Homework 6

Travis Sheperd
ME 751

Problem 1. [MATLAB/Python/C] Implement two (possibly more, if you choose to) functions that provide all the computational kinematics quantities that are associated with the basic GCons Φ^{DP2} and Φ^{D}. Stick with the $\mathbf{r}-\mathbf{p}$ formulation. Specifically, your code should be able to return any or all of the following quantities:

(i) The value of the expression of the constraint

(ii) The right-hand side of the velocity equation ν

(iii) The right-hand side of the acceleration equation γ

(iv) The expression of the partial derivatives $\Phi_{\mathbf{r}}$ and $\Phi_{\mathbf{p}}$

You might decide to have one function for Φ^{DP2} and yet another function for Φ^{D} to produce for these two GCons the quantities in (i) through (iv) above. Since you don't need all these quantities all the time, you should devise a methodology (perhaps using flags) to instruct the subroutine what quantities are actually needed.

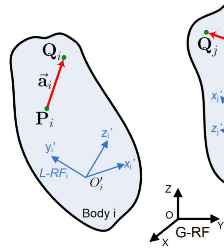
$$\Phi^{DP1}(i,\bar{\mathbf{a}}_i,j,\bar{\mathbf{a}}_j,f(t)) = \bar{\mathbf{a}}_i^T\mathbf{A}_i^T\mathbf{A}_j\bar{\mathbf{a}}_j - f(t) = 0$$

$$\hat{\gamma}^{DP1}(i,\bar{\mathbf{a}}_i,j,\bar{\mathbf{a}}_j,f(t)) = -\mathbf{a}_i^T\mathbf{B}(\dot{\mathbf{p}}_j,\bar{\mathbf{a}}_j)\dot{\mathbf{p}}_j - \mathbf{a}_j^T\mathbf{B}(\dot{\mathbf{p}}_i,\bar{\mathbf{a}}_i)\dot{\mathbf{p}}_i - 2\dot{\mathbf{a}}_i^T\dot{\mathbf{a}}_j + \ddot{f}(t)$$

$$\nu^{DP1} = -\frac{\partial\Phi^{DP1}}{\partial t} = \frac{\partial f}{\partial t}$$

$$\Phi_{\mathbf{q}}^{DP1} = \begin{bmatrix} \frac{\partial\Phi^{DP1}}{\partial\mathbf{r}_i} & \frac{\partial\Phi^{DP1}}{\partial\mathbf{p}_i} & \frac{\partial\Phi^{DP1}}{\partial\mathbf{r}_j} & \frac{\partial\Phi^{DP1}}{\partial\mathbf{p}_j} \end{bmatrix}$$

$$\Phi_{\mathbf{q}}^{DP1} = \begin{bmatrix} \mathbf{0}_{1\times3} & \mathbf{a}_j^T\mathbf{B}(\mathbf{p}_i,\bar{\mathbf{a}}_i) & \mathbf{0}_{1\times3} & \mathbf{a}_i^T\mathbf{B}(\mathbf{p}_j,\bar{\mathbf{a}}_j) \end{bmatrix}$$

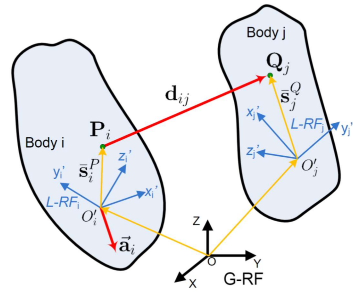


$$\Phi^{DP2}(i,\bar{\mathbf{a}}_i,\bar{\mathbf{s}}_i^P,j,\bar{\mathbf{s}}_j^Q,f(t)) = \bar{\mathbf{a}}_i^T\mathbf{A}_i^T\mathbf{d}_{ij} - f(t) = \bar{\mathbf{a}}_i^T\mathbf{A}_i^T(\mathbf{r}_j + \mathbf{A}_j\bar{\mathbf{s}}_j^Q - \mathbf{r}_i - \mathbf{A}_i\bar{\mathbf{s}}_i^P) - f(t) = 0$$

