OTHER CANONICAL FORMS

MATLAB

```matlab
[Abn, Bln, Chn, Dln] = canon(A,B,C,D, 'Type',
Type = 'Companion',
Abn = [0 0 0 0 -a_0

1 0 0 0 -a_1

0 1 0 0 -a_2

0 0 1 0 -a_3

0 0 0 1 -a_{m-1}]

Type = 'modal'

Abn = [2 0 0 0

0 2 0 0

0 0 2 0

0 0 0 2] where \( \lambda_3 = \alpha + j\beta \)

\( \lambda_4 = \alpha - j\beta \)
Why change state variables?
If we use an equivalence transformation, the output-input remain the same:

\[ \begin{array}{c}
\text{SYSTEM A} \\
\downarrow \\
\text{SYSTEM B}
\end{array} \]

\[ u(t) \rightarrow y(t) \rightarrow v(t) \]

The only things that change are state variables. Why change the state variables? We change the state variables:

1) To decouple, as much as possible, the state variables (for example, change \( A \) to \( A_{\text{Jordan block form}} \) or change \( A \) to a modal form).

2) To rescale the control law state variables and/or system state variables so that the dynamic range differences are minimized for digital and analog implementations and

3) To rescale the state variables so they do not exceed op amp/power supply limitations and so that the dynamic range of low voltage varying state var. is increased.
% example4_5.m, MDH 10/10/00

A =[-0.1000  2.0000
     0   -1.0000]

B = [10.0000
     0.1000]

C = [0.1000  -1.0000]
D = [0]
eigval=eig(A)
eigval =
-0.1000
-1.0000
taumax= 1.0/min(abs(eigval))
taumax =
10

tmax= 7*taumax  % Time to approx. reach Steady State
tmax =
70  % Run simulation for 70 seconds.
taumin= 1.0/max(abs(eigval))
taumin =
1
T=taumin/10  % Sample interval - Rule of Thumb
T =
0.1000

t=0:T:tmax; u=ones(1,length(t)); X0=[0;0];
y=[]; tp=[]; X=[];

% LSIM - Help lsim output format
[y,X]= lsim(A,B,C,D,u,t,X0);
ymax=max(y)
ymax =
10.0907  % THE BOOK IS INCORECT, YMAX = 10.1 NOT 20
Xmax=max(X)
Xmax =
 101.9068  0.1000

% lsim(A,B,C,D,u,t,X0);
figure(2)
clf

% y and X are column vectors. y is one column and X is two columns.
plot(t,y,'k-',t,X(:,1),'k:',t,X(:,2),'k--')
grid
xlabel('TIME (sec)')
% We want x1max and x2max to be in the order of y_max = 10.1:
% x1max = 102, x2max = 0.1
% We rescale by setting x1bar=(0.1)x1 and x2bar=(100)x2; hence,
% Xbar = P*X and
P = [0.1 0; 0 100.]

\[
P = \begin{bmatrix}
0.1000 & 0 \\
0 & 100.0000
\end{bmatrix}
\]  \textit{DIFFERENT THAN THE BOOK}

[Abar, Bbar, Cbar, Dbar] = ss2ss(A, B, C, D, P)

Abar =
\[
\begin{bmatrix}
-0.1000 & 0.0020 \\
0 & -1.0000
\end{bmatrix}
\]

Bbar =
\[
\begin{bmatrix}
1 \\
10
\end{bmatrix}
\]

Cbar =
\[
\begin{bmatrix}
1.0000 & -0.0100
\end{bmatrix}
\]

Dbar =
\[
\begin{bmatrix}
0
\end{bmatrix}
\]

[y, Xbar] = lsim(Abar, Bbar, Cbar, Dbar, u, t, X0);
y_max = max(y)
y_max =
\[
10.0907
\]

Xbar_max = max(Xbar)
Xbar_max = [10.1907 10.0000]
figure(3)
clf
% y and X are column vectors. y is one column and X is two columns.
plot(t,y,'k-',t,Xbar(:,1),'k:',t,Xbar(:,2),'k--')
grid
xlabel('TIME (sec)')
- Passive element rescaling example based on rescaling the state variables

The state equation for Example 4.5 before rescaling is

\[
\begin{align*}
\dot{x}_1 &= -0.1x_1 + 2x_2 + 10u \\
\dot{x}_2 &= -x_2 + 0.1u
\end{align*}
\]

Diagramming using Laplace notation

An electrical circuit implementation follows:

**Noninverting low-pass with gain**

\[
\frac{\text{Vin}(s)}{\text{Out}(s)} = \frac{1 + \frac{R_2}{R_1}}{R_2 s + 1}
\]

**Inverting low-pass with gain**

\[
\frac{\text{Out}}{\text{Vin}} = -\frac{R_2 s + 1}{R_2 s + 1}
\]
\[ y = 0.1z_1 - z_2 \]

\[ z_1 = -0.1z_1 + 2z_2 + 10u \]

\[ z_2 = -z_2 + 0.1u \]

\[ -1/(s+1) \]

Rescaling as in example 4.5 yields:

\[ \overline{z} = -0.1\overline{z}_1 + 0.002\overline{z}_2 + u \]

\[ \overline{z}_2 = -\overline{z}_2 + 10u \]

\[ y = \overline{z}_1 - 0.01\overline{z}_2 \]
IF WE WANT TO RESCALE THE OUTPUT OF EITHER CIRCUIT ABOVE TO 15:

\[ K_m = \frac{Y_{\text{new-max}}}{Y_{\text{max}}} = \frac{15}{10.1} = 1.4851 \]

IF WE RESCALE THE LAST CIRCUITS (STATE VARIABLES RESCALED) OUTPUT TO 15,

ALL RESISTOR VALUES BECOME \( R_{\text{new}} = K_m R \) AND ALL CAPACITOR VALUES BECOME \( C_{\text{new}} = \frac{C}{K_m} \).

IF WE HAD INDUCTORS IN THE CIRCUIT, ALL \( L_{\text{new}} = K_m L \).

OF COURSE REscaling THE COMPONENTS, EITHER THE STATE-VARIABLES AND/OR THE OUTPUT MAY RESULT IN RLC VALUES THAT ARE NOT STANDARD VALUES.
REALIZATION

**Definition:**

Given \( \frac{Y(s)}{U(s)} = G(s) \), \( U(s) \in \mathbb{R}^n \), \( Y(s) \in \mathbb{R}^m \)

If a state equation (defined by matrices \( A, B, C, D \)) can be found such that

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

Then the \( G(s) \) matrix is said to be "realizable."

\( G(s) \) is realizable if it is a "proper" rational matrix, i.e., every entry in the \( G(s) \) is at least proper (\( \text{deg}(G(s)) \leq \text{deg}(D(s)) \) where \( \text{deg}(G(s)) \in G(s) \)).

**Examples**

\[
G(s) = \begin{bmatrix}
\frac{-s+1}{s+2} & 1 \\
\frac{s}{s+2} & -1
\end{bmatrix} \quad \text{\( G(s) \) is proper; hence, realizable (A, B, C, D can be found)}
\]

\[
G(s) = \begin{bmatrix}
\frac{s}{s^2} \\
\frac{s}{s+1}
\end{bmatrix} \quad \text{\( G(s) \) is not proper; hence, a mapping of \( U \) to \( Y \) is not possible and \( G(s) \) is not realizable without changes to \( G(s) \).}
\]

*For every LTI state equation, a \( G(s) \) exists.*