COMPUTER ASSIGNMENT

DESIGN OF SEAT BELTS FOR VEHICLE CRASH TEST

PROBLEM:
The dummy in the figure shown below is driving his new VW-Rabbit into a wall! Will his shock-absorbing bumper \( k_2, b_2 \) and his seat belts \( k_1, b_1 \) prevent him from hitting the windshield without breaking his collarbone?

DATA ON INJURIES (SAE Handbook)
Seat belts must be tested to 3000 lbs. \((1.334 \times 10^4) \text{ N}\)
Chest can sustain a force of 1500 lbs. distributed over 30 in\(^2\).
Seat belt effective area = 30 in\(^2\)
Shoulder strap-seat belt combination = 60 in\(^2\)

PHYSICAL PARAMETERS

\[
M = 1500 \text{ Kg, } k_1 = 1 \times 10^4 \text{ N/m, } b_1 = 500 \text{ N-s/m} \\
m = 100 \text{ kg, } k_2 = 3 \times 10^5 \text{ N/m, } b_2 = 8 \times 10^4 \text{ N-s/m}
\]

Shown above is an engineering model of the car, person, seat belts, bumper etc. Transform the engineering model of reality into a computer model for simulation using SIMULINK and CAMP/G/MATLAB. Perform interactive simulation and generate numerical and graphical outputs using considering the design criteria. Interpret the results make design decisions and perform
PROBLEM:
The dummy in the figure shown below is driving his new VW-Rabbit into a wall! Will his shock-absorbing bumper \( (k_2, b_2) \) and his seat belts \( (k_1, b_1) \) prevent him from hitting the windshield without breaking his collarbone?

DATA ON INJURIES (SAE Handbook)
Seat belts must be tested to 3000 lbs. \((1.334 \times 10^4) \text{ N}\)
Chest can sustain a force of 1500 lbs. distributed over 30 in\(^2\).
Seat belt effective area = 30 in\(^2\)
Shoulder strap-seat belt combination = 60 in\(^2\)

PHYSICAL PARAMETERS

\[
\begin{align*}
M &= 1500 \text{ Kg}, \quad k_1 = 1 \times 10^4 \text{ N/m}, \quad b_1 = 500 \text{ N-s/m} \\
m &= 100 \text{ kg}, \quad k_2 = 3 \times 10^5 \text{ N/m}, \quad b_2 = 8 \times 10^4 \text{ N-s/m}
\end{align*}
\]

Shown above is an engineering model of the system. Transform the engineering model of reality into a computer model for simulation using SIMULINK and SIMULINK/MATLAB. Perform interactive simulation and generate numerical and graphical outputs.
PROBLEM:
The dummy in the figure shown below is driving his new VW-Rabbit into a wall! Will his shock-absorbing bumper (k2,b2) and his seat belts (k1, b1) prevent him from hitting the windshield without breaking his collarbone?

DATA ON INJURIES (SAE Handbook)
Seat belts must be tested to 3000 lbs. (1.334 x 10^4) N
Chest can sustain a force of 1500 lbs. distributed over 30 in^2.
Seat belt effective area = 30 in^2
Shoulder strap-seat belt combination = 60 in^2

PHYSICAL PARAMETERS

M= 1500 Kg, k1= 1 x 10^4 N/m, b1=500 N-s/m
m= 100 kg, k2= 3 x 10^5 N/m, b2=8 x 10^4 N-s/m

Shown above is an engineering model of the dummy, bumper, etc. Transform the engineering model of reality into a computer model for simulation using SIMULINK and CAMPIC/MATLAB. Perform interactive simulation and generate numerical and graphical outputs.
PROBLEM:
The dummy in the figure shown below is driving his new VW-Rabbit into a wall! Will his shock-absorbing bumper (k2, b2) and his seat belts (k1, b1) prevent him from hitting the windshield without breaking his collarbone?

DATA ON INJURIES (SAE Handbook)
Seat belts must be tested to 3000 lbs. (1.334 x 104) N
Chest can sustain a force of 1500 lbs. distributed over 30 in².
Seat belt effective area = 30 in²
Shoulder strap-seat belt combination = 60 in²

PHYSICAL PARAMETERS

\[ M = 1500 \text{ Kg, } k_1 = 1 \times 10^4 \text{ N/m, } b_1 = 500 \text{ N-s/m} \]
\[ m = 100 \text{ kg, } k_2 = 3 \times 10^5 \text{ N/m, } b_2 = 8 \times 10^4 \text{ N-s/m} \]

Shown above is an engineering model of the system with spring, the bumper etc. Transform the engineering model of reality into a computer model for simulation using SIMULINK and CAMPG/MATLAB. Perform interactive simulation and generate numerical and graphical outputs.
PROBLEM:
The dummy in the figure shown below is driving his new VW-Rabbit into a wall! Will his shock-absorbing bumper \((k_2, b_2)\) and his seat belts \((k_1, b_1)\) prevent him from hitting the windshield without breaking his collarbone?

DATA ON INJURIES (SAE Handbook)
Seat belts must be tested to 3000 lbs. \((1.334 \times 10^4)\) N
Chest can sustain a force of 1500 lbs. distributed over 30 in\(^2\).
Seat belt effective area = 30 in\(^2\)
Shoulder strap-seat belt combination = 60 in\(^2\)

PHYSICAL PARAMETERS

\[ M = 1500 \text{ Kg}, \quad k_i = 1 \times 10^4 \text{ N/m}, \quad b_i = 500 \text{ N-s/m} \]
\[ m = 100 \text{ Kg}, \quad k_2 = 3 \times 10^5 \text{ N/m}, \quad b_2 = 8 \times 10^4 \text{ N-s/m} \]

Shown above is an engineering model of the dummy, bumper, seat belt, etc. Transform the engineering model of reality into a computer model for simulation using SIMULINK and CAMEO/MATLAB. Perform interactive simulation and generate numerical and graphical outputs.
PROBLEM:
The dummy in the figure shown below is driving his new VW-Rabbit into a wall! Will his shock-absorbing bumper (k2, b2) and his seat belts (k1, b1) prevent him from hitting the windshield without breaking his collarbone?

DATA ON INJURIES (SAE Handbook)
Seat belts must be tested to 3000 lbs. (1.334 x 104) N
Chest can sustain a force of 1500 lbs. distributed over 30 in².
Seat belt effective area = 30 in²
Shoulder strap-seat belt combination = 60 in²

PHYSICAL PARAMETERS

\[ M = 1500 \text{ Kg}, \quad k_1 = 1 \times 10^4 \text{ N/m}, \quad b_1 = 500 \text{ N-s/m} \]
\[ m = 100 \text{ Kg}, \quad k_2 = 3 \times 10^5 \text{ N/m}, \quad b_2 = 8 \times 10^4 \text{ N-s/m} \]

Shown above is an engineering model of the vehicle with its bumper and the dummy in contact. Transform the engineering model of reality into a computer model for simulation using SIMULINK and CAMPG/MATLAB. Perform interactive simulation and generate numerical and graphical outputs.
PROBLEM:
The dummy in the figure shown below is driving his new VW-Rabbit into a wall! Will his shock-absorbing bumper ($k_2$,$b_2$) and his seat belts ($k_1$, $b_1$) prevent him from hitting the windshield without breaking his collarbone?

DATA ON INJURIES (SAE Handbook)
Seat belts must be tested to 3000 lbs. (1.334 x 104) N
Chest can sustain a force of 1500 lbs. distributed over 30 in².
Seat belt effective area = 30 in²
Shoulder strap-seat belt combination = 60 in²

PHYSICAL PARAMETERS

\[ M = 1500 \text{ Kg}, \quad k_1 = 1 \times 10^4 \text{ N/m}, \quad b_1 = 500 \text{ N-s/m} \]
\[ m = 100 \text{ kg}, \quad k_2 = 3 \times 10^5 \text{ N/m}, \quad b_2 = 8 \times 10^4 \text{ N-s/m} \]

Shown above is an engineering model of the dummy interacting with the bumper etc. Transform the engineering model of reality into a computer model for simulation using SIMULINK and SIMULINK/MATLAB. Perform interactive simulation and generate numerical and graphical outputs.
PROBLEM:
The dummy in the figure shown below is driving his new VW-Rabbit into a wall! Will his shock-absorbing bumper (k2, b2) and his seat belts (k1, b1) prevent him from hitting the windshield without breaking his collarbone?

DATA ON INJURIES (SAE Handbook)
Seat belts must be tested to 3000 lbs. (1.334 x 104) N
Chest can sustain a force of 1500 lbs. distributed over 30 in²
Seat belt effective area = 30 in²
Shoulder strap-seat belt combination = 60 in²

PHYSICAL PARAMETERS

\[ M = 1500 \text{ Kg}, \quad k_1 = 1 \times 10^4 \text{ N/m}, \quad b_1 = 500 \text{ N-s/m} \]
\[ m = 100 \text{ Kg}, \quad k_2 = 3 \times 10^5 \text{ N/m}, \quad b_2 = 8 \times 10^4 \text{ N-s/m} \]

Shown above is an engineering model of the dummy, car, bumper, and bumper etc. Transform the engineering model of reality into a computer model for simulation using SIMULINK and CAMPG/MATLAB. Perform interactive simulation and generate numerical and graphical outputs.
PROBLEM:
The dummy in the figure shown below is driving his new VW-Rabbit into a wall! Will his shock-absorbing bumper \( b_2 = b_2 \) and his seat belts \( b_1 = b_1 \) prevent him from hitting the windshield without breaking his collarbone?

DATA ON INJURIES (SAE Handbook)
Seat belts must be tested to 3000 lbs. \((1.334 \times 10^4) \text{ N}\)
Chest can sustain a force of 1500 lbs. distributed over 30 in \(^2\).
Seat belt effective area = 30 in \(^2\)
Shoulder strap-seat belt combination = 60 in \(^2\)

PHYSICAL PARAMETERS

\[
M = 1500 \text{ Kg, } k_1 = 1 \times 10^4 \text{ N/m, } b_1 = 500 \text{ N-s/m}
\]
\[
m = 100 \text{ kg, } k_2 = 3 \times 10^5 \text{ N/m, } b_2 = 8 \times 10^4 \text{ N-s/m}
\]

We need the analysis when both are in contact.