$$\hat{\gamma}^{DP2}(i,\bar{\mathbf{a}}_i,\bar{\mathbf{s}}_i^P,j,\bar{\mathbf{s}}_j^Q,f(t)) = -\mathbf{a}_i^T\mathbf{B}(\dot{\mathbf{p}}_j,\bar{\mathbf{s}}_j^Q)\dot{\mathbf{p}}_j + \mathbf{a}_i^T\mathbf{B}(\dot{\mathbf{p}}_i,\bar{\mathbf{s}}_i^P)\dot{\mathbf{p}}_i - \mathbf{d}_{ij}^T\mathbf{B}(\dot{\mathbf{p}}_i,\bar{\mathbf{a}}_i)\dot{\mathbf{p}}_i - 2\dot{\mathbf{a}}_i^T\dot{\mathbf{d}}_{ij} + \ddot{f}(t)$$

$$\nu^{DP2} = -\frac{\partial\Phi^{DP2}}{\partial t} = \frac{\partial f}{\partial t}$$

$$\begin{aligned} \Phi_{\mathbf{q}_i,\mathbf{q}_j}^{DP2}(\mathbf{a}_i,\mathbf{d}_{ij}) &= \mathbf{a}_i^T(\mathbf{d}_{ij})_{\mathbf{q}_i,\mathbf{q}_j} + \mathbf{d}_{ij}^T(\mathbf{a}_i)_{\mathbf{q}_i,\mathbf{q}_j} \\ &= \mathbf{a}_i^T[\ -\mathbf{I}_3 \quad -\mathbf{B}(\mathbf{p}_i,\bar{\mathbf{s}}_i^P) \quad \mathbf{I}_3 \quad \mathbf{B}(\mathbf{p}_j,\bar{\mathbf{s}}_j^Q)\ ] + \mathbf{d}_{ij}^T[\ \mathbf{0} \quad \mathbf{B}(\mathbf{p}_i,\bar{\mathbf{a}}_i) \quad \mathbf{0} \quad \mathbf{0}\ ] \\ &= [\ -\mathbf{a}_i^T \quad \mathbf{d}_{ij}^T\mathbf{B}(\mathbf{p}_i,\bar{\mathbf{a}}_i) - \mathbf{a}_i^T\mathbf{B}(\mathbf{p}_i,\bar{\mathbf{s}}_i^P) \quad \mathbf{a}_i^T \quad \mathbf{a}_i^T\mathbf{B}(\mathbf{p}_j,\bar{\mathbf{s}}_j^Q)\ ] \end{aligned}$$

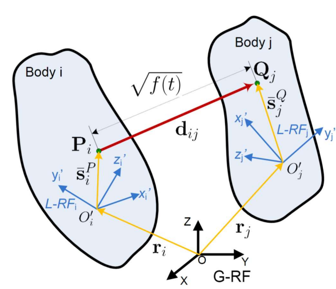


$$\Phi^{D}(i,\bar{\mathbf{s}}_i^P,j,\bar{\mathbf{s}}_j^Q,f(t)) = \mathbf{d}_{ij}^T\mathbf{d}_{ij} - f(t) = (\mathbf{r}_j + \mathbf{A}_j\bar{\mathbf{s}}_j^Q - \mathbf{r}_i - \mathbf{A}_i\bar{\mathbf{s}}_i^P)^T(\mathbf{r}_j + \mathbf{A}_j\bar{\mathbf{s}}_j^Q - \mathbf{r}_i - \mathbf{A}_i\bar{\mathbf{s}}_i^P) - f(t) = 0$$

$$\hat{\gamma}^{D}(i,\bar{\mathbf{s}}_i^P,j,\bar{\mathbf{s}}_j^Q,f(t)) = -2\mathbf{d}_{ij}^T\mathbf{B}(\dot{\mathbf{p}}_j,\bar{\mathbf{s}}_j^Q)\dot{\mathbf{p}}_j + 2\mathbf{d}_{ij}^T\mathbf{B}(\dot{\mathbf{p}}_i,\bar{\mathbf{s}}_i^P)\dot{\mathbf{p}}_i - 2\dot{\mathbf{d}}_{ij}^T\dot{\mathbf{d}}_{ij} + \ddot{f}(t)$$

$$\nu^{D} = -\frac{\partial\Phi^{D}}{\partial t} = \frac{\partial f}{\partial t}$$

