MATLAB Assignment 4

October 16, 2014

Due: October 23, 2014

When working on this assignment you might want to take a look at MATLAB code that was developed by students who took ME451 in previous years. The students back then did not come up with identical solutions. Take a look at their solutions and develop your own.
2010: http://sbel.wisc.edu/Courses/ME451/2010/SimEngine2D/index.htm
2011: http://sbel.wisc.edu/Courses/ME451/2011/SimEngine2D/index.htm

Turning in your assignment: place all your files in a directory called "lastName_Matlab_04", zip that directory, and upload the resulting file "lastName_Matlab_04.zip" in the appropriate Dropbox Folder at Learn@UW.

Problem 1. In a previous MATLAB Assignment, you implemented code to parse the following definition of an *AbsoluteX* constraint:

```
{
  "name": "absX",
  "id": 1,
  "type": "AbsoluteX",
  "body1": 1,
  "sP1": [-2, 0],
  "xGround": {\color{red} 4},
  "fun": "0.1*t + 1/9"
}
```

Based on the discussion we had during Lecture of October 16 regarding a unified treatment of absolute and relative constraints, slightly modify the definition of an *AbsoluteX* constraint to the following, more flexible, definition:

```
{
  "name": "absX",
  "id": 1,
  "type": "AbsoluteX",
  "body1": 1,
  "sP1": [-2, 0],
  "fun": "{\color{red} 4} + 0.1*t + 1/9"
}
```

Notes:

- IMPORTANT! With this change, `fun` can **never** be **"NONE"**: you always have to prescribe a function of time, which however can be a constant function.

Problem 2. Same as Problem 1, but for an *AbsoluteY* constraint: the new definition for the *AbsoluteY* constraint from MATLAB Assignment 2 should now read:

```
{
  "name": "absY",
  "id": 2,
  "type": "AbsoluteY",
  "body1": 1,
  "sP1": [-2, 0],
  "fun": "3.5"
}
```

Notes:

- The property `fun` is now 3.5 (i.e. a constant function, equal in value to what used to be `yGround`). As mentioned before, `fun` can **never** be `"NONE"`.

Problem 3. The previous assignments and the problems above collectively introduced what effectively is a **model definition language**. Using this language, you will generate a first model.

Consider the pendulum in Example 3.2.1 on page 60 of the textbook, whose total length is 2. Assume that its mass is 2.5 and its mass moment of inertia is 1 (all units S.I). Also assume that a motion is prescribed on the pendulum to the effect that its orientation should change like $\phi(t) = \pi/2 + 2\pi t$. The state of this model is to be characterized by an array of Cartesian generalized coordinates $\mathbf{q} = [x, y, \phi]^T$.

Generate a pair of files, simplePend.acf and simplePend.adm that you would use in conjunction with simEngine2D to run a Kinematics Analysis for 1 second, with a time step of 0.01 seconds, and output saved at 100 points.

Notes:

- Your ADM file should be a succession of modeling elements that have been introduced in this and the previous assignment that together completely define the mechanism considered. Use an *AbsoluteX* and an *AbsoluteY* constraints to model this mechanism (use the improved versions for *AbsoluteX* and *AbsoluteY* above.)

- For this assignment, you only have to generate the ADM file. In a future assignment you'll have to parse the ADM file and do something useful with the information you extract out of it.

Problem 4. Consider the Example 3.1.2 on page 54 of the textbook. For that mechanism, use $l = 2$ and impose a motion on the revolute joint between body 1 and ground that results in a constant angular velocity $\omega_1 = 6$ rad/sec. Note that at $t = 0$, $\phi_1 = 0$. Generate a pair of input files twoBody.acf and twoBody.adm that will be used in a future assignment by your simEngine2D to perform 10 seconds worth of Kinematics Analysis of the mechanism using a step size of $\Delta t = 0.001$ seconds. Specify in the ACF file that the output should be saved at 500 intermediate points.

Problem 5. Consider the four bar mechanism discussed in Problem 3.3.3 and also described in this Note. Use Approach 1 to generate a pair of input files fourbar1.acf and fourbar1.adm that will be used in a future assignment by your simEngine2D to perform 5 seconds worth of Kinematics Analysis of the mechanism using a step size of $\Delta t = 0.01$ seconds. Specify in the ACF file that the output should be saved at 100 intermediate points. Note that a joint motion is specified between bodies 1 and 4 to control the angle between the GRF and the LRF associated with body 1. The expression for this motion reads $\phi_1 - \pi/2 - 4t = 0$.

Same as above, but now use Approach 2 to generate a pair of input files fourbar2.acf and fourbar2.adm.