Wall + car

Shown above is an engineering model of the dummy, the car, the bumper etc. Transform the engineering model of reality into a computer model for simulation using SIMULINK and CAMPG/MATLAB. Perform interactive simulation and generate numerical and graphical outputs.
1. Recognize Elements
1. Recognize Elements
1. Recognize Elements
1. Recognize Elements
1. Recognize Elements
1. Recognize Elements
1. Recognize Elements
1. Recognize Elements
1. Recognize Elements
2. Use I functions for V's
1. Recognize Elements
2. Use I functions for v's

1. \( k_2 \)

2. \( b_2 \)

\[ V_{wo} \]
1. Recognize Elements
2. Use I functions for v's
1. Recognize Elements
2. Use \( I \) functions for \( v \)'s
1. Recognize Elements
2. Use $f(t)$ functions for $v(t)$
1. Recognize Elements
2. Use I functions for V's
3. Attach Elements
1. Recognize Elements
2. Use I functions for V's
3. Attach elements

\[ \text{Diagram with labels: } k_2, b_2, M, \text{ etc.} \]
1. Recognize Elements
2. Use I functions for V's
3. Attach elements
1. Recognize Elements
2. Use I functions for V's
3. Attach Elements
1. Recognize Elements
2. Use I functions for V's
3. Attach elements

m
C
b_1
k_2
k_2
M
b_2

I
ν_{0}
ν_{wall}
ν_{con}
1. Recognize Elements
2. Use I functions for v's
3. Attach elements
1. Recognize Elements
2. Use I functions for V's
3. Attach elements
1. Recognize Elements
2. Use I functions for V's
3. Attach elements
1. Recognize Elements
2. Use I functions for V's
3. Attach elements
4. Use 0's for different in velocities
1. Recognize Elements
2. Use I functions for v's
3. Attach elements
4. Use 0's for different in velocities
1. Recognize Elements
2. Use I functions for V's
3. Attach Elements
4. Use 0's for different in velocities
1. Recognize Elements
2. Use \( I \) functions for \( V \)'s
3. Attach elements
4. Use O's for different in velocities

\[ I - I \]

V_{V\text{rel}}

Wall

0

\[ 0 \]

1 - 1
1. Recognize Elements
2. Use $I$ functions for $V$'s
3. Attach elements
4. Use $0$'s for differences in velocities

\[ I \rightarrow \text{inertia} \]

\[ c \quad \text{for contact} \]

\[ I \quad \text{for invariance} \]
1. Recognize Elements
2. Use I functions for V's
3. Attach elements
4. Use 0's for different in velocities

\[ I \quad I \quad v_{wall} \quad v \quad v_{0} \quad v_{1} \quad v_{2} \]
1. Recognize Elements
2. Use I functions for v's
3. Attach elements
4. Use 0's for different in velocities

\[ I \rightarrow \text{V}_\text{term} \]

\[ \text{V}_\text{wall} = \text{V} \]
1. Recognize Elements
2. Use I functions for v's
3. Attach elements
4. Use 0's for different in velocities

\[ I \to \text{V} \quad \text{V}_{\text{wall}} \quad \text{V} \quad \text{V}_{\text{car}} \]
1. Recognize Elements
2. Use I functions for v's
3. Attach elements
4. Use 0's for different in velocities

\[ \text{V}_{\text{wall}} - \text{V}_{\text{can}} = \text{V}_{\text{wall}} - \text{V}_{\text{can}} \]
1. Recognize Elements
2. Use I functions for v's
3. Attach elements
4. Use 0's for different in velocities
1. Recognize Elements
2. Use \( I \) functions for \( v \)'s
3. Attach element
4. Use \( 0 \)'s for different in velocities
1. Recognize Elements
2. Use I functions for $v$'s
3. Attach elements
4. Use $0$'s for different in velocities

$\text{St} \quad \text{I} \quad \text{V}_{\text{wall}} \\
\text{I} \quad \text{V}_{\text{can}} \quad \text{V}_{\text{wall}}$

$\text{I} \quad \text{V}_{\text{can}} \quad \text{St}$
1. Recognize elements
2. Use I functions for v’s
3. Attach elements
4. Use o’s for different in velocities
1. Recognize Elements
2. Use I functions for v's
3. Attach elements
4. Use 0's for differences in velocities

\[ I - I \text{ (currents)} \]

\[ V_{\text{wall}} - V \]

\[ V_{\text{can}} - V \]

\[ S - I \text{ (forces)} \]
1. Recognize Elements
2. Use I functions for \( V \)’s
3. Attach elements
4. Use 0’s for different in velocities

\[ V_{\text{wall}} - 0 \]

\[ V_{\text{can}} - V_{\text{wall}} \]
1. Recognize elements
2. Use I functions for v's
3. Attach elements
4. Use 0's for different in velocities
5. 

\[ v_{wall} = \frac{V_{in}}{R_{wall}} \]
\[ v_{can} = \frac{V_{in}}{R_{can}} \]
1. **Recognize Elements**
2. Use \( I \) functions for \( v \)'s
3. **Attach elements**
4. Use \( 0 \)'s for different in velocities
5. **Attach elements that**
1. Recognize Elements
2. Use I functions for v's
3. Attach elements
4. Use 0's for differences in velocities
5. Attach elements that express the relative velocities

\[ V_{wall - v} \]
\[ V_{can - v} \]
1. Recognize Elements
2. Use I functions for v's
3. Attach elements
4. Use 0's for different in velocities
5. Attach elements that express the relative velocities

\[ \text{Diagram with labeled elements} \]
1. Recognize elements.
2. Use I functions for v's.
3. Attach elements.
4. Use 0's for different in velocities.
5. Attach elements that express the relative velocities.

\[ v_{wall} - v_{can} \]
1. Recognize Elements
2. Use I functions for v's
3. Attach elements
4. Use 0's for different in velocities
5. Attach elements that express the relative velocities

\[ I \rightarrow \begin{align*}
& \rightarrow 0 \\
& \rightarrow I \\
& \rightarrow C \\
& \rightarrow R \\
& \rightarrow V_{\text{wall}} \\
& \rightarrow V_{\text{can}} \\
& \rightarrow V_{\text{can}} - V_{\text{wall}}
\end{align*} \]
1. Recognize Elements
2. Use I functions for v's
3. Attach elements
4. Use 0's for different in velocities
5. Attach elements that express the relative velocities
1. Recognize Elements
2. Use I functions for v's
3. Attach elements
4. Use 0's for different in velocities
5. Attach elements that express the relative velocities
1. Recognize Elements
2. Use I functions for V's
3. Attach Elements
4. Use 0's for different in velocities
5. Attach elements that express the relative velocities.
1. Recognize Elements
2. Use I functions for v’s
3. Attach elements
4. Use 0’s for differences in velocities
5. Attach elements that express the relative velocities
1. Recognize elements
2. Use I functions for v's
3. Attach elements
4. Use 0's for differences in velocities
5. Attach elements that express the relative velocities
6. Conv
1. Recognize Elements
2. Use I functions for v's
3. Attach elements
4. Use 0's for different in velocities
5. Attach elements that express the relative velocities
6. Complete causal

\[ \text{not legible} \]
1. Recognize Elements
2. Use I functions for v’s
3. Attach elements
4. Use 0’s for different in velocities
5. Attach elements that express the relative velocities
6. Complete course marks (can)
1. Recognize Elements
2. Use I functions for v's
3. Attach elements
4. Use 0's for different in velocities
5. Attach elements that express the relative velocities
6. Complete causal marks (Campy m0)
1. Recognize Elements
2. Use I functions for V's
3. Attach elements
4. Use 0's for different in velocities
5. Attach elements that express the relative velocities
6. Complete causal marks (Campbell may do this automatically)
1. Recognize Elements
2. Use I functions for V's
3. Attach elements
4. Use 0's for different in velocities
5. Attach elements that express the relative velocities
6. Complete caused marks (camper may do this automatically)
1. Recognize Elements
2. Use I functions for V's
3. Attach elements
4. Use 0's for different in velocities
5. Attach elements that expose the relative velocities
6. Complete causal marks (Causal may dothis automatically)
Recognize Elements
Use 1 functions for v's
Attach elements
Use 0's for different in velocities
Attach elements that express the relative velocities
6. Complete causal marks (Campy may do this automatically)
2. Use I junction
3. Attach elements
4. Use 0's for in velocity
5. Attach the expression relative
6. Complete a mark may do auto
I \leftarrow 1 < 0 \rightarrow 1 \quad \text{V cannabinum}

V_{\text{wall}} = 0 \quad \text{V cannabinum} \rightarrow V_{\text{wall}}

SF 1 \quad \text{Cannabinum} \quad 1 \quad R

loading from "session.bg", done
previous session loaded (from "session") and placed
can't locate "from" element for bond!
new element "SF"
previous session loaded (from "session") and placed
can’t locate "from" element for bond!
new element "SF"
new element "1"
new element "C"
bond from 1.3 to C (bond 4)
new element "R"
bond from 1.3, 4 to R (bond 5)
I - 1 < V_{p fem} < 0  

V_{wall} = 0  

C < 1 < V_{can} - V_{wall}  

new element "R"  

bond from 1.3.4 to R (bond 5)  

new element "1"  

bond from 1
\[ I - 1 - O - 1 \]

\[ V_{wall} = 0 \]

\[ V_{can} - V_{wall} \]

new element "R"

bond from 1.3.4 to R (bond 5)

new element "1"

bond from 1 to 0.2.3 (bond 6)
I - 1 < 0 - 1
\[ \text{perm} \]
\[ V_{\text{wall}} = 0 \]
\[ V_{\text{can}} - V_{\text{wall}} \]

new element "1"
bond from 0.7.9 to 1 (bond 11)
new element "C"
bond from 1.11 to C (bond 12)
$$I - 1 < 0 \rightarrow 1$$

$$V_{\text{wall}} = 0$$

$$V_{\text{can}} = V_{\text{wall}}$$

$$V_{\text{can}}$$

$$V_{\text{wall}}$$

$$C$$

$$R$$

NEW ELEMENT "C"

bond from 1_11 to C (bond 12)

NEW ELEMENT "R"

bond from 1_11_12 to R (bond 13)
The CAMP-G/MATLAB model files have been successfully generated and written to files in the working directory.

