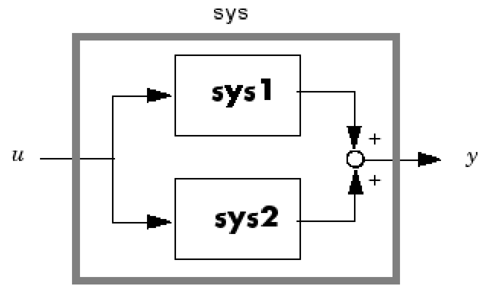
Syntax

```
parallel
sys = parallel(sys1,sys2)
sys = parallel(sys1,sys2,inp1,inp2,out1,out2)
sys = parallel(sys1,sys2,'name')
```

Description

`parallel` connects two model objects in parallel. This function accepts any type of model. The two systems must be either both continuous or both discrete with identical sample time. Static gains are neutral and can be specified as regular matrices.

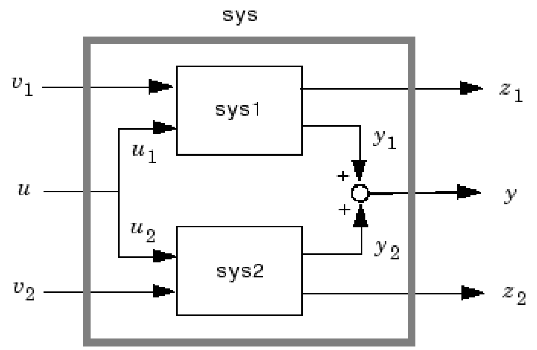
`sys = parallel(sys1,sys2)` forms the basic parallel connection shown in the following figure.



This command equals the direct addition

```
sys = sys1 + sys2
```

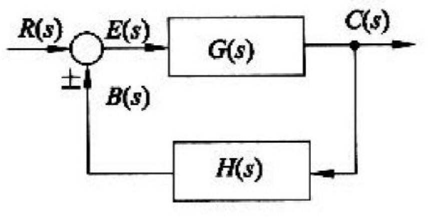
`sys = parallel(sys1,sys2,inp1,inp2,out1,out2)` forms the more general parallel connection shown in the following figure.



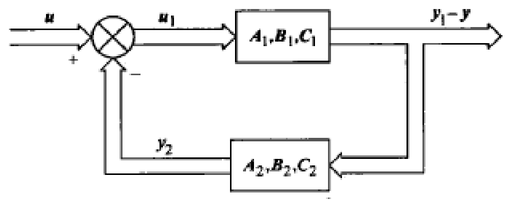
Open Help Browser

F1 to toggle focus; Escape to close

多子系统连接下的整体系统模型



(a)