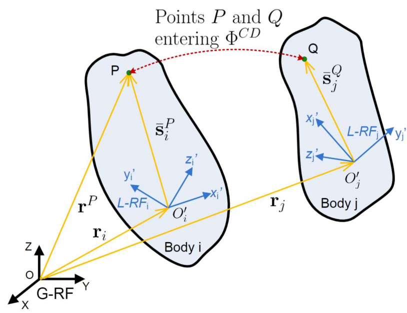
$$\begin{aligned} \Phi_{\mathbf{q}_i,\mathbf{q}_j}^{D} &= (\mathbf{d}_{ij}^T\mathbf{d}_{ij})_{\mathbf{q}_i,\mathbf{q}_j} = 2\mathbf{d}_{ij}^T[\mathbf{d}_{ij}]_{\mathbf{q}_i,\mathbf{q}_j} \\ &= 2\mathbf{d}_{ij}^T[\ -\mathbf{I}_3 \quad -\mathbf{B}(\mathbf{p}_i,\bar{\mathbf{s}}_i^P) \quad \mathbf{I}_3 \quad \mathbf{B}(\mathbf{p}_j,\bar{\mathbf{s}}_j^Q)\ ] \\ &= [\ -2\mathbf{d}_{ij}^T \quad -2\mathbf{d}_{ij}^T\mathbf{B}(\mathbf{p}_i,\bar{\mathbf{s}}_i^P) \quad 2\mathbf{d}_{ij}^T \quad 2\mathbf{d}_{ij}^T\mathbf{B}(\mathbf{p}_j,\bar{\mathbf{s}}_j^Q)\ ] \end{aligned}$$



$$
\begin{aligned}
\Phi^{CD}(\mathbf{c}, i, \bar{\mathbf{s}}_i^P, j, \bar{\mathbf{s}}_j^Q, f(t)) &= \mathbf{c}^T \mathbf{d}_{ij} - f(t) = \mathbf{c}^T(\mathbf{r}_j + \mathbf{A}_j \bar{\mathbf{s}}_j^Q - \mathbf{r}_i - \mathbf{A}_i \bar{\mathbf{s}}_i^P) - f(t) = 0 \\
\hat{\gamma}^{CD}(\mathbf{c}, i, \bar{\mathbf{s}}_i^P, j, \bar{\mathbf{s}}_j^Q, f(t)) &= \mathbf{c}^T \mathbf{B}(\dot{\mathbf{p}}_i, \bar{\mathbf{s}}_i^P)\dot{\mathbf{p}}_i - \mathbf{c}^T \mathbf{B}(\dot{\mathbf{p}}_j, \bar{\mathbf{s}}_j^Q)\dot{\mathbf{p}}_j + \ddot{f}(t)
\end{aligned}
$$

$$
\nu^D = -\frac{\partial \Phi^D}{\partial t} = \frac{\partial f}{\partial t}
$$

$$
\begin{aligned}
\Phi^{CD}_{\mathbf{q}_i, \mathbf{q}_j} &= (\mathbf{c}^T \mathbf{d}_{ij})_{\mathbf{q}_i, \mathbf{q}_j} = \mathbf{c}^T [\mathbf{d}_{ij}]_{\mathbf{q}_i, \mathbf{q}_j} \\
&= \mathbf{c}^T \begin{bmatrix} -\mathbf{I}_3 & -\mathbf{B}(\mathbf{p}_i, \bar{\mathbf{s}}_i^P) & \mathbf{I}_3 & \mathbf{B}(\mathbf{p}_j, \bar{\mathbf{s}}_j^Q) \end{bmatrix} \\
&= \begin{bmatrix} -\mathbf{c}^T & -\mathbf{c}^T \mathbf{B}(\mathbf{p}_i, \bar{\mathbf{s}}_i^P) & \mathbf{c}^T & \mathbf{c}^T \mathbf{B}(\mathbf{p}_j, \bar{\mathbf{s}}_j^Q) \end{bmatrix}
\end{aligned}
$$