Modify the files named below to complete model
Use the MATLAB Editor, Windows Notepad or other editor
- campgmod.m Model Parameter Initialization/Plot control
- campgOeq.m Model Differential Equations. Non-linear simulation
- campgym.m Symbolic State Space Model Matrices, Transfer Functions
- campgnum.m Numerical State Space Model, Transfer Functions, Matrices

... These will open automatically when MATLAB starts ...
You can also open them independently by entering “campgmatlab” “cag2mat” at the MATLAB prompt or by clicking the MATLAB icon on the CAMPG icons group.
PRESS ANY KEY TO CONTINUE ...
This is a Classroom License for instructional use only. Research and commercial use is prohibited.
% CAMP-G/MATLAB - Numeric State Space Model
% campgnum.m  CAMP-G/MATLAB function ............
% Generates: Numeric A,B,C,D matrices
% Numeric Transfer Functions
% Uses: Frequency Response using Transfer Functions
% Rlocus using numeric Transfer Functions.
% MATLAB Control system tool box operations
% using numeric transfer functions.

clear
more off

......... Initial conditions ............
Q12IN = ?; Q4IN = ?;
P10IN = ?; P0IN = ?;
initial = [Q12IN; Q4IN; P10IN; P0IN];
......... System Physical Parameters............
global C4 R5 IE I10 C12 R13
Define system physical parameters and system matrices numeric values
C4 = ?; R5 = ?
C12 = ?; R13 = ?
global SF1
fprintf ('\n Inputs Vector ')
fprintf ('\n u=[SF1 ] '
fprintf ('\n Input # 1 is SF1 '

 campgmod.m  x  campgequ.m  x  campgym.m  x  campgnum.m  x
CAMP/G-MATLAB - Numeric State Space Model

campgnum.m  CAMP/G-MATLAB function ............

Generates: Numeric A,B,C,D matrices
Numeric Transfer Functions

Uses: Frequency Response using Transfer Functions
Riccati using numeric Transfer Functions.
MATLAB Control system tool box operations
using numeric transfer functions.

clear
more off

........ Initial conditions ........

Q12IN= ? ; Q4IN= ? ;
P10IN= ? ; P5IN= ? ;
initial = [Q12IN; Q4IN; P10IN; P5IN] ;

........ System Physical Parameters ........

global C4 R5 I8 I10 C12 R13
Define system physical parameters and system matrices numeric values

C4 = ? ; R5 = ? ;
I8 = ? ; I10 = ? ;
C12 = ? ; R13 = ? ;

global SF1
fprintf ('\n Inputs Vector ') 
fprintf (' \n u=[ SF1 ] ')
fprintf (' \n Input # 1 is SF1 ')
campgnum.m  campgequ.m  campgcm.m  campgnum.m  campgnum.m

Click and drag to move campgequ.m or its button...
% CAMP-G/MATLAB  - Numeric State Space Model

...camplenum.m  CAMP-G/MATLAB function ...........

% Generates: Numeric A, B, C, D matrices
% Numeric Transfer Functions

% Uses:  Frequency Response using Transfer Functions
% Rlocus using numeric Transfer Functions.
% MATLAB Control system tool box operations
% using numeric transfer functions.

clear
more off

% Initial conditions ..........
Q12IN = ?; Q4IN = ?;
P10IN = ?; P8IN = ?;
initial = [Q12IN; Q4IN; P10IN; P8IN];

% System Physical Parameters.......

% Define system physical parameters and system matrices numeric values

% External inputs x(t), y(t) ........

fprintf('

Inputs Vector

u=[ SFI ]

Input # 1 is SFI')
Save A
The quick brown fox jumps over the lazy dog. 1234567890
%% CAMP-G/MATLAB - Numeric State Space Model
%% ............campgnum.m  CAMP-G/MATLAB func
%%
%% Generates: Numeric A,B,C,D matrices
%% Numeric Transfer Functions
%%
%% Uses:  Frequency Response using Transfer Function
%% Rlocus using numeric Transfer Function
%% MATLAB Control system tool box using numeric transfer function

%% clear
%% more off

%% Initial conditions ........
%% Q12IN= ? ; Q4IN= ? ;
% poly a, b, c, d, e

-campgsym.m Symbolic State Space Model
-campgnum.m Numeric State Space Model
% CAMP-G/MATLAB - Numeric State Space Model

% ............campaingnum.m CAMP-G/MATLAB func

% Generates: Numeric A,B,C,D matrices
% Numeric Transfer Functions

% Uses: Frequency Response using Transfer
% Rlocus using numeric Transfer Func
% MATLAB Control system tool box
% using numeric transfer functions

% clear
% more off
% .... Initial conditions ........
% Q12IN = ? ; Q4IN = ? ;
% ...

-campgsym.m Symbolic State Space Model
-campgnum.m Numeric State Space Model
%% CAMPG/MATLAB - GENERATED MODEL DESCRIPTION

The following files have been generated:

- campgmod.m  => m file containing model initial conditions, source
- campgequ.m  => m function containing 1st order differential
- campgsym.m  => m file containing symbolic state space
- campgnum.m  => m file containing numeric state space

For simulation and control, edit these files. Enter values for physical parameters, initial conditions, inputs and time controls.
% .... Initial conditions .......
Q12IN = ? ; Q4IN = ? ;
P10IN = ? ; P8IN = ? ;

initial = [Q12IN; Q4IN; P10IN; P8IN];

% .... System Physical Parameters ........
global C4 R5 I8 I10 C12 R13
C4 = ? ; R5 = ? ; I8 = ? ; I10 = ? ; C12 = ? ; R13 = ? ;

% .... External inputs se(t), sf(t) ......
global SF1
SF1 = ? ;

% .... Simulation Time Control ......
t0 = ? ; % Initial Time
% CAMPGMD.M - MATLAB MODEL INPUT FILE

clear

more on

% Initial conditions ........
Q12IN = ?; Q4IN = ?
P10IN = ?; P8IN = ?
initial = [Q12IN; Q4IN; P10IN; P8IN];

% System Physical Parameters........

global C4 R5 I8 I10 C12 R13

C4 = ?; R5 = ?
I8 = ?; I10 = ?
C12 = ?; R13 = ?

% External inputs se(t), sf(t) ........

global SF1

-- 3/15/2012 10:03 AM

Select a file to view details

fx >>
clear

%...... Initial conditions .........
I
Q12IN= ?; Q4IN= ?
P10IN= ?; P8IN= ?
initial = [Q12IN; Q4IN; P10IN; P8IN];

%...... System Physical Parameters .........
global C4 R5 I8 I10 C12 R13

C4 = ?; R5 = ?
I8 = ?; I10 = ?
C12 = ?; R13 = ?

%...... External inputs se(t), sf(t) .........
global SF1
clear
more on

% ........ Initial conditions ........
Q12IN = ?; Q4IN = ?;
P10IN = ?; P8IN = ?;

initial = [Q12IN; Q4IN; P10IN; P8IN];

% ........ System Physical Parameters........
global C4 R5 I8 I10 C12 R13
C4 = ?; R5 = ?;
I8 = ?; I10 = ?;
C12 = ?; R13 = ?;

global SF1

% ........ External inputs se(t), sf(t) ........

3/15/2012 10:05 AM
%......CAMPGMOD.M - MATLAB MODEL INPUT FILE

clear
more on

%...... Initial conditions ........
Q12IN=?; Q4IN=?;
P10IN=?; P8IN=?;
initial = [Q12IN; Q4IN; P10IN; P8IN];

%...... System Physical Parameters ........
global C4 R5 I8 I10 C12 R13
C4=??; R5=??;
I8=??; I10=??;
C12=??; R13=??;

%...... External inputs se(t), sf(t) ........
global SF1

%...... State Space Model ........

campgmod.m Symbolic State Space Model
-campgsym.m Numeric State Space Model
-campgnum.m Numeric State Space Model
initial = [Q12IN; Q4IN; P10IN; P8IN] ;

% .... System Physical Parameters ....

global C4 R5 I8 I10 C12 R13

C4 = ? ; R5 = ? ;
I8 = ? ; I10 = ? ;
C12 = ? ; R13 = ? ;

% .... External inputs se(t), sf(t) ....

global SF1

SF1 = ? ;

% .... Simulation Time Control ....

t0 = ? ; % Initial Time
tfinal = ? ; % Final Time
tspan = [t0 tfinal];

% .... Define Outputs ....

-campgsym.m Symbolic State Space Model
-campgnum.m Numeric State Space Model

fx >>
% External inputs se(t), sf(t) ......

global SF1

SF1 = ?

% Simulation Time Control ......
t0=1; % Initial Time
tfinal= ? ; % Final Time
tspan = [t0 tfinal];

% Define Outputs ......

global TIME STEP EFFORTS FLOWS

STEP = 1;

% Computer Simulation ......

fx >>
27 - C4 = ? ; R5 = ? ;
28 - I8 = ? ; I10 = ? ;
29 - C12 = ? ; R13 = ? ;
30 - % ...... External inputs se(t), sf(t) ......
31 - global SF1
32 - SF1 = ? ;
33 - %..... Simulation Time Control ......
34 - t0= ? ; % Initial Time
35 - tfinal= ? ; % Final Time
36 - tspan= [t0 tfinal];
37 - %..... Define Outputs ......
38 - global TIME STEP EFFORTS FLOWS
39 - STEP=1;
40 - % ...... Computer Simulation ......

-fx >>
% The "campgequ.m" function contains the system equations in state variable form.

