MATLAB Assignment

Due Date: November 9, 2010

October 26, 2010

When working on this assignment you might want to take a look at MATLAB code that was put together by students who took ME751 last Spring. The format for defining an input file there was a little different, yet the parsing approach can be used as a starting point when dealing with this assignment. A couple of places where you could look:

```
http://sbel.wisc.edu/Courses/ME751/2010/SimulationEngine/HW12/unicorn/unicorn_HW12/project/getModel.m
http://sbel.wisc.edu/Courses/ME751/2010/SimulationEngine/HW10/Shaq/loadModel.m
```

MATLAB source code generated by students for ME751 can be found here:

```
http://sbel.wisc.edu/Courses/ME751/2010/SimulationEngine/index.htm
```

Turning in your homework: place all your files in a directory called "lastNameDate"; zip that directory and email to TA.

Problem 1. Implement MATLAB code that opens a file, called "input.acf" (acf stands for analysis control file) and parses the text below in order to generate all the information required to define a simulation. Note that there is no space between ":" and a keyword such as "simulation", for instance.

```
simulation: kinematics
tend: 20.0
stepSize: 0.01
outputSteps: 400
output: position, all
output: velocity, all
output: acceleration, 2
```

The acf file above indicates that your simEngine2D MATLAB code is expected to run a kinematics analysis, with step size of 0.01 seconds, end the analysis after 20 seconds and report information at 400 intermediate station points. At each of these intermediate time milestones the following information

needs to be output: the position of all bodies in the model, the velocities of all bodies in the model, the acceleration of body with id 2.

Problem 2. Implement MATLAB code that opens a file, called "model.adm" (adm comes for "analysis data for the model") and parses the text below in order to generate all the information required to fully characterize a rigid body in the context of 2D Kinematics and Dynamics analysis. 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.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Body: 7                             % id of this body
Mass: 3                             % mass of body
Jbar: 0.4                           % mass moment of inertia
xZero: 0.23                         % initial X position
yZero: -0.3                         % initial Y position
phiZero: 1.5707963267948            % initial orientation (this is pi/2)
xDotZero: 1.0                       % initial velX
yDotZero: 0.0                       % initial velY
phiDotZero: 0.1                     % initial velPhi
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Problem 3. Same as problem 2, yet this time implement a MATLAB routine to generate all the information required to fully characterize an absolute X constraint in the context of 2D Kinematics and Dynamics analysis. The absolute X is defined in the adm file as follows (LPR stand for Local Reference Frame, that is the reference frame attached to the body):

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
AbsoluteX: 3                      % id of this constraint
Body: 3                           % id of participating body
xPprime: 0.4                      % x of point P on moving body, expressed in LRF
yPprime: -0.3                     % y of point P on moving body, expressed in LRF
xPground: 4                       % x of point P on ground
yPground: 2.3                     % y of point P on ground
CmotionFunction: 0.1*t+1/9        % provides expression for C(t)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Problem 4. Same as problem 3, except that now you'll have to handle the absolute Y constraint. The absolute Y constraint is defined in the adm file as follows:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
AbsoluteY: 4                    % id of this constraint
Body: 1                         % id of participating body
xPprime: 4.1                    % x of point P on moving body, expressed in LRF
yPprime: 10.3                   % y of point P on moving body, expressed in LRF
xPground: 0                     % x of point P on ground
yPground: 0.3                   % y of point P on ground
CmotionFunction: t^2+3*t+1      % provides expression for C(t)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Problem 5. Same as problem 3, yet this time implement a MATLAB routine to generate all the information required to fully characterize a relative X constraint in the context of 2D Kinematics and Dynamics analysis.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RelativeX: 1                    % id of this constraint
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
CmotionFunction: 2.3            % provides expression for C(t)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Problem 6. Same as problem 5, yet this time parse a relative Y constraint defined in an adm file as follows:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RelativeY: 8                    % id of this constraint
BodyI: 2                        % id of body I
BodyJ: 1                        % id of body J
xPprimeI: 1.4                   % x of point P on body I
yPprimeI: 0.63                  % y of point P on body I
xPprimeJ: -.9                   % x of point P on body J
yPprimeJ: 12.3                  % y of point P on body J
CmotionFunction: sin(10*t)      % provides expression for C(t)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Problem 7. Same as problem 3, yet this time implement a MATLAB routine to generate all the information required to fully characterize a revolute joint in the context of 2D Kinematics and Dynamics analysis when the joint is specified like below. Note that there is a motion that can be prescribed on this joint. This is precisely the motion that we discussed in class and you can find it in the textbook at page 93, Fig. 3.5.7. For simplicity, take $\theta_i = \theta_j = 0$.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RevoluteJoint: 23               % id of this constraint
BodyI: 1                        % id of body I
BodyJ: 9                        % id of body J
xPprimeI: 1.4                   % x of point P on body I
yPprimeI: 11.1                  % y of point P on body I
xPprimeJ: -12                   % x of point P on body J
yPprimeJ: 12.3                  % y of point P on body J
CmotionFunction: sin(t + pi/3)  % provides expression for C(t)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Important Observation: Note that the "CmotionFunction" line is *optional*. If you have it there, it means that there is a motion prescribed in conjunction with this joint. If it's missing, it means that there is no motion on the revolute joint.