$$
\mathbf{A} = [(2e_0^2 - 1)\mathbf{I} + 2(\mathbf{e}\mathbf{e}^T + e_0 \tilde{\mathbf{e}})] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = 2 \begin{bmatrix} e_0^2 + e_1^2 - \frac{1}{2} & e_1 e_2 - e_0 e_3 & e_1 e_3 + e_0 e_2 \\ e_1 e_2 + e_0 e_3 & e_0^2 + e_2^2 - \frac{1}{2} & e_2 e_3 - e_0 e_1 \\ e_1 e_3 - e_0 e_2 & e_2 e_3 + e_0 e_1 & e_0^2 + e_3^2 - \frac{1}{2} \end{bmatrix}
$$

$$
\mathbf{B}(\mathbf{p}, \bar{\mathbf{a}}) \equiv 2 \begin{bmatrix} (e_0 \mathbf{I}_3 + \tilde{\mathbf{e}})\bar{\mathbf{a}} & \mathbf{e}\bar{\mathbf{a}}^{\mathrm{T}} - (e_0 \mathbf{I}_3 + \tilde{\mathbf{e}})\tilde{\bar{\mathbf{a}}} \end{bmatrix}
$$

$$
[\mathbf{A}(\mathbf{p})\bar{\mathbf{a}}]_{\mathbf{p}} = \mathbf{a}_{\mathbf{p}} = \mathbf{B}(\mathbf{p}, \bar{\mathbf{a}}) \in \mathbb{R}^{3\times 4}
$$

$$
\mathbf{p} = \begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{bmatrix}
$$

$$
\mathbf{a}_i = \mathbf{A}_i \bar{\mathbf{a}}_i \quad \Rightarrow \quad \dot{\mathbf{a}}_i = \mathbf{B}(\mathbf{p}_i, \bar{\mathbf{a}}_i)\dot{\mathbf{p}}_i \quad \Rightarrow \quad \ddot{\mathbf{a}}_i = \mathbf{B}(\dot{\mathbf{p}}_i, \bar{\mathbf{a}}_i)\dot{\mathbf{p}}_i + \mathbf{B}(\mathbf{p}_i, \bar{\mathbf{a}}_i)\ddot{\mathbf{p}}_i
$$

asside:

$$
\tilde{a} = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}
$$

$$
\dot{\mathbf{d}}_{ij} = \dot{\mathbf{r}}_j + \mathbf{B}(\mathbf{p}_j, \bar{\mathbf{s}}_j^Q)\dot{\mathbf{p}}_j - \dot{\mathbf{r}}_i - \mathbf{B}(\mathbf{p}_i, \bar{\mathbf{s}}_i^P)\dot{\mathbf{p}}_i
$$

$$
\ddot{\mathbf{d}}_{ij} = \ddot{\mathbf{r}}_j + \mathbf{B}(\mathbf{p}_j, \bar{\mathbf{s}}_j^Q)\ddot{\mathbf{p}}_j + \mathbf{B}(\dot{\mathbf{p}}_j, \bar{\mathbf{s}}_j^Q)\dot{\mathbf{p}}_j - \ddot{\mathbf{r}}_i - \mathbf{B}(\mathbf{p}_i, \bar{\mathbf{s}}_i^P)\ddot{\mathbf{p}}_i - \mathbf{B}(\dot{\mathbf{p}}_i, \bar{\mathbf{s}}_i^P)\dot{\mathbf{p}}_i
$$

Problem 2. [MATLAB/Python/C] In an input file "`revJoint.mdl`" (or in your `driver.py` file, if you chose to hardcode the GCon attributes), define your first mechanism model that will have one pendulum hanging as shown in Figure 1. The pendulum has $L = 2$, it is symmetric as shown in the Figure 1, 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 pendulum is subjected to a motion specified as $f(t) = \frac{\pi}{4}\cos(2t)$. One way to implement this motion is through a *DP1* basic geometric constraint.

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

Your code should be able to provide the following information at $t = 0$: value of $\mathbf{\Phi}(\mathbf{q}, t)$, $\mathbf{\Phi}_{\mathbf{q}}$, ν, and γ. All units used are SI. The program that will be executed for this problem should be named "`simEngine3D-A6P2`" followed by the filename extension of your programming language.

