chapter12_3_2 Modeling in the Frequency Domain for Example 12.7

```
% Onwubolu, G. C.
% Mechatronics: Principles & Applications
% Elsevier
%
% Mechatronics: Principles & Applications Toolbox Version 1.0
% Copyright © 2005 by Elsevier
%
% Chapter 12.3: Modeling in the Time Domain
%
% Example 12.7: Transfer functions represented either by numerator and
% denominator or an LTI object can be converted to state space. For
numerator
% and denominator representation, the conversion can be implemented using
% [A,B,C,D] = tf2ss(num,den). The A matrix is returned in a form called the
% controller canonical form, which will be explained in Chapter 5 in the text.
To
% obtain the phase-variable form, [Ap, Bp, Cp, Dp], we perform the following
% operations: Ap = inv(P)*A*P; Bp = inv(P)*B; Cp = C*P, Dp  = D, where P is
a matrix
% with 1's along the anti-diagonal and 0's elsewhere. These transformations
will be
% explained in Chapter 5. The command inv(X) finds the inverse of a square
% matrix. The symbol * signifies multiplication. For systems represented as
LTI
% objects, the command ss(F), where F is an LTI transfer-function object, can
be used
% to convert F to a state-space object. Let us look at Example 3.4 in the text.
For the
% numerator-denominator representation, notice that the MATLAB response
associates
% the gain, 24, with the vector C rather than the vector B as in the example in
the text.
% Both representations are equivalent. For the LTI transfer-function object,
the
% conversion to state space does not yield the phase-variable form. The
result is
% a balanced model that improves the accuracy of calculating eigenvalues,
which are
% covered in Chapter 4. Since ss(F) does not yield familiar forms of the state
% equations (nor is it possible to easily convert to familiar forms), we will have
% limited use for that transformation at this time.

'Example 12.7'               % Display label.
'Numerator-denominator representation conversion'
                             % Display label.
'Controller canonical form'        % Display label.
num=30*[1 30 20 50];              % Define numerator of G(s)=C(s)/R(s).
den=[1 600 500 75 40];             % Define denominator of G(s).
```

```
[A,B,C,D]=tf2ss(num,den)            % Convert G(s) to controller
                                 % canonical form,
                                 % store matrices A, B, C, D, and
                                 % display.
'Phase-variable form'            % Display label.
P=[0 0 0 1;0 0 1 0;0 1 0 0;1 0 0 0]; % Form transformation matrix.
Ap=inv(P)*A*P                    % Form A matrix, phase-variable form.
Bp=inv(P)*B                      % Form B vector, phase-variable form.
Cp=C*P                           % Form C vector, phase-variable form.
Dp=D                             % Form D phase-variable form.
'LTI object representation'      % Display label.
T=tf(num,den)                    % Represent T(s)=24/(s^3+9s^2+26s+24)
                                 % as an LTI transfer-function object.
Tss=ss(T)                        % Convert T(s) to state space.
Pause
```