% It returns the vector [t,p-q] where:
% t = time and p-q = vector of state variables.
% [t,p_q] is a column vector with rows [t, p_q].
% % t = [t, p_q] = ode45('campgequ',tspan,initial);
% Q12 = p_q(1); % Q4 = p_q(2)
% P10 = p_q(3); % P8 = p_q(4)
% p_q = [Q12; Q4; P10; P8];

% Sample Matlab structure for plotting simulation

figure(1)
subplot (211), plot(t,p_q(:,1),'b'), grid
42 \% The "campgequ.m" function contains the sys
43 \% equations in state variable form.
44
45 \% It returns the vector \([t, p-q]\) where:
46 \% \(t\) = time and \(p-q\) = vector of state variab
47 \% \([t, p_q]\) is a column vector with rows \([t,]
48 \% \color{blue}{[t, p_q]} = \text{ode45}'campgequ',tspan,initial);\)
49 \% Q12 = p_q(1); \% Q4 = p_q(2
50 \% P10 = p_q(3); \% P8 = p_q(4
51 \% p_q = [Q12; Q4; P10; P8];
52
53 \% Sample Matlab structure for plotting simula
54 \% \text{figure}(1)
55 \% \text{subplot} (211),\text{plot}(t, p_q(:,1),'b'),\text{grid}
The "campgequ.m" function contains the system equations in state variable form.

It returns the vector \([t,p\_q]\) where:
\[ t = \text{time and } p\_q = \text{vector of state variables} \]

\[ [t,p\_q] = \text{ode45('campgequ',tspan,initial);} \]

\[ \text{Q12} = p\_q(1); \quad \text{Q4} = p\_q(2) \]
\[ \quad \text{P10} = p\_q(3); \quad \text{P8} = p\_q(4) \]

\[ p\_q = [\text{Q12}; \text{Q4}; \text{P10}; \text{P8}] \]

Sample Matlab structure for plotting simulation:

```matlab
figure(1)
subplot (211), plot(t,p\_q(:,1), 'b'), grid
```
The "campgequ.m" function contains the system equations in state variable form.

It returns the vector \([t, p\_q]\) where:
\[ t = \text{time} \quad \text{and} \quad p\_q = \text{vector of state variables}. \]

\[ [t, p\_q] = \text{ode45}('campgequ', tspan, initial); \]

\[ Q12 = p\_q(1); \quad Q4 = p\_q(2); \]
\[ P10 = p\_q(3); \quad P8 = p\_q(4); \]
\[ p\_q = [Q12; Q4; P10; P8]; \]

Sample Matlab structure for plotting simulation:

```matlab
figure(1)
subplot (211), plot(t, p\_q(:,1), 'b'), grid
```

```
f mex >>
```
% p_q = [Q12; Q4; P10; P8] ;

% Sample Matlab structure for plotting simulation results
figure(1)
subplot (211), plot(t, p_q(:,1), 'b'), grid
title(' Variable p_q(:,1) (stored in column 1), color bl
ylabel ('p_q(1) (units)'), xlabel('Time (seconds)')
subplot (212), plot(t, p_q(:,2), 'm'), grid
title(' variable p_q(:,2) (stored in column 2), color m
ylabel ('p_q(2) (units)'), xlabel('Time (seconds)')

% Sample structure for plotting Output Variables as defined
% Example: If the efforts and flows were defined as:
% EFFORTS(STEP,:) = [e1 e11 e4];
initial = [Q12IN; Q4IN; P10IN; P8IN] ;

% .... System Physical Parameters ....

global C4 R5 I8 I10 C12 R13

C4 = ? ; R5 = ? ;
I8 = ? ; I10 = ? ;
C12 = ? ; R13 = ? ;

% .... External inputs se(t), sf(t) ....

global SF1

SF1 = ? ;

% .... Simulation Time Control ....

t0= ? ; % Initial Time

tfina1= ? ; % Final Time

tspan= [t0 tfinal];

% .... Define Outputs ....

campgsym.m Symbolic State Space Model

campgnnum.m Numeric State Space Model

fx >>
initial = [Q12IN; Q4IN; P10IN; P8IN];

% ...... System Physical Parameters ........

global C4 R5 I8 I10 C12 R13

C4 = ? ; R5 = ? ;
I8 = ? ; I10 = ? ;
C12 = ? ; R13 = ? ;

% ...... External inputs se(t), sf(t) ........

global SF1

SF1 = ? ;

% ...... Simulation Time Control ........

t0= ? ; % Initial Time
tfinal= ? ; % Final Time
tspan= [t0 tfinal];

% ...... Define Outputs ........

-campgsym.m Symbolic State Space Model
-campgnum.m Numeric State Space Model

fx >>
function p_qdot = campgequ(t,p_q)
%
% ............campgequ.m  CAMP-G/MATLAB function ........
%
% System differential equations, state Vectors
%
global C4 R5 I8 I10 C12 R13
global SF1
global TIME STEP EFFORTS FLOWS
%
% System Differential Equations-First Order Form
%
%...... Define State Variables .......

Q12 = p_q(1) ;  Q4  = p_q(2) ;
P10 = p_q(3) ;  P8  = p_q(4) ;

% p_q = [Q12; Q4; P10; P8] ;

%...... Define derivatives (dp,dq) and output variables (e

f1 = SF1 ;
f2 = f1 ;
f7=f8 ;
e12=Q12/C12 ;
f9=f10 ;
f12=f11 ;
dQ12=f12 ;
f4=f3  
;
e13=f13*R13 ;
dQ4=f4  ;
e5=f5*R5 ;
e11=e12+e13 ;
e9=e11 ;
dP10=e10 ;
e6=e3  ;
e7=e11 ;
e8=-e6-e7 ;
e2=e3  ;
e2=e3 ;
% Define derivatives (dp, dq) and output variables (e)

\[ f_1 = S_{F1} \]
\[ f_2 = f_1 \]
\[ e_4 = \frac{Q_4}{C_4} \]
\[ f_7 = f_8 \]
\[ f_8 = P_8 / I_8 \]
\[ e_{12} = \frac{Q_{12}}{C_{12}} \]
\[ f_6 = f_8 \]
\[ f_9 = f_{10} \]
\[ f_{10} = P_{10} / I_{10} \]
\[ f_{11} = f_{7} - f_9 \]
\[ f_{12} = f_{11} \]
\[ e_{13} = f_{13} \times R_{13} \]
\[ dQ_4 = f_4 \]
\[ f_4 = f_3 \]
\[ f_5 = f_3 \]
\[ e_5 = f_5 \times R_5 \]
\[ e_{11} = e_{12} + e_{13} \]
\[ e_9 = e_{11} \]
\[ e_{10} = e_9 \]
\[ dP_{10} = e_{10} \]
\[ e_6 = e_3 \]
\[ e_7 = e_{11} \]
% Define derivatives (dp, dq) and output variables (e)

f1 = Sf1;
f2 = f1;
e4 = Q4/C4;
f7 = f8;
f8 = P8/I8;
e12 = Q12/C12;
f6 = f8;
f9 = f10;
f10 = P10/I10;
f12 = f11;
ed = f12;

\textcolor{blue}{dQ12 = f12;}

f4 = f3;
f5 = f3;
e13 = f13 * R13;
dQ4 = f4;
e5 = f5 * R5;
e9 = e11;
e11 = e12 + e13;
e10 = e9;
edp10 = e10;
e3 = e4 + e5;
e6 = e3;
e7 = e11;

% Define derivatives (dp, dq) and output variables (e)

f1 = SF1;

f2 = f1;

e4 = Q4 / C4;

f8 = P8 / I8;

e12 = Q12 / C12;

f10 = P10 / I10;

f6 = f8;

f11 = f7 - f9;

f13 = f11;

dQ12 = f12;

f3 = - f2 + f6;

f4 = f3;

f5 = f3;

e13 = f13 * R13;

f1 = f13;

dQ4 = f4;

e5 = f5 * R5;

e11 = e12 + e13;

e9 = e11;

e2 = e11;

dP10 = e10;

e6 = e3;

e7 = e11;

e3 = e4 + e5;

3/15/2012 10:04 AM
f12 = f11 ;

20 -
dQ12 = f12 ;
f4 = f3 ;
22 -
e13 = f13 * R13 ;
23 -
e5 = f5 * R5 ;
24 -
e9 = e11 ;
25 -
edP10 = e10 ;
26 -
e6 = e3 ;
27 -
e8 = e6 - e7 ;
e2 = e3 ;
28 -
f13 = f11 ;
f3 = - f2 + f6 ;
f5 = f3 ;
edQ4 = f4 ;

e11 = e12 + e13 ;
e10 = e9 ;
e3 = e4 + e5 ;
e7 = e11 ;
dP8 = e8 ;
e1 = e2 ;

% ... Build vector of derivatives p_qdot(n) ...
% p_qdot1) = dQ12 ;
% p_qdot2) = dQ4 ;
% p_qdot3) = dP10 ;
% p_qdot4) = dP8 ;

-campgsym.m Symbolic State Space Model
-campgnum.m Numeric State Space Model
f12 = f11

\begin{align*}
19 & \quad f13 = f11 \\
20 & \quad dQ12 = f12 \\
21 & \quad f4 = f3 \\
22 & \quad e13 = f13 \times R13 \\
23 & \quad e5 = f5 \times R5 \\
24 & \quad e9 = e11 \\
25 & \quad dP10 = e10 \\
26 & \quad e6 = e3 \\
27 & \quad e8 = e6 - e7 \\
28 & \quad e2 = e3
\end{align*}

% ... Build vector of derivatives \texttt{p\_qdot}(n) ...

