MATLAB Assignment 2

Due Date: September 29, 2011

September 23, 2011

When working on this assignment you might want to take a look at MATLAB code that was put together by students who took ME451 last Fall. The students back then did not come up with identical solutions. Take a look at their solutions (look under Assignment 3), and develop your own.

http://sbel.wisc.edu/Courses/ME451/2010/SimEngine2D/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
```

The acf file above indicates that your simEngine2D MATLAB code will be 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.

What do you need to do? What you need to do is write a MATLAB script that parses (reads in) the file and has the values 'kinematics' (a string), 20.0 (a number), 0.01 (another number), 400 (yet another number), stored in the MATLAB variables that have the names *simluation*, *tend*, *stepSize*, and *outputSteps*, respectively.

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)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

NOTE: When you write the code to parse the declarations above and you handle the *CmotionFunction* MATLAB variable keep in mind that this should be a function of time. To deal with it you'll have to recall what you did in the MATLAB assignment of last week to read in a function that you can later one call to evaluate for various values of time 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)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```