

ME751: Assignment 6

Problem 1 [**simEngine3D track**]. Implement MATLAB functions that provide all the computational kinematics quantities that are associated with the basic GCons Φ^{DP^2} and Φ^D . Specifically, the MATLAB functions should be able to return any or all of the following quantities: the value of the expression of the constraint, the right-hand side of the velocity equation ν , the right-hand side of the acceleration equation, the expression of the partial derivatives Φ_r and Φ_p . Since you don't need all these quantities all the time, you should devise a mechanism (using maybe flags) to instruct the subroutine what quantities are actually needed.

Observation 1: There are two ways to go about specifying the attributes of your GCon; i.e., who i is, who j is, where point P is, etc.:

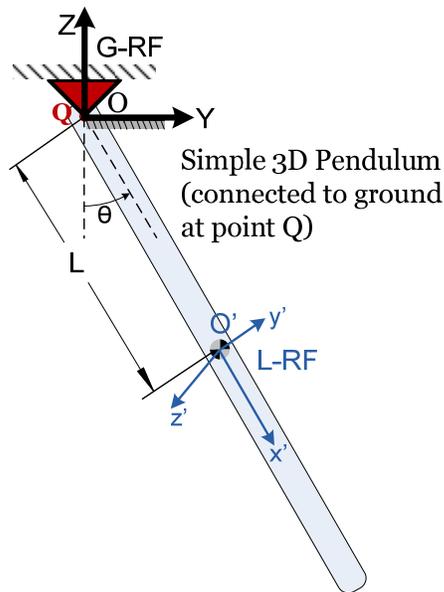
- a) Nice but hard: use an input file that contains information about the *attributes* of your model's GCons. You will also have to think about the format in which you expect the user to provide the required attributes. In conclusion, think about how you want the model attributes to be provided to you, and then generate a file with extension "me751.mdl" (from model), that actually stores the attributes. As an example, take a look at the ADAMS adm file that is uploaded on the class website. If you want to push it to an extreme, you can actually start using adm files as your model definition files. To put things in perspective, this mdl file is the input file, what the `simEngine3D` code picks up to generate the actual GCons that you have in your model.
- b) Easier: hardcode in MATLAB the attributes of the GCons you have in your mechanism. For each mechanism you will have to have a MATLAB file that sets up the model.

Observation 2: You will have to have another MATLAB file called "simEngine3D-A6P1.m", which when executed is expected to generate all the Kinematics quantities listed above by reading the attributes and data specified in a "me751.mdl" file that you also have to generate. You won't have an input file if you go with solution b) above.

Observation 3: Keep in mind that body j can be the ground. In this case, $j = 0$. This is relevant since the number of columns in the Jacobian is half – there are no partial derivatives with respect to \mathbf{r}_j or \mathbf{p} . You will have to be able to properly dimension the size of the vectors/matrices that you work with to account for the fact that one of the bodies in the GCon is ground and as such it does not bring generalized coordinates along.

Observation 4: Keep in mind that the RHS of the acceleration equation needs to be the one for the $\mathbf{r-p}$ formulation since this is what you implement. We discussed about how to derive $\hat{\gamma}$ in class.

NOTE: I will look into your GitHub repo and take "simEngine3D-A6P1.m" for a spin to gauge the quality of the work.



Problem 2 [simEngine3D track]. In an input file “revJoint.mdl”, define your first mechanism model that will have one body hanging as shown in the figure. The pendulum has $L = 2$, it is symmetric as shown in the figure, and it is attached to ground at its tip (point Q). The revolute joint on the ground is at point O of location $(0, 0, 0)$; that is, the origin of the G-RF. Note that the pendulum moves in the global OYZ plane; also note that the $O'x'y'$ plane of the L-RF is parallel to the OYZ plane of the G-RF. The body is subjected to a motion specified as $\theta(t) = \frac{\pi}{4} \cos(2t)$. One way to implement this motion is through a DPI basic geometric constraint.

For this homework, at $t = 0$, provide the following information: value of $\Phi^F(\mathbf{q}, t)$, Φ_q^F , ν , γ . All units used are SI. The program that will be executed for this

problem should be “simEngine3D-A6P2.m”.

NOTE: I will look into your GitHub repo and take “simEngine3D-A6P2.m” for a spin to gauge the quality of the work

Problem 3 [simEngine3D track]. This problem builds on previous problem. The schematic of the mechanism is shown in the figure above. The rigid body is subjected to a motion specified as

$$\theta(t) = \frac{\pi}{4} \cos(2t).$$

You will have to carry out a Kinematics Analysis for the mechanism for 10 seconds of its evolution. To this end, use a time grid with time steps of $\Delta t = 10^{-3}$. In the folder solution include six plots. The first three will display the location of point O' in the G-RF as a function of time, the second one its velocity in the G-RF as a function of time, and the third one will display its acceleration in the G-RF. The last set of three plots will display to same information for the tip of the pendulum located at Q . When you return the handwritten component of this homework please include an explanation of the results that you obtain for point Q . The program that will be executed for this problem should be “simEngine3D-A6P3.m”.

NOTE: I will look into your GitHub repo and take “simEngine3D-A6P3.m” for a spin to gauge the quality of the work.

Problem 4 [Chrono track]. Go to <http://lim.ii.udc.es/mbsbenchmark/docs/specifications.html>. Pick up the pendulum example and carry out the analysis whose results are reported under the “View results” link. For this problem:

- Generate a movie of the motion of the pendulum
- Generate a plot that shows the magnitude of the reaction force in the joint
- Generate a plot of the time evolution of the tip of the pendulum (three plots, for x, y, and z)
- Generate a plot of the velocity of the tip of the pendulum as a function of time (three plots, for x, y, and z)

The results should be uploaded in a zip file at Learn@UW.