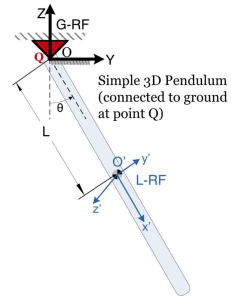


Figure 1: Pendulum with revolute joint.

joint: revolute $\Rightarrow$ $3 \times$ CD

$2 \times$ DP1

CD_1 : $\hat{i}\cdot(\vec{o} - \vec{r}^{a}) = 0$ $\quad \bar{s}^{a} = (-2,0,0)^T$

CD_2 : $\hat{j}\cdot(\vec{o} - \vec{r}^{a}) = 0$ $\Rightarrow f(t) = 0$

CD_3 : $\hat{k}\cdot(\vec{o} - \vec{r}^{a}) = 0$

$DP1_1$: $\hat{x}'\cdot\hat{x} = 0$

$DP1_2$: $\hat{y}'\cdot\hat{x} = 0$ $\Rightarrow f(t) = 0$

Driving : $\hat{x}'\cdot -\hat{z} = \overset{1}{\cancel{|\hat{x}'||\hat{z}|}}\cos(\theta(t)) = \cos\left(\frac{\pi}{4}\cos(2t)\right)$

Note: I had a problem in prob. 3 using this; I would get the absolute val. of the answer for y position of the origin
$\Rightarrow$ I switch this to:

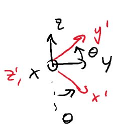
$$\hat{y}'\cdot(-\hat{z}) = 1\cdot1\cdot\cos\left(\theta(t) + \frac{\pi}{2}\right) = \cos\left(\frac{\pi}{4}\cos(2t) + \frac{\pi}{2}\right)$$

$$\Rightarrow f(t) = \cos\left(\frac{\pi}{4}\cos(2t) + \frac{\pi}{2}\right)$$

$$f'(t) = \frac{\pi}{2}\sin(2t)\cos\left(\frac{\pi}{4}\cos(2t)\right)$$

$$f''(t) = \pi\cos(2t)\cos\left(\frac{\pi}{4}\cos(2t)\right) + \frac{\pi^2}{4}\sin^2(2t)\sin\left(\frac{\pi}{4}\cos(2t)\right)$$

initial conditions



$$\hat{x}' = 0\hat{x} + \sin\theta\,\hat{y} - \cos\theta\,\hat{z}$$
$$\hat{y}' = 0\hat{x} + \cos\theta\,\hat{y} + \sin\theta\,\hat{z}$$
$$\hat{z}' = 1\hat{x} + 0\hat{y} + 0\hat{z}$$

$$\Rightarrow A = [\hat{x}' \quad \hat{y}' \quad \hat{z}'] = \begin{bmatrix} 0 & 0 & 1 \\ \sin\theta & \cos\theta & 0 \\ -\cos\theta & \sin\theta & 0 \end{bmatrix}$$

$$\theta(0) = \frac{\pi}{4}\cos(2\cdot 0)) = \frac{\pi}{4}$$

$$e_0 = \left(\frac{tr(A)+1}{4}\right)^{1/2} = \left(\frac{\cos\left(\frac{\pi}{4}\right)+1}{4}\right)^{1/2} = 0.453281\ldots$$

- Step 1: Compute e_0^2 as

$e_0^2 = \frac{tr(\mathbf{A})+1}{\cdot}$

$e_0 = (\quad 4 \quad , \quad (\quad , \quad)$

- Step 1: Compute e_0^2 as

$$e_0^2 = \frac{tr(\mathbf{A})+1}{4}$$

- Step 2: Compute e_0 from e_0^2; choose any sign you wish for e_0, maybe should be close to some previous value of it that you just computed
- Step 3: For $i = 1, 2, 3$, compute

$$e_i^2 = \frac{2a_{ii} - tr(\mathbf{A})+1}{4}$$

- Step 4: When computing e_i from e_i^2, the sign of e_i should be determined based on the following relations:

$$a_{32} - a_{23} = 4e_0e_1$$
$$a_{13} - a_{31} = 4e_0e_2$$
$$a_{21} - a_{12} = 4e_0e_3$$

$$e_1 = \left(\frac{2(0) - \cos\left(\frac{\pi}{4}\right)+1}{4}\right)^{1/2} = 0.27\ldots$$