feedback()函数

功能：实现两个子系统的反馈连接

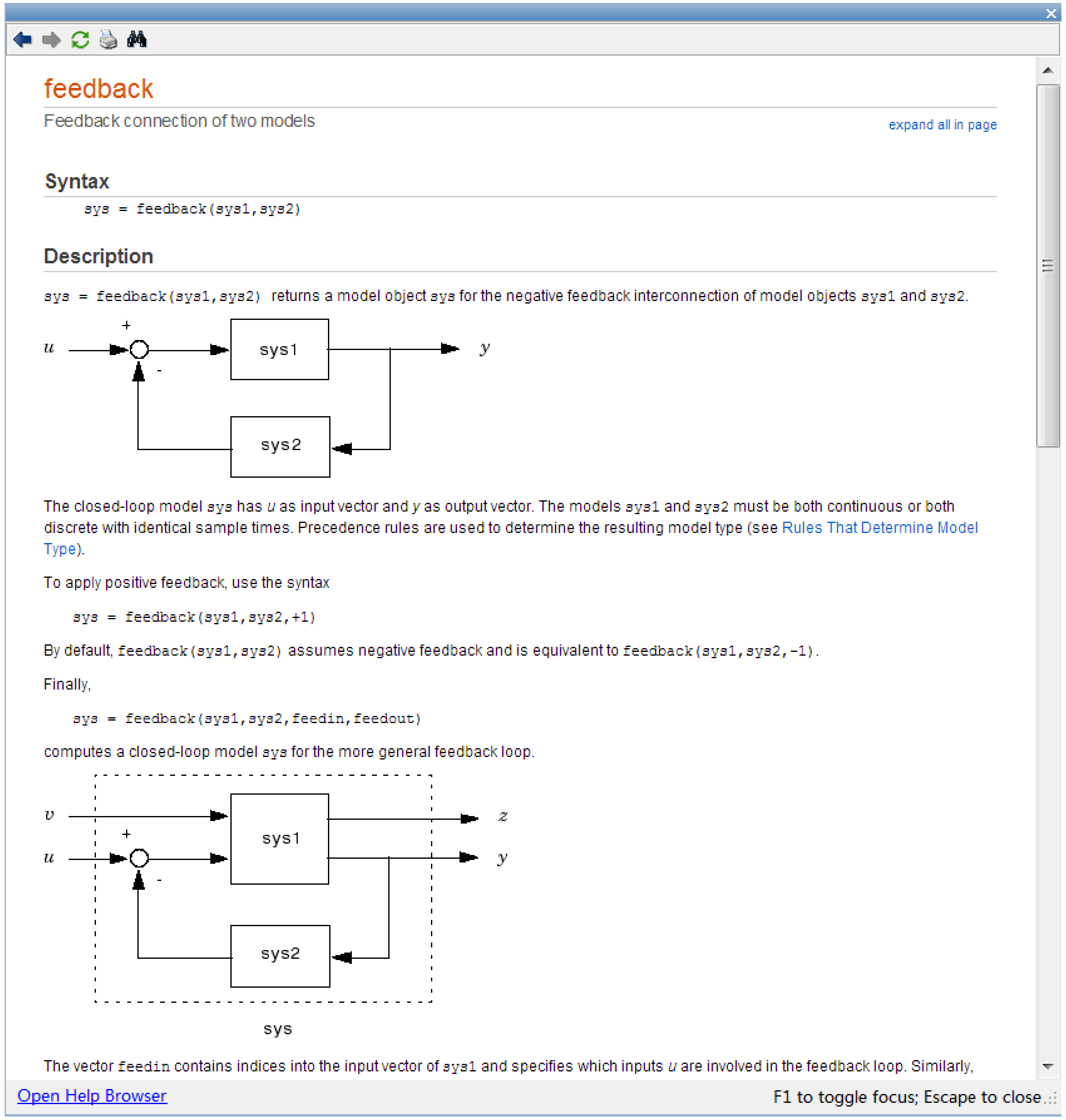
格式：

sys_new = feedback(sys1,sys2,sign)

[A,B,C,D] = feedback(A1,B1,C1,D1, A2,B2,C2,D2,sign)

[num,den] = feedback(num1,den1, num2,den2,sign)

其中，sys1和sys2为同类表示子系统，sign为反馈类型（1~正反馈、-1~负反馈）



feedback

Feedback connection of two models

expand all in page

Syntax

```
sys = feedback(sys1,sys2)
```

Description

`sys = feedback(sys1,sys2)` returns a model object `sys` for the negative feedback interconnection of model objects `sys1` and `sys2`.

The closed-loop model `sys` has *u* as input vector and *y* as output vector. The models `sys1` and `sys2` must be both continuous or both discrete with identical sample times. Precedence rules are used to determine the resulting model type (see Rules That Determine Model Type).

To apply positive feedback, use the syntax

```
sys = feedback(sys1,sys2,+1)
```

By default, `feedback(sys1,sys2)` assumes negative feedback and is equivalent to `feedback(sys1,sys2,-1)`.

Finally,

```
sys = feedback(sys1,sys2,feedin,feedout)
```

computes a closed-loop model `sys` for the more general feedback loop.

The vector `feedin` contains indices into the input vector of `sys1` and specifies which inputs *u* are involved in the feedback loop. Similarly,