% \texttt{p\_qdot1}) = dQ12 \\
% \texttt{p\_qdot2}) = dQ4 \\
% \texttt{p\_qdot3}) = dP10 \\
% \texttt{p\_qdot4}) = dP8

\texttt{-- 3/15/2012 10:54 AM}

- \texttt{campgsym.m} Symbolic State Space Model
- \texttt{campgnum.m} Numeric State Space Model

fx >>
24  initial = [Q12IN; Q4IN; P10IN; P8IN] ;
25  % ...... System Physical Parameters ........
26  global C4 R5 I8 I10 C12 R13
27  C4 = ? ;  R5 = ? ;
28  I8 = ? ;  I10 = ? ;
29  C12 = ? ;  R13 = ? ;
30  % ...... External inputs se(t), sf(t) ......
31  global SF1
32  SF1 = ? ;
33  % ...... Simulation Time Control ......
34  t0= ? ;  % Initial Time
35  tfinal= ? ;  % Final Time
36  tspan= [t0 tfinal];
37  % ...... Define Outputs ......
38  campgmod.m
39  campgeq.m
40  campgsym.m
41  campgnum.m
f12 = f11;
dQ12 = f12;
f4 = f3;
e13 = f13 * R13;
e5 = f5 * R5;
e9 = e11;
e10 = e9;
dP10 = e10;
e6 = e3;
e7 = e11;
e8 = e6 - e7;
e2 = e3;

f13 = f11;
f3 = -f2 + f6;
f5 = f3;
edQ4 = f4;
e11 = e12 + e13;
e10 = e9;
e3 = e4 + e5;
e7 = e11;
e1 = e2;

dP8 = e8;

% ... Build vector of derivatives p_qdot(n)...
% p_qdot1) = dQ12 ; % p_qdot2) = dQ4 ;
% p_qdot3) = dP10 ; % p_qdot4) = dP8 ;
% ... Generation of A, B matrices corresponding to states:

% dQ12=P8/I8-P10/I10
% ... Number of States 4
  A(1,:) = [sb, sb, sb, sb];
  A(1,:) = [0, 0, -1/I10, 1/I8];
  B(1,:) = [sb];
  D(1,:) = [sb];
  B(1,:) = [0];
% dQ4=-SF1+P8/I8
  A(2,:) = [0, 0, 0, +1/I8];
  B(2,:) = [-1];
% dP10=Q12/C12+P8/I8*R13-P10/I10*R13
  A(3,:) = [1/C12, 0, -1/I10*R13, +1/I8*R13];
% ... Generation of A, B matrices corresponding to states:

% dQ12=P8/I8-P10/I10
% ... Number of States 4

A(1,:) = [sb, sb, sb, sb];
A(1,:) = [0, 0, -1/I10, 1/I8];
B(1,:) = [sb];
D(1,:) = [sb];
B(1,:) = [0];

% dQ4=-SF1+P8/I8
A(2,:) = [0, 0, 0, +1/I8];
B(2,:) = [-1];

% dP10=Q12/C12+P8/I8*R13-P10/I10*R13
A(3,:) = [1/C12, 0, -1/I10*R13, +1/I8*R13];
B(3,:) = [0];

-campgsym.m Symbolic State Space Model
-campgnum.m Numeric State Space Model
% ... Generation of A, B matrices corresponding to states:

% dQ12=P8/I8-P10/I10

% ... Number of States 4

A(1,:) = [sb, sb, sb, sb];
A(1,:) = [0, 0, -1/I10, 1/I8];
B(1,:) = [sb];
D(1,:) = [sb];
B(1,:) = [0];

% dQ4=-SF1+P8/I8

A(2,:) = [0, 0, 0, +1/I8];
B(2,:) = [-1];

% dP10=Q12+C12+P8/I8*R13-P10/I10*R13

A(3,:) = [1/C12, 0, -1/I10*R13, +1/I8*R13];
% Number of States 4
A(1,:) = [sb, sb, sb, sb];
A(1,:) = [0, 0, -1/I10, 1/I8];
B(1,:) = [sb];
D(1,:) = [sb];
B(1,:) = [0];

% dQ4=-SF1+P8/I8
A(2,:) = [0, 0, 0, +1/I8];
B(2,:) = [-1];

% dP10=Q12/C12+P8/I8*R13-P10/I10*R13
A(3,:) = [1/C12, 0, -1/I10*R13, +1/I8*R13];
B(3,:) = [0];

A(4,:) = [-1/C12, -1/C4, +1/I10*R13, -1/I8*R5-1/I8*R13];

- campgsym.m Symbolic State Space Model
- campgnum.m Numeric State Space Model
B(1,:) = [sb];
D(1,:) = [sb];
B(1,:) = [0];
dQ4=-SF1+P8/I8
A(2,:) = [0,0,0,+1/I8];
B(2,:) = [-1];
dP10=Q12/C12+P8/I8*R13-P10/I10*R13
A(3,:) = [1/C12,0,-1/I10*R13,+1/I8*R13];
B(3,:) = [0];
A(4,:) = [-1/C12,-1/C4,+1/I10*R13,-1/I8*R5-1/I8*R13];
B(4,:) = [+1*R5];

Generate C and D matrices corresponding to outputs e's and

- campgsym.m Symbolic State Space Model
- campgnum.m Numeric State Space Model

fx >>
46 - \( B(1,:) = [s, b] \);
47 - \( D(1,:) = [s, b] \);
48 - \( B(1,:) = [0] \);
49 - \( \text{dQ4} = -S F1 + P8 / I8 \)
50 - \( A(2,:) = [0, 0, 0, 0 + 1 / I8] \);
51 - \( B(2,:) = [-1] \);
52 - \( \text{dP10} = Q12 / C12 + P8 / I8 * R13 - P10 / I10 * R13 \)
53 - \( A(3,:) = [1 / C12, 0, -1 / I10 * R13, +1 / I8 * R13] \);
54 - \( B(3,:) = [0] \);
55 - \( \text{dP8} = -Q4 / C4 + S F1 * R5 - P8 / I8 * R5 - Q12 / C12 - P8 / I8 * R13 + P10 / I10 * R13 \)
57 - \( B(4,:) = [+1 * R5] \);
58 - Generate C and D matrices corresponding to outputs e's and

-campgsym.m Symbolic State Space Model
-campgnum.m Numeric State Space Model
46 - \( B(1,:) = [sb]; \)
47 - \( D(1,:) = [sb]; \)
48 - \( B(1,:) = [0]; \)
49 - \( dQ4=-SF1+P8/I8 \)
50 - \( A(2,:) = [0,0,0,+1/I8]; \)
51 - \( B(2,:) = [-1]; \)
52 - \( dP10=Q12/C12+P8/I8*R13-P10/I10*R13 \)
53 - \( A(3,:) = [1/C12,0,-1/I10*R13,+1/I8*R13]; \)
54 - \( B(3,:) = [0]; \)
55 - \( dP8= Q4/C4+SF1*R5-P8/I8*R5-Q12/C12-P8/I8*R13+P10/I10*R13 \)
56 - \( A(4,:) = [-1/C12,-1/C4,+1/I10*R13,-1/I8*R5-1/I8*R13 ]; \)
57 - \( B(4,:) = [+1*R5 ]; \)
58
59 - Generate C and D matrices corresponding to outputs e's and
B(1,:) = [sb];

D(1,:) = [sb];

B(1,:) = [0];

dQ4 = -SF1 + P8/I8

A(2,:) = [0,0,0, +1/I8];
B(2,:) = [-1];

dP10 = Q12/C12 + P8/I8*R13 - P10/I10*R13

A(3,:) = [1/C12, 0, -1/I10*R13, +1/I8*R13];
B(3,:) = [0];


A(4,:) = [-1/C12, -1/C4, +1/I10*R13, -1/I8*R5 - 1/I8*R13];
B(4,:) = [+1*R5];

Generate C and D matrices corresponding to outputs e's and
```matlab
46 - B(1,:) = [sb];
47 - D(1,:) = [sb];
48 - B(1,:) = [0];
49 - dQ4=-SF1+P8/I8
50 - A(2,:) = [0,0,0,+1/I8];
51 - B(2,:) = [-1];
52 - dP10=Q12/C12+P8/I8*R13-P10/I10*R13
53 - A(3,:) = [1/C12,0,-1/I10*R13,+1/I8*R13];
54 - B(3,:) = [0];
56 - A(4,:) = [-1/C12,-1/C4,+1/I10*R13,-1/I8*R5-1/I8*R13];
57 - B(4,:) = [+1*R5];

Generate C and D matrices corresponding to outputs e's and
```

```matlab
-campgsym.m Symbolic State Space Model
-campgnum.m Numeric State Space Model
```
B(1,:) = [sb];
D(1,:) = [sb];
B(1,:) = [0];
dQ4 = -SP1P0/50;
A(2,:) = B(2,:);
dP10 = Q1;
A(3,:) = B(3,:);
dP8 = -Q4;
A(4,:) = B(4,:);
loading from "session.bg", done
previous session loaded (from "session") and placed
B(1,:), B(2,:), A(3,:), B(3,:), dQ4=-SF, dP10=Q1, dP8=-Q4, A(4,:), B(4,:), Generate

loading from "session.bg", done
previous session loaded (from "session") and placed
loading from "session.bg", done
previous session loaded (from "session") and placed

52 - dP10=Q12/C12+P8/I8*R13-P10/I10*R1;
53 - A(3,:)= [1/C12, 0, -1/I10*R13, +1/I6;
54 - B(3,:)= [0];
<table>
<thead>
<tr>
<th>Line</th>
<th>MATLAB Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>47</td>
<td><code>D(1,:) = [sb];</code></td>
</tr>
<tr>
<td>48</td>
<td><code>B(1,:) = [0];</code></td>
</tr>
<tr>
<td>49</td>
<td><code>% dQ4=-SF1+P8/I8</code></td>
</tr>
<tr>
<td>50</td>
<td><code>A(2,:) = [0,0,0,+1/I8];</code></td>
</tr>
<tr>
<td>51</td>
<td><code>B(2,:) = [-1];</code></td>
</tr>
<tr>
<td>52</td>
<td><code>% dP10=Q12/C12+P8/I8*R13-P10/I10*R</code></td>
</tr>
<tr>
<td>53</td>
<td><code>A(3,:) = [1/C12,0,-1/I10*R13,+1/</code></td>
</tr>
<tr>
<td>54</td>
<td><code>B(3,:) = [0];</code></td>
</tr>
<tr>
<td>55</td>
<td><code>% dP8=-Q4/C4+SF1*R5-P8/I8*R5-Q12/C</code></td>
</tr>
</tbody>
</table>
loading from "session.bg", done
prevements session loaded (from "session") and placed

B(1,:) = [sb];
D(1,:) = [s];
B(1,:) = [0];

% dQ4=-SF1+P8/I8
A(2,:) = [0,0,0,1/I8];
B(2,:) = [-1];
% dP10=Q12/C12+P8/I8*R13-P10/I10*R
A(3,:) = [1/C12,0,-1/I10*R13,1];
B(3,:) = [0];
% dP8=-Q4/C4+SF1*R5-P8/I8*R5-Q12/C.
initial = [Q12IN; Q4IN; P10IN; ...]

