MATLAB Assignment

Due Date: November 30, 2010 @ 11:59 PM

November 24, 2010

<u>How to turn in your homework</u>: Place all your files in a directory called "lastName-Date"; zip that directory and email to TA.

Problem 1. Implement MATLAB code that opens a file, called "model.adm" and parses the text below in order to generate all the information required to fully characterize a force $\mathbf{F}^P$ acting on a rigid body at point P. Note that anything that follows a "%" on a line in the file must be considered a comment and as such ignored during parsing. In other words, the rest of a line following a "%" character is there only for the purpose of clarifying a construct and can be ignored when parsed.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Force: 3                  % id of this force
Body: 2                   % id of the body the force acts on
Fx: 3*t+1                 % x-component of the Force, might depend on time
Fy: sin(0.4*t)            % y-component of the Force, might depend on time
IsInGlobal: 0             % indicates whether the force F is provided in global RF
xPprime: 4.1              % x of point P where F acts, expressed in LRF
yPprime: 10.3             % y of point P where F acts, expressed in LRF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Note: The attribute IsInGlobal indicates whether the force is expressed in the global or local reference frame (G-RF or L-RF). A zero value for this flag indicates that the expression of the force is in L-RF, in other words,

$$\bar{\mathbf{F}}^P = \begin{bmatrix} 3t+1 \\ \sin(0.4t) \end{bmatrix}$$

Then, the force that you would be working with is $\mathbf{F}^P = \mathbf{A}_2\bar{\mathbf{F}}^P$; i.e., you have to express the force in the G-RF. If the force is already provided in the G-RF; i.e., if IsInGlobal=1, you don't have to do anything to the expression of the force prior to using it to generate the associated generalized force.

Problem 2. Implement MATLAB code that opens a file, called “model.adm” and parses the text below in order to generate all the information required to fully characterize a torque n acting on a rigid body. Note that anything that follows a “%” on a line in the file must be considered a comment and as such ignored during parsing. In other words, the rest of a line following a “%” character is there only for the purpose of clarifying a construct and can be ignored when parsed.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Torque: 4            % id of this torque
Body: 5              % id of the body the torque acts on
n: 13*t+1            % value assume by torque, might depend on time
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Problem 3. Implement MATLAB code that opens a file, called “model.adm” and parses the text below in order to generate all the information required to fully characterize a Translational-Spring-Damper-Actuator (TSDA) element acting between P_i on body i and P_j on body j. Note that anything that follows a “%” on a line in the file must be considered a comment and as such ignored during parsing. In other words, the rest of a line following a “%” character is there only for the purpose of clarifying a construct and can be ignored when parsed.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
TSDA: 12                        % id of this TSDA element
BodyI: 3                        % id of body I
BodyJ: 5                        % id of body J
xPprimeI: 0.4                   % x of point P on body I
yPprimeI: -0.3                  % y of point P on body I
xPprimeJ: 0.4                   % x of point P on body J
yPprimeJ: -0.3                  % y of point P on body J
kTSDA: 300                      % value of the spring stiffness
cTSDA: 12                       % value of the damping coefficient
hTSDA: sin(10*t)                % expression for the actuation force
lZero: 0.1                      % spring length at zero tension (a constant)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Note: We will assume that the actuator can only provide a function that depends on time, like $\sin(10t)$ in this example.

Problem 4. This problem is similar to the previous problem. The difference is that the TSDA acts between P_i on body i and a point P on ground.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
TSDA: 23                        % id of this TSDA element
```

```
BodyI: 3                        % id of body I
xPprimeI: 0.4                   % x of point P on body I
yPprimeI: -0.3                  % y of point P on body I
xPground: 1.3                   % x of point P on ground
yPground: -2.6                  % y of point P on ground
kTSDA: 600                      % value of the spring stiffness
cTSDA: 32                       % value of the damping coefficient
hTSDA: cos(10*t)                % expression for the actuation force
lZero: 0.4                      % spring length at zero tension (a constant)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Note: We will assume that the actuator can only provide a function that depends on time, like $\cos(10t)$ in this example.

Problem 5. Assume that an applied force is defined by the attributes specified in the snippet of adm file shown in **Problem 1**. Implement MATLAB code that based on the information available after parsing the adm file will provide at an arbitrary time t the value of the generalized force $\mathbf{Q}^A \in \mathbb{R}^3$ using state information $\mathbf{q}(t)$ and $\dot{\mathbf{q}}(t)$.

Problem 6. Assume that an applied torque is defined by the attributes specified in the snippet of adm file shown in **Problem 2**. Implement MATLAB code that based on the information available after parsing the adm file will provide at an arbitrary time t the value of the generalized force $\mathbf{Q}^A \in \mathbb{R}^3$ using state information $\mathbf{q}(t)$ and $\dot{\mathbf{q}}(t)$.

Problem 7. Assume that a TSDA element is defined by the attributes specified in the snippet of adm file shown in **Problem 3**. Implement MATLAB code that based on the information available after parsing the adm file will provide at an arbitrary time t the value of the generalized force $\mathbf{Q}^A \in \mathbb{R}^3$ using state information $\mathbf{q}(t)$ and $\dot{\mathbf{q}}(t)$.

Problem 8. Assume that a TSDA element is defined by the attributes specified in the snippet of adm file shown in **Problem 4**. Implement MATLAB code that based on the information available after parsing the adm file will provide at an arbitrary time t the value of the generalized force $\mathbf{Q}^A \in \mathbb{R}^3$ using state information $\mathbf{q}(t)$ and $\dot{\mathbf{q}}(t)$.