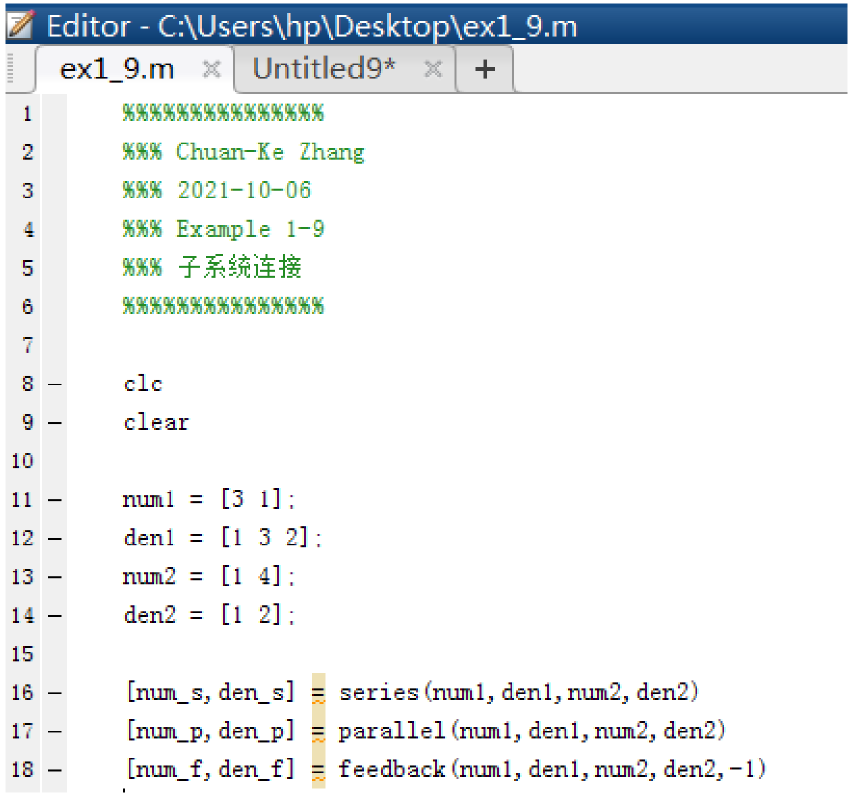
Open Help Browser

F1 to toggle focus; Escape to close

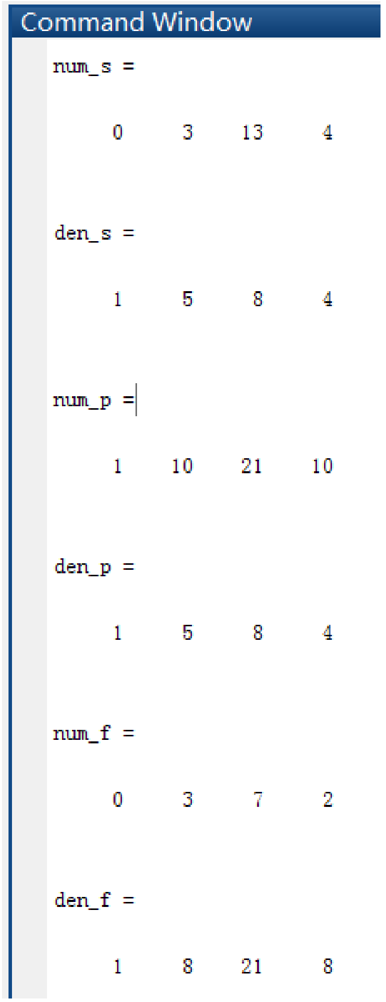
多子系统连接下的整体系统模型

例1-9：给定如下两个子系统，求其串联、并联、或负反馈连接后的新系统模型

$$W_1(s)=\frac{3s+1}{s^2+3s+2},\quad W_2(s)=\frac{s+4}{s+2}$$



```
%%%%%%%%%%%%%%%%
%%% Chuan-Ke Zhang
%%% 2021-10-06
%%% Example 1-9
%%% 子系统连接
%%%%%%%%%%%%%%%%

clc
clear

num1 = [3 1];
den1 = [1 3 2];
num2 = [1 4];
den2 = [1 2];

[num_s, den_s] = series(num1, den1, num2, den2)
[num_p, den_p] = parallel(num1, den1, num2, den2)
[num_f, den_f] = feedback(num1, den1, num2, den2, -1)
```



```
num_s =

     0     3    13     4


den_s =

     1     5     8     4


num_p =

     1    10    21    10


den_p =

     1     5     8     4


num_f =

     0     3     7     2


den_f =

     1     8    21     8
```