% System Physical Parameters

global C4 R5 I8 I10 C12 R13

c4 = ? ; r5 = ? ;
i8 = ? ; i10 = ? ;
c12 = ? ; r13 = ? ;

% External inputs se(t), sf

global SF1

sf1 = ? ;
loading from "session.bg", done
previous session loaded (from "session") and placed

```
initial = [Q12IN; Q4IN; P10IN; E
% .... System Physical Parameters
  global C4 R5 I8 I10 C12 R13
  C4 = ? ; R5 = ? ;
  I8 = ? ; I10 = ? ;
  C12 = ? ; R13 = ? ;
% .... External inputs se(t), sf
  global SF1
  SF1 = ? ;
```
CAMP-G - System Simulation

initial = [Q12IN; Q4IN; P10IN; P8IN]

% ....... System Physical Parameters...

global C4 R5 I8 I10 C12 R13

C4 = ?  ;  R5 = ?  ;
I8 = ?  ;  I10 = ?  ;
C12 = ?  ;  R13 = ?  ;

% ....... External inputs se(t), sf(t)

global SF1

SF1 = ?

% ....... Simulation Time Control ......

loading from "session.bg", done
previous session loaded (from "session") and placed
initial = [Q12IN; Q4IN; P10IN; P8IN]

% .... System Physical Parameters...

global C4 R5 I8 I10 C12 R13

C4 = ?; R5 = ?

I8 = ?; I10 = ?

C12 = ?; R13 = ?

% .... External inputs se(t), sf(t)

global SF1

SF1 = ?

% .... Simulation Time Control ....
initial = [Q12IN; Q4IN; P10IN; P8IN]
% .... System Physical Parameters...
global C4 R5 I8 I10 C12 R13
C4 = ?; R5 = ?; I8 = ?; I10 = ?; C12 = ?; R13 = ?;
% .... External inputs se(t), sf(t)
global SF1
SF1 = ?;
% .... Simulation Time Control ......
loading from "session.bg", done
previous session loaded (from "session") and placed
initial = [Q12IN; Q4IN; P10IN; P8IN]

% .... System Physical Parameters ...

% global C4 R5 I8 I10 C12 R13
C4 = ? ; R5 = ?
I8 = ? ; I10 = ?
C12 = ? ; R13 = ?

% ... External inputs se(t), sf(t)
global SF1
SF1 = ?

% ... Simulation Time Control ....
loading from "session.bg", done
previous session loaded {from "session"} and placed

initial = [Q12IN; Q4IN; P10IN; P8IN]

% ...... System Physical Parameters...

global C4 R5 I8 I10 C12 R13

C4 = ? ; R5 = ? ;
I8 = ? ; I10 = ? ;
C12 = ? ; R13 = ? ;

% ...... External inputs se(t), sf(t)

global SF1

SF1 = ? ;
loading from "session.bg", done
previously session loaded (from "session") and placed
initial = [Q12IN; Q4IN; P10IN; P8IN]

% .... System Physical Parameters...

global C4 R5 I8 I10 C12 R13

C4 = ? ; R5 = ? ;
I8 = ? ; I10 = ? ;
C12 = ? ; R13 = ? ;

% .... External inputs se(t), sf(t)

global SF1

SF1 = ? ;

% .... Simulation Time Control ....

loading from "session.bg", done
previous session loaded (from "session") and placed
loading from "session.bg", done
previsions session loaded (from "session") and placed
```
initial = [Q12IN; Q4IN; P10IN; P8IN]

% .... System Physical Parameters...

global C4 R5 I8 I10 C12 R13
C4 = ?; R5 = ?;
I8 = ?; I10 = ?;
C12 = ?; R13 = ?;

% .... External inputs se(t), sf(t)

global SF1
SF1 = ?;

% .... Simulation Time Control ....
```
initial = [Q12IN; Q4IN; P10IN; P8IN]

% ...... System Physical Parameters...

global C4 R5 I8 I10 C12 R13

C4 = ? ; R5 = ? ;
I8 = ? ; I10 = ? ;
C12 = ? ; R13 = ? ;

% ...... External inputs se(t), sf(t)

global SF1

SF1 = ? ;

% ...... Simulation Time Control ......
initial = [Q12IN; Q4IN; P10IN; P8IN]

% .... System Physical Parameters...

global C4 R5 I8 I10 C12 R13

C4 = ? ; R5 = ? ;
I8 = ? ; I10 = ? ;
C12 = ? ; R13 = ? ;

% .... External inputs se(t), sf(t)

global SF1

SF1 = ? ;

% .... Simulation Time Control ....

loading from "session.bg", done
previons session loaded (from "session") and placed
```matlab
initial = [Q12IN; Q4IN; P10IN; P8IN];

% .... System Physical Parameters...

global C4 R5 I8 I10 C12 R13
C4 = ?; R5 = ?;
I8 = ?; I10 = ?;
C12 = ?; R13 = ?;

% .... External inputs se(t), sf(t)

global SF1
SF1 = ?;

% .... Simulation Time Control ....
```
initial = [Q12IN; Q4IN; P10IN; P8IN]

% .... System Physical Parameters...

global C4 R5 I8 I10 C12 R13

C4 = ? ; R5 = ? ;
I8 = ? ; I10 = ? ;
C12 = ? ; R13 = ? ;

% .... External inputs se(t), sf(t)

global SF1

SF1 = ? ;

% .... Simulation Time Control ......

loading from "session.bg", done
previously session loaded (from "session") and placed
initial = \{Q12IN; Q4IN; P10IN; P8IN\}

% .... System Physical Parameters...

global C4 R5 I8 I10 C12 R13

C4 = ?; R5 = ?; I8 = ?; I10 = ?; C12 = ?; R13 = ?;

% .... External inputs se(t), sf(t)

global SF1

SF1 = ?;

% .... Simulation Time Control ....
loading from "session.bg", done
previous session loaded (from "session") and placed

initial = [Q12IN; Q4IN; P10IN; P8IN]

% .... System Physical Parameters...

global C4 R5 I8 I10 C12 R13

C4 = ? ; R5 = ? ;
I8 = ? ; I10 = ? ;
C12 = ? ; R13 = ? ;

% .... External inputs se(t), sf(t)

global SF1

SF1 = ? ;

% .... Simulation Time Control ....
initial = [Q12IN; Q4IN; P10IN; P8IN]

%......System Physical Parameters...

global C4 R5 I8 I10 C12 R13

C4 = ?; R5 = ?; I8 = ?; I10 = ?; C12 = ?; R13 = ?

%......External inputs se(t), sf(t)

global SF1

SF1 = ?

%......Simulation Time Control......
loading from "session.bg", done
previouse session loaded (from "session") and placed

initial = [Q11IN; Q4IN; P10IN; P8IN]

% .... System Physical Parameters ...

global C4 R5 I8 I10 C12 R13

C4 = ? ; R5 = ? ; I8 = ? ; I10 = ? ; C12 = ? ; R13 = ? ;

% .... External inputs se(t), sf(t)

global SF1

SF1 = ? ;

% .... Simulation Time Control ....
initial = [Q12IN; Q4IN; P10IN; P8IN]

% .... System Physical Parameters...

global C4 R5 I8 I10 C12 R13

C4 = ?; R5 = ?;
I8 = ?; I10 = ?;
C12 = ?; R13 = ?;

% .... External inputs se(t), sf(t)

global SF1

SF1 = ?;

% .... Simulation Time Control . . . . .
initial = [Q12IN; Q4IN; P10IN; P8IN]

% .... System Physical Parameters...

global C4 R5 I8 I10 C12 R13

C4 = ? ; R5 = ? ;
I8 = ? ; I10 = ? ;
C12 = ? ; R13 = ? ;

% .... External inputs se(t), sf(t)

global SF1

SF1 = ? ;

% .... Simulation Time Control ....
initial = [Q12IN; Q4IN; P10IN; P8IN]

global C4 R5 I8 I10 C12 R13
C4 = ? ; R5 = ? ;
I8 = ? ; I10 = ? ;
C12 = ? ; R13 = ? ;

global SF1
SF1 = ? ;

% .... External inputs se(t), sf(t)
% .... Simulation Time Control ....

loading from "session.bg", done
previous session loaded (from "session") and placed
initial = [Q12IN; Q4IN; P10IN; P8IN]

% .... System Physical Parameters...

global C4 R5 I8 I10 C12 R13

C4 = ?; R5 = ?
I8 = ?; I10 = ?
C12 = ?; R13 = ?

% .... External inputs se(t), sf(t)

global SF1

SF1 = ?

% .... Simulation Time Control ....

loading from "session.bg", done
previously session loaded (from "session") and placed
It returns the vector \([t, p-q]\) where \(t\) = time and \(p-q\) = vector of state.

\([t, p_q]\) is a column vector with rows:

\[\begin{bmatrix}
Q_{12} \\
P_{10}
\end{bmatrix}\]

Sample Matlab structure for plotting:

```matlab
figure(1)
```
It returns the vector \([t, p-q]\) where \(t\) = time and \(p-q\) = vector of state. \([t, p_q]\) is a column vector with rows
\[
[t, p_q] = \text{ode45('campgequ', tspan, ini),}
\]
\(Q12 = p_q(1)\)
\(Q4 = p_q(4)\)
\(P10 = p_q(3)\)
\(p_q = [Q12; Q4; P10; P8]\)