Problem 8. Same as problem 3, yet this time implement a similar set of MATLAB routines to generate all the information required to fully characterize a translational joint in the context of 2D Kinematics and Dynamics analysis when the joint is specified like below. Note that there is a motion that can be prescribed on this joint. This is precisely the translational motion that we discussed in class and you can find in the textbook at page 94, Fig. 3.5.9. For simplicity make sure you define your $\bar{\mathbf{v}}_i$ and $\bar{\mathbf{v}}_j$ vectors such that $||\bar{\mathbf{v}}_i|| = ||\bar{\mathbf{v}}_j|| = 1$. You might want to normalized them just in case, see example below, where $\bar{\mathbf{v}}_i = [1\ 1]^T$ is not a unit vector if defined as in the following amd file snippet:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
TranslationalJoint: 3     % id of this constraint
BodyI: 4                  % id of body I
BodyJ: 3                  % id of body J
xPprimeI: 0.6             % x of point P on body I
yPprimeI: -3.1            % y of point P on body I
xPprimeJ: 0.3             % x of point P on body J
yPprimeJ: -4.1            % y of point P on body J
vPrimeIx: 1               % x coordinate of the v_i vector expressed in LRF
vPrimeIy: 1               % y coordinate of the v_i vector expressed in LRF
vPrimeJx: 0               % x coordinate of the v_j vector expressed in LRF
vPrimeJy: 1               % y coordinate of the v_j vector expressed in LRF
CmotionFunction: 3*t      % provides expression for C(t)
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Important Observation: Note that the "CmotionFunction" line is *optional*. If you have it there, it means that there is a motion prescribed in conjunction with this joint. If it's missing, it means that there is no motion on the translational joint.

Problem 9. Implement a MATLAB routine to generate all the information required to fully characterize a absolute angle constraint in the context of 2D Kinematics and Dynamics analysis when the constraint is specified like below.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
AbsoluteAngle: 3                % id of this constraint
Body: 4                         % id of body the constraint refers to
CmotionFunction: 0.1*t+1/9      % provides expression for C(t)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Problem 10. Implement a MATLAB routine to generate all the information required to fully characterize a absolute distance constraint in the context of 2D Kinematics and Dynamics analysis when the constraint is specified in the adm file like below.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
AbsoluteDistance: 4             % id of this constraint
Body: 1                         % id of participating body
xPprime: 2.1                    % x of point P on moving body, expressed in LRF
yPprime: 1.3                    % y of point P on moving body, expressed in LRF
xPground: 0.65                  % x of point P on ground
yPground: -1.3                  % y of point P on ground
CmotionFunction: t^2+3*t+1      % provides expression for C(t)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Problem 11. Implement a MATLAB routine to generate all the information required to fully characterize a relative distance constraint in the context of 2D Kinematics and Dynamics analysis when the constraint is specified in the adm file like below.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
RelativeDistance: 1              % id of this constraint
BodyI: 1                         % id of body I
BodyJ: 4                         % id of body J
xPprimeI: 1.2                    % x of point P on body I
yPprimeI: -1.3                   % y of point P on body I
xPprimeJ: 4.1                    % x of point P on body J
```

```
yPprimeJ: 10.2                  % y of point P on body J
CmotionFunction: 2.3            % provides expression for C(t)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Problem 12. Problems 1 through 11 define a language that can be used to put together an adm file describing a mechanism. Using this modeling language, generate an adm file, called "myFrist-Model.adm" that is associated with the mechanism in Example 3.1.1, page 51. In that example, consider $f(t) = sin(2t)$. Your adm file should be a succession of modeling elements introduced in Problem 1 through 12 that together completely define the mechanism considered.