$$W_s(s)=\frac{3s^2+13s+4}{s^3+5s^2+8s+4}$$

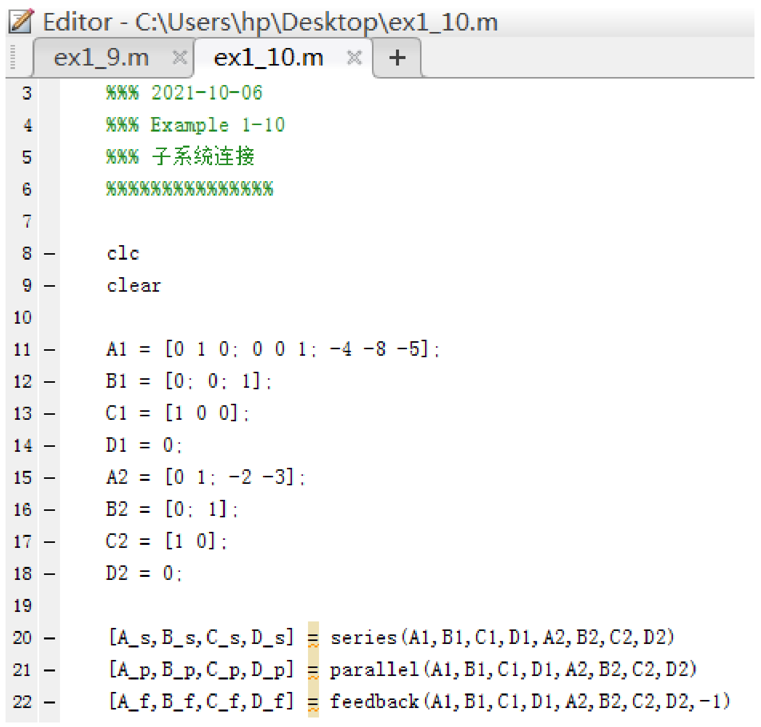
$$W_p(s)=\frac{s^3+10s^2+21s+10}{s^3+5s^2+8s+4}$$

$$W_f(s)=\frac{3s^2+7s+2}{s^3+8s^2+21s+8}$$

多子系统连接下的整体系统模型

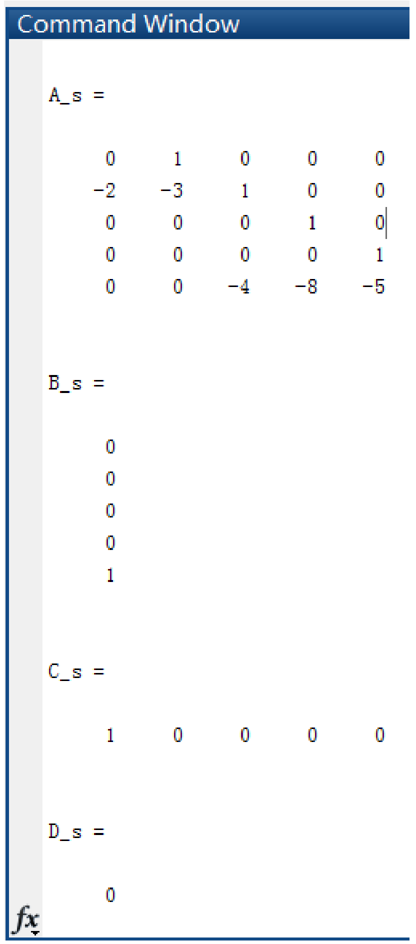
例1-9：给定如下两个子系统，求其串联、并联、或负反馈连接后的新系统模型

$$\begin{cases} \dot{x}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -4 & -8 & -5 \end{bmatrix} x_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_1 \\ y_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x_1 \end{cases} \qquad \begin{cases} \dot{x}_2 = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} x_2 + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_2 \\ y_2 = \begin{bmatrix} 1 & 0 \end{bmatrix} x_2 \end{cases}$$