Sample Matlab structure for plotting

```
figure(1)
```
45  \% It returns the vector \[t, p-q\] where
46  \% \( t \) = time and \( p-q \) = vector of state
47  \% \[t, p_q\] is a column vector with ro
48 -  \[t, p_q\] = ode45('campgequ', tspan, ini
49  \% \( Q12 = p_q(1) \);
50  \% \( P10 = p_q(3) \);
51  \% \( p_q = [Q12, Q4, P10; P8] \);
52  \%
53  \% Sample Matlab structure for plotting
54  figure(1)
```matlab
51  p_q = [Q12; Q4; P10; P8];
52  
53  % Sample Matlab structure for plotting
54  figure(1)
55  subplot (211), plot(t,p_q(:,1)', 'b'), g
56  title('Variable p_q(:,1) (stored in
57  ylabel ('p_q(1) (units)'), xlabel('Ti
58  subplot (212), plot(t,p_q(:,2)', 'm'), g
59  title('variable p_q(:,2) (stored in
60  ylabel ('p_q(2) (units)'), xlabel('Ti
```
```matlab
p_q = [Q12; Q4; P10; P8];

% Sample Matlab structure for plotting
figure(1)

subplot(211), plot(t, p_q(:,1), 'b'), grid

title('Variable p_q(:,1) (stored in yLabel)' 'p_q(1) (units)'), xlabel('Time')

subplot(212), plot(t, p_q(:,2), 'm'), grid

title('Variable p_q(:,2) (stored in yLabel)' 'p_q(2) (units)'), xlabel('Time')
```

loading from "session.bg", done
previous session loaded (from "session") and placed
51  \( p_q = [Q12; Q4; P10; P8] \); 
52  
53  \% Sample Matlab structure for plotting
54  figure(1)
55  subplot (211), plot(t, p_q(:,1), 'b'), g
56  title(' Variable p_q(:,1) (stored in
57  ylabel ('p_q(1) (units)'), xlabel('Ti
58  subplot (212), plot(t, p_q(:,2), 'm'), g
59  title(' variable p_q(:,2) (stored in
60  ylabel ('p_q(2) (units)'), xlabel('Ti
```matlab
54 - figure(1)
55 - subplot (211), plot(t,p_q(:,1), 'b'), g
56 - title('Variable p_q(:,1) (stored in
57 - ylabel ('p_q(1) (units)'), xlabel('Ti
58 - subplot (212), plot(t,p_q(:,2), 'm'), g
59 - title('variable p_q(:,2) (stored in
60 - ylabel ('p_q(2) (units)'), xlabel('Ti
61 - % Sample structure for plotting Output
62 - % Example: If the efforts and flows
```
```matlab
54 - figure(1)
subplot (211), plot(t, p_q(:, 1), 'b'),
title(' Variable p_q(:, 1) (stored in
ylabel ('p_q(1) (units)'), xlabel('T:
subplot (212), plot(t, p_q(:, 2), 'm'),
title(' Variable p_q(:, 2) (stored in
ylabel ('p_q(2) (units)'), xlabel('T:
%
% Sample structure for plotting Output:

loading from "session.bg", done
previous session loaded (from "session") and placed
```
66 - \text{tre}(2)
67 - \text{tel} \text{ vs TIME (First column of "EFFORTS")}
68 - \text{plot} (211), \text{plot} (\text{TIME}, \text{EFFORTS}(::1), 'b')
69 - \text{le}('The scope of variable 'EFFORTS' is global. Changes to its value might span multiple workspaces.
70 - \text{f1} \text{ vs time (First column of "FLOWS")}
71 - \text{plot} (212), \text{plot} (\text{TIME}, \text{FLOWS}(::1), 'm')
72 - \text{le}('Flow variable of vector "FLOWS" :1
73 - \% .............BOND GRAPH NOTATION
74 - \% GENERALIZED VARIABLES BOND GRAPH
75 - \%
e2=e3 ;

% ... Build vector of derivatives p_qdot
%    p_qdot1) = dQ12 ;   %   p_
%    p_qdot3) = dP10 ;   %   p_
% Derivatives vector
p_qdot = [dQ12 ; dQ4 ; dP10 ; dP8] ;

% ... Define output vectors (efforts,:
% Sample Structure. Add any effort, fl
STEP = STEP + 1;

% In "campgmod.m" the following struct;
% Sample structure for plotting Output
% Example: If the efforts and flows
% EFFORTS(STEP,:) = [e5 e11 e4];
% FLOWS(STEP,:) = [f2 f9 f8];
% figure(2)
% Plot e11 vs TIME (Second column of
% subplot (211), plot (TIME, EFFORTS(}
FLOWS(STEP,:) = [f1]; % Define flow
STEP=STEP+1;

% In "campgmod.m" the following struct:
% Sample structure for plotting Output
% Example: If the efforts and flows
EFFORTS(STEP,:) = [e125 e11 e4];
FLOWS(STEP,:) = [f2 f9 f8];

% Plot e11 vs TIME (Second column of
loading from "session.bg", done
previous session loaded (from "session") and placed
BOND: FROM
FLOWS(STEP,:) = [f1]; % Define flows
STEP = STEP + 1;

% In "campgmod.m" the following structure:
% Sample structure for plotting Output
% Example: If the efforts and flows
EFFORTS(STEP,:) = [e1 e21 e4];
FLOWS(STEP,:) = [f2 f9 f8];
figure(2)

% Plot e11 vs TIME (Second column of
% Define flows
FLOWS(STEP,: ) = [f1];

STEP=STEP+1;

% In "campgmod.m" the following struct:
% Sample structure for plotting Output
% Example: If the efforts and flows
EFFORTS(STEP,:) = [e124];
FLOWS(STEP,:) = [f2 f9 f8];

figure(2)

% Plot e11 vs TIME (Second column of
loading from "session.bg", done
previous session loaded {from "session"} and placed
BOND: FROM
FLOWS(STEP,:) = [f1]; % Define flows
STEP = STEP + 1;

% In "campgmod.m" the following struct:
% Sample structure for plotting Output:
% Example: If the efforts and flows
EFFORTS(STEP,:) = [e12];
FLOWS(STEP,:) = [f2 f9 f8];
figure(2)

% Plot e11 vs TIME (Second column of
loading from "session.bg", done
previous session loaded (from "session") and placed
42 - FLOWS(STEP,:) = [f1]; % Define flows
43 - STEP=STEP+1;
44 - % In "campgmod.m" the following structure:
45 - % Sample structure for plotting Output
46 - % Example: If the efforts and flows
47 - EFFORTS(STEP,:) = [e12];
48 - FLOWS(STEP,:) = [f2 f9 f8];
49 - figure(2)
50 - % Plot e11 vs TIME (Second column of
```
42 - FLOWS(STEP,:) = [f1];  % Define flows
43 - STEP=STEP+1;
44
45  % In "campgmod.m" the following structure:
46  % Sample structure for plotting Output
47  % Example: If the efforts and flows
48  EFFORTS(STEP,:) = [e12];
49  FLOWS(STEP,:) = [f2 f9 f8];
50  figure(2)
51  % Plot e11 vs TIME (Second column of
```


```
42 - FLOWS(STEP,:) = [f1]; % Define flow
43 - STEP=STEP+1;
44 -
45 - In "campgmod.m" the following struct:
46 - Sample structure for plotting Output
47 - Example: If the efforts and flows
48 - EFFORTS(STEP,:) = [e12];
49 - FLOWS(STEP,:) = [f2 f9 f8];
50 - figure(2)
51 - % Plot e11 vs TIME (Second column of
```

loading from "session.bg", done
previous session loaded {from "session"} and placed
FLOWS(STEP,:) = [f1]; % Define flow
STEP = STEP + 1;
% In "campgmod.m" the following struct:
% Sample structure for plotting Output
% Example: If the efforts and flows
EFFORTS(STEP,:) = [e12];
FLOWS(STEP,:) = [f2 f9 f8];
figure(2);
% Plot e11 vs TIME (Second column of
42 - FLOWS(STEP,:) = [f1]; % Define flow
43 - STEP = STEP + 1;

% In "campgmod.m" the following struct:
% Sample structure for plotting Output
% Example: If the efforts and flows
45 - EFFORTS(STEP,:) = [e12];
46 - FLOWS(STEP,:) = [f2 f9 f8];
47 - figure(2)
48 - % Plot e11 vs TIME (Second column of

loading from "session.bg", done
previous session loaded [from "session"] and placed

BOND: FROM
initial = [Q12IN; Q4IN; P10IN; P8IN]

% ....System Physical Parameters...

global C4 R5 I8 I10 C12 R13

C4 = ? ; R5 = ? ;
I8 = ? ; I10 = ? ;
C12 = ? ; R13 = ? ;

% ....External inputs se(t), sf(t)

global SF1

SF1 = ? ;

% ....Simulation Time Control ....
initial = [Q12IN; Q4IN; P10IN; P8IN];

... System Physical Parameters ........

.obal C4 R5 I8 I10 C12 R13

C4 = ? ; R5 = ? ;
I8 = ? ; I10 = ? ;
C12 = ? ; R13 = ? ;

... External inputs se(t), sf(t) ....

.obal SF1

SF1 = ? ;

.. Simulation Time Control ....

loading from "session.bg", done
previous session loaded (from "session") and placed

BOND: FROM
CAMPGMOD.M - MATLAB MODEL INPUT FILE

...ear

... Initial conditions ........
Q12IN= ? ; Q4IN= ? ;
P10IN= ? ; P8IN= ? ;

initial = [Q12IN; Q4IN; P10IN; P8IN] ;

...System Physical Parameters........
.global C4 R5 I8 I10 C12 R13

C4 = ? ; R5 = ? ;
...CAMPGMOD.M - MATLAB MODEL INPUT FILE

ear

> on

.... Initial conditions ........

Q12IN = 0; Q4IN = ?;

P10IN = ?; P8IN = ?

initial = [Q12IN; Q4IN; P10IN; P8IN];

....System Physical Parameters........