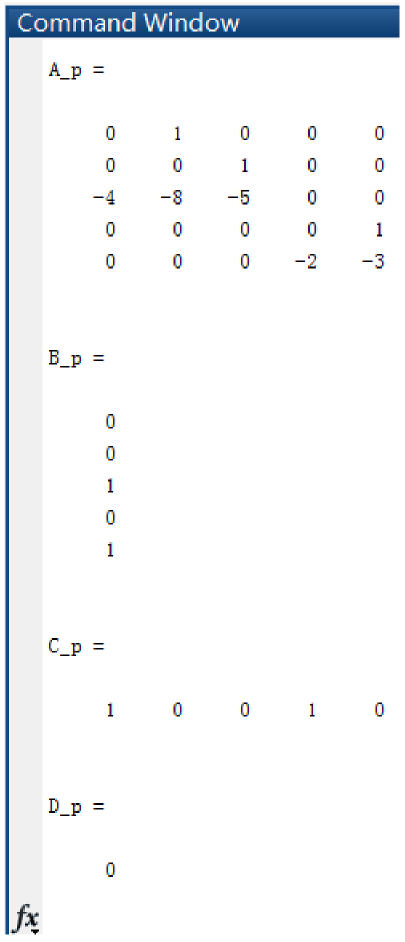

Editor - C:\Users\hp\Desktop\ex1_10.m

ex1_9.m　ex1_10.m　+

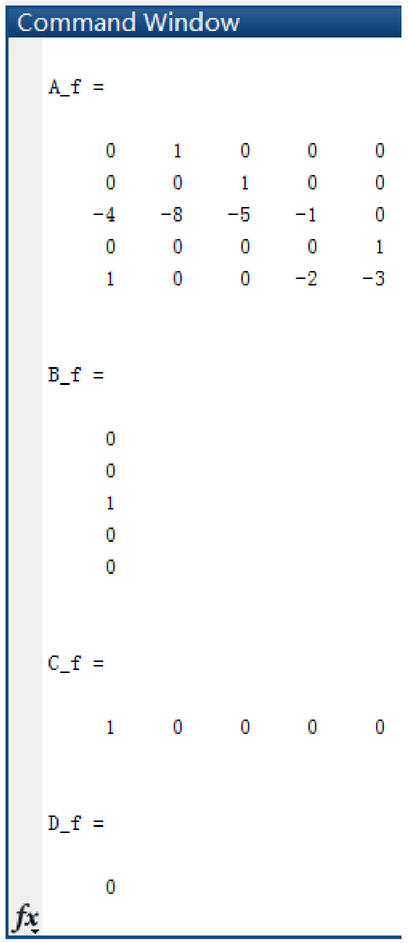
```matlab
%%% 2021-10-06
%%% Example 1-10
%%% 子系统连接
%%%%%%%%%%%%%%%%

clc
clear

A1 = [0 1 0; 0 0 1; -4 -8 -5];
B1 = [0; 0; 1];
C1 = [1 0 0];
D1 = 0;
A2 = [0 1; -2 -3];
B2 = [0; 1];
C2 = [1 0];
D2 = 0;

[A_s,B_s,C_s,D_s] = series(A1,B1,C1,D1,A2,B2,C2,D2)
[A_p,B_p,C_p,D_p] = parallel(A1,B1,C1,D1,A2,B2,C2,D2)
[A_f,B_f,C_f,D_f] = feedback(A1,B1,C1,D1,A2,B2,C2,D2,-1)
```


Command Window

```
A_s =

     0     1     0     0     0
    -2    -3     1     0     0
     0     0     0     1     0
     0     0     0     0     1
     0     0    -4    -8    -5


B_s =

     0
     0
     0
     0
     1


C_s =

     1     0     0     0     0


D_s =

     0
```


Command Window

```
A_p =

     0     1     0     0     0
     0     0     1     0     0
    -4    -8    -5     0     0
     0     0     0     0     1
     0     0     0    -2    -3


B_p =

     0
     0
     1
     0
     1


C_p =

     1     0     0     1     0


D_p =

     0
```


Command Window

```
A_f =

     0     1     0     0     0
     0     0     1     0     0
    -4    -8    -5    -1     0
     0     0     0     0     1
     1     0     0    -2    -3


B_f =

     0
     0
     1
     0
     0


C_f =

     1     0     0     0     0


D_f =

     0
```

本部分实验小结

- MATLAB使用
  - 如何在MATLAB中 编写代码、调试运行、查看结果、等常规操作
  - 如何利用simulink搭建系统结构框图

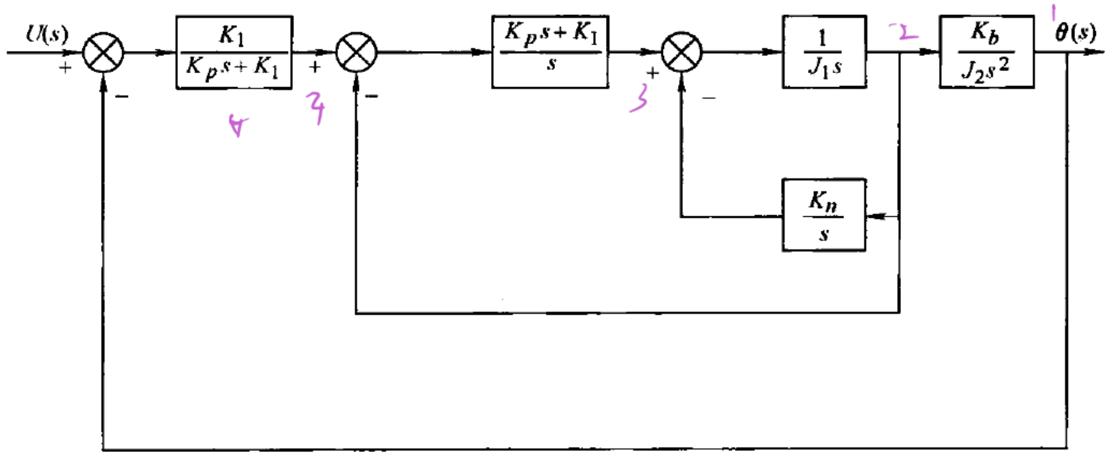
- 系统模型
  - 如何用MATLAB表示各类系统数学模型
  - 如何用MATLAB实现各类别模型之间的转化
  - 如何用MATLAB合并多个子系统生成整体系统模型