.global C4 R5 I8 I10 C12 R13

C4 = ?; R5 = ?;
...CAMPGMOD.M - MATLAB MODEL INPUT FILE:

19
20
21
22
23
24
25
26
27

... Initial conditions ........
Q12IN = 0; Q4IN = ?;
P10IN = ?; P8IN = ?;

initial = [Q12IN; Q4IN; P10IN; P8IN];

... System Physical Parameters....

cbval C4 R5 I8 I10 C12 R13

C4 = ?; R5 = ?;
```matlab
18    ... CAMPGMOD.M - MATLAB MODEL INPUT FILE
19    .ear
20    pre on
21    .... Initial conditions ......
22    Q12IN = 0 ; Q4IN = 1
23    P10IN = ? ; P8IN = ?
24    initial = [Q12IN; Q4IN; P10IN; P8IN] ;
25    .... System Physical Parameters........
26    .obal C4 R5 I8 I10 C12 R13
27    C4 = ? ; R5 = ?
```
... CAMPGMOD.M - MATLAB MODEL INPUT FILE ...

ear

... Initial conditions ...........
Q12IN = 0; Q4IN = 0;
P10IN = ?; P8IN = ?;

initial = [Q12IN; Q4IN; P10IN; P8IN];

... System Physical Parameters ........
bal C4 R5 I8 I10 C12 R13

C4 = ?; R5 = ?;
CAMPGMOD.M - MATLAB MODEL INPUT FILE

ear
pre on

... Initial conditions ........
Q12IN = 0; Q4IN = 0;
P10IN = P; P8IN = ?;
initial = [Q12IN; Q4IN; P10IN; P8IN];

... System Physical Parameters ........
.global C4 R5 I8 I10 C12 R13

C4 = ?; R5 = ?;
CAMPGMOD.M - MATLAB MODEL INPUT FILE

.ear
.on

.... Initial conditions ........
Q12IN = 0 ; Q4IN = 0 ;
P10IN = 100 ; P8IN = ? ;
initial = [Q12IN ; Q4IN ; P10IN ; P8IN] ;

....System Physical Parameters........
.global C4 R5 I8 I10 C12 R13

C4 = ? ; R5 = ? ;
18  ...CAMPGMOD.M - MATLAB MODEL INPUT FILE
19
20  .ear
21  ...on
22  .... Initial conditions ........
23  Q12IN = 0; Q4IN = 0;
24  P10IN = 100; P8IN = ?;
25  initial = [Q12IN; Q4IN; P10IN; P8IN] ;
26  .... System Physical Parameters ........
27  global C4 R5 I8 I10 C12 R13
28  C4 = ?; R5 = ?;
CAMPGMOD.M - MATLAB MODEL INPUT FILE

18 ...CAMPGMOD.M - MATLAB MODEL INPUT FILE
19 .ear
20 re on
21 .... Initial conditions ........
22 Q12IN = 0 ; Q4IN = 0 ;
23 P10IN = 100 ; P8IN = 1000 ;
24 initial = [Q12IN; Q4IN; P10IN; P8IN] ;
25 .... System Physical Parameters ........
26 .obal C4 R5 I8 I10 C12 R13
27 C4 = ? ; R5 = ? ;

loading from "session.bg", done
previous session loaded (from "session") and placed
24 - initial = [Q12IN; Q4IN; P10IN; P8IN];
25 - ....System Physical Parameters..........
26 - .obal C4 R5 I8 I10 C12 R13
27 - C4 = ? ; R5 = ? ;
28 - I8 = ? ; I10 = ? ;
29 - C12 = ? ; R13 = ? ;
30 - .... External inputs se(t), sf(t) ....
31 - .obal SF1
32 - SF1 = ? ;
33 - .. Simulation Time Control .....
... Initial conditions .......

Q12IN = 0 ; Q4IN = 0 ;
P10IN = 100 ; P8IN = 1000 ;

initial = [Q12IN; Q4IN; P10IN; P8IN];

... System Physical Parameters ...........

obal C4 R5 I8 I10 C12 R13

C4 = I R5 = ? I
I8 = ? I I10 = ? I
C12 = ? I R13 = ? I

... External inputs se(t), sf(t) .......

loading from "session.bg", done
previous session loaded (from "session") and placed
21  .... Initial conditions .......
22  Q12IN= 0 ; Q4IN= 0 ;
23  P10IN= 100 ; P8IN= 1000 ;
24  initial = [Q12IN; Q4IN; P10IN; P8IN] ;
25  .... System Physical Parameters .......
26  global C4 R5 I8 I10 C12 R13
27  C4 = 1/40 ; R5 = ? ;
28  I8 = ? ; I10 = ? ;
29  C12 = ? ; R13 = ?;
30  .... External inputs se(t), sf(t) .....
.... Initial conditions ........
22 - Q12IN = 0 ; Q4IN = 0 ;
23 - P10IN = 100 ; P8IN = 1000 ;
24 - initial = [Q12IN; Q4IN; P10IN; P8IN] ;
25 - .... System Physical Parameters ........
26 - global C4 R5 I8 I10 C12 R13
27 - C4 = 1/40 ; R5 = ? ;
28 - I8 = ? ; I10 = ? ;
29 - C12 = 1/60 ; R13 = ? ;
30 - .... External inputs se(t), sf(t) ........
.... Initial conditions ........
Q12IN = 0 ; Q4IN = 0 ;
P10IN = 100 ; P8IN = 1000 ;
initial = [Q12IN; Q4IN; P10IN; P8IN] ;
.... System Physical Parameters ........
local C4 R5 I8 I10 C12 R13
C4 = 1/40 ; R5 = ? ;
I8 = 1000 ; I10 = ? ;
C12 = 1/60 ; R13 = ? ;
.... External inputs se(t), sf(t) ........
.... Initial conditions ......
Q12IN = 0 ; Q4IN = 0 ;
P10IN = 100 ; P8IN = 1000 ;
initial = [Q12IN; Q4IN; P10IN; P8IN] ;
.... System Physical Parameters ......

. obal C4 R5 I8 I10 C12 R13
C4 = 1/40 ; | R5 = ? ;
I8 = 1000 ; | I10 = ? ;
C12 = 1/60 ; | R13 = ? ;

.... External inputs se(t), sf(t) ......

loading from "session.bg", done
previous session loaded (from "session") and placed
.... Initial conditions ........
22-
23-
24-
25-
26-
27-
28-
29-
30-

.... System Physical Parameters ........

... Global C4 R5 I8 I10 C12 R13

C4 = 1/40 ; | R5 = ? |
I8 = 1500 ; | I10 = ? |
C12 = 1/60 ; | R13 = ? |

.... External inputs se(t), sf(t) ........

loading from "session.bg", done
previous session loaded {from "session"} and placed
21. .... Initial conditions ........
22. Q12IN = 0 ; Q4IN = 0 ;
23. P10IN = 100 ; P8IN = 1000 ;
24. initial = [Q12IN; Q4IN; P10IN; P8IN];
25. ....System Physical Parameters ........
26. Global C4 R5 I8 I10 C12 R13
27. C4 = 1/40 ; R5 = ?;
28. I8 = 1500 ; I10 = 100 ;
29. C12 = 1/60 ; R13 = ?;
30. .... External inputs se(t), sf(t) .......
.... Initial conditions ........
Q12IN = 0 ; Q4IN = 0 ;
P10IN = 100 ; P8IN = 1000 ;

initial = [Q12IN; Q4IN; P10IN; P8IN];

.... System Physical Parameters ........

global C4 R5 I8 I10 C12 R13

c4 = 1/40 ; r5 = 20 ;
i8 = 1500 ; i10 = 100 ;
c12 = 1/60 ; r13 = ? ;

.... External inputs se(t), sf(t) ........
21. .... Initial conditions ........
22. - Q12IN = 0 ; Q4IN = 0 ;
23. - P10IN = 100 ; P8IN = 1000 ;
24. - initial = [Q12IN; Q4IN; P10IN; P8IN];
25. .... System Physical Parameters........
26. - global C4 R5 I8 I10 C12 R13
27. - C4 = 1/40 ; R5 = 20 ;
28. - I8 = 1500 ; I10 = 100 ;
29. - C12 = 1/60 ; R13 = 50 ;
30. .... External inputs se(t), sf(t) .......
C4 = 1/40 ; R5 = 20 ;
I8 = 1500 ; I10 = 100 ;
C12 = 1/60 ; R13 = 50 ;

.... External inputs se(t), sf(t) ....

.global SF1

SF1 = ?

Simulation Time Control ..... 
t0= ? ; % Initial Time
tfinal= ? ; % Final Time
tspan= [t0 tfinal];

loading from "session.bg", done
previous session loaded [from "session"] and placed
C4 = 1/40 ; R5 = 20 ;
I8 = 1500 ; I10 = 100 ;
C12 = 1/60 ; R13 = 50 ;

... External inputs se(t), sf(t) ....

.global SF1

SF1 = 0

... Simulation Time Control ....
t0= ? ; % Initial Time
tfinal= ? ; % Final Time
tspan= [t0 tfinal];
.... External inputs se(t), sf(t) ....

.. Define Outputs ..... 

... Define Outputs ..... 

.. Simulation Time Control ..... 

t0 = ? ; % Initial Time

tfinal = ? ; % Final Time

tspan = [t0 tfinal];

.. Define Outputs ..... 

... Define Outputs ..... 

EP = 1;
.... External inputs se(t), sf(t) ....

EMALE SF1

SF1 = 0;

.. Simulation Time Control ....

t0 = 0;  % Initial Time

tfinal = ?;  % Final Time

tspan = [t0 tfinal];

.. Define Outputs ....

EMALE TIME STEP EFFORTS FLOWS

EP = 1;

loading from "session.bg", done
previous session loaded (from "session") and placed
.... External inputs se(t), sf(t) ....
31
32
33
34
35
36
37
38
39
30     .... External inputs se(t), sf(t) ....
31     .obal SF1
32     SF1 = 0 ;
33     .. Simulation Time Control ..... 
34     t0= 0 ;  % Initial Time
35     tfinal= 5 ;  % Final Time
36     tspan= [t0 tfinal];
37     .. Define Outputs ..... 
38     .obal TIME STEP EFFORTS FLOWS
39     'EP=1;
The "campgequ.m" function contains equations in state variable form.

It returns the vector \([t,p-q]\) where:

- \(t\) = time and \(p-q\) = vector of state variables

\([t,p_q] = \text{ode45}('campgequ',tspan,in:)\)

\(Q12 = \text{p}(1)\); \(Q4 = \text{p}(3)\); \(P10 = \text{p}(6)\); \(P8\) = \([Q12; Q4; P10; P8]\);
The function contains a state variable form.

The vector \([t, p-q]\) where \(p-q = \text{vector of state}\) column vector with \(r\):

```matlab
load ('campgequ', 'tspan', 'inp')
p_q(1) ; % Q
q (3) ; % P
Q
P1
P8] ;
```
Variable $p_{d1}$ (stored in column 1), color blue.

Variable $p_{d2}$ (stored in column 2), color magenta.

Loading from "session.bg", done previous session loaded (from "session") and placed.
The "campgequ.m" function contains equations in state variable form.

Figure 1

Variable $p_{q1}$ (stored in column 1), color blue

Variable $p_{q2}$ (stored in column 2), color magenta

Figure 2

BOND: FROM
The "campgequ.m" function contains equations in state variable form.
The "campgequ.m" function contains equations in state variable form.
The "campgequ.m" function contains equations in state variable form.
The "campgequ.m" function contains equations in state variable form.