本部分实验练习题

- 作业一：简单练习题
  - 尝试利用MATLAB完成教材第1章习题，1-5、1-6、1-7、1-9、1-10、1-11

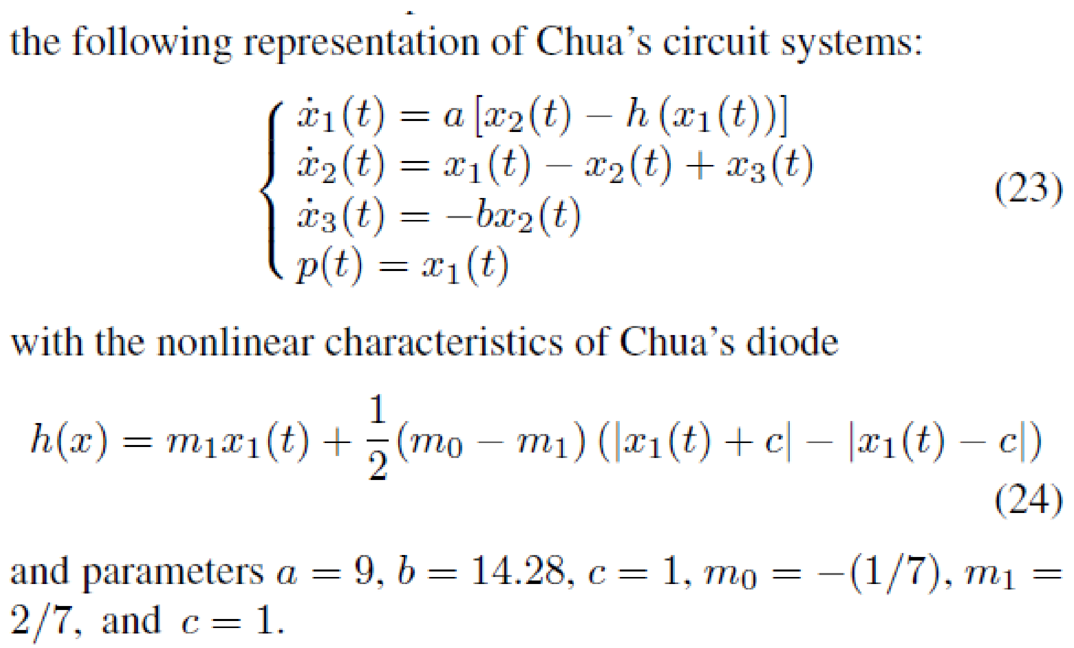
本部分实验练习题

- 作业二：综合应用题（验收、报告 必需包含题）
  - 利用MATLAB获取如下系统的 传递函数模型、状态空间模型
  - 利用Simulink搭建上述模型，并观察其单位阶跃响应（相关参数可自由取值）



本部分实验练习题

- 作业二：综合应用题（验收、报告 必需包含题）
  - 利用Simulink搭建如下系统结构框图
  - 选初始条件[-0.2; 0.3; 0.7]，观察系统的状态响应（用Simulink的示波器模块看）



the following representation of Chua's circuit systems:

$$\begin{cases} \dot{x}_1(t) = a\left[x_2(t) - h\left(x_1(t)\right)\right] \\ \dot{x}_2(t) = x_1(t) - x_2(t) + x_3(t) \\ \dot{x}_3(t) = -bx_2(t) \\ p(t) = x_1(t) \end{cases} \tag{23}$$

with the nonlinear characteristics of Chua's diode

$$h(x) = m_1 x_1(t) + \frac{1}{2}(m_0 - m_1)\left(|x_1(t) + c| - |x_1(t) - c|\right) \tag{24}$$

and parameters $a = 9$, $b = 14.28$, $c = 1$, $m_0 = -(1/7)$, $m_1 = 2/7$, and $c = 1$.

谢谢！