$e_2 = .5\ldots \qquad e_3 = 0.27\ldots$

$$\vec{r}_{ini} = A \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} = \begin{pmatrix} 0 \\ 1.41 \\ -1.41 \end{pmatrix}$$

use to find the analytical sol'ns

$$\mathbf{r}^P = \mathbf{r} + \mathbf{s}^P = \mathbf{r} + \mathbf{A}\bar{\mathbf{s}}^P$$

$$\vec{\mathbf{v}}^P = \frac{d\vec{\mathbf{r}}^P}{dt} = \dot{\vec{\mathbf{r}}} + \dot{\vec{\mathbf{s}}}^P = \dot{\vec{\mathbf{r}}} + \vec{\omega} \times \vec{\mathbf{s}}^P$$

$$\vec{\mathbf{a}}^P \equiv \frac{d^2\vec{\mathbf{r}}^P}{dt^2} = \ddot{\vec{\mathbf{r}}} + \vec{\omega} \times \vec{\omega} \times \vec{\mathbf{s}}^P + \dot{\vec{\omega}} \times \vec{\mathbf{s}}^P$$

$$\dot{\mathbf{r}}^P = \dot{\mathbf{r}} + \dot{\mathbf{s}}^P = \dot{\mathbf{r}} + \dot{\mathbf{A}}\bar{\mathbf{s}}^P = \dot{\mathbf{r}} + \tilde{\omega}\mathbf{A}\bar{\mathbf{s}}^P = \dot{\mathbf{r}} + \tilde{\omega}\mathbf{s}^P$$

$$\mathbf{a}^P \equiv \ddot{\mathbf{r}}^P = \ddot{\mathbf{r}} + \ddot{\mathbf{s}}^P = \ddot{\mathbf{r}} + \tilde{\omega}\tilde{\omega}\mathbf{A}\bar{\mathbf{s}}^P + \tilde{\dot{\omega}}\mathbf{A}\bar{\mathbf{s}}^P = \ddot{\mathbf{r}} + \tilde{\omega}\tilde{\omega}\mathbf{s}^P + \tilde{\dot{\omega}}\mathbf{s}^P$$

$$\vec{r}^Q = \vec{r}^O + \vec{\bar{a}}^Q \qquad \vec{\bar{a}}^Q = \begin{bmatrix} -2 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{a}_i = \mathbf{A}_i\bar{\mathbf{a}}_i \quad \Rightarrow \quad \dot{\mathbf{a}}_i = \mathbf{B}(\mathbf{p}_i, \bar{\mathbf{a}}_i)\dot{\mathbf{p}}_i \quad \Rightarrow \quad \ddot{\mathbf{a}}_i = \mathbf{B}(\dot{\mathbf{p}}_i, \bar{\mathbf{a}}_i)\dot{\mathbf{p}}_i + \mathbf{B}(\mathbf{p}_i, \bar{\mathbf{a}}_i)\ddot{\mathbf{p}}_i$$

construct a global Φ, nu, γ, Φ_p

$$\Phi^K(\mathbf{q}, t) = \begin{bmatrix} \Phi^G(\mathbf{q}) \\ \Phi^D(\mathbf{q}, t) \end{bmatrix} = \mathbf{0}_{6nb} \qquad \Phi^F(\mathbf{q}, t) = \begin{bmatrix} \Phi^G(\mathbf{q}) \\ \Phi^D(\mathbf{q}, t) \\ \Phi^{\mathbf{P}}(\mathbf{p}) \end{bmatrix} = \mathbf{0}_{7nb}$$

$$\nu^{\mathbf{P}} = \mathbf{0}_{nb} \quad \& \quad \gamma^{\mathbf{P}} = \begin{bmatrix} -2\dot{\mathbf{p}}_1^T\dot{\mathbf{p}}_1 \\ \cdots \\ -2\dot{\mathbf{p}}_{nb}^T\,\dot{\mathbf{p}}_{nb} \end{bmatrix} \qquad \Phi_p^P = \left(2p_i \quad \ddots \right)$$

$$\begin{aligned} \left[\boxed{\Phi_{\mathbf{q}}^F(\mathbf{q}^{(k)})}\right] \Delta\mathbf{q}^{(k+1)} &= \Phi(\mathbf{q}^{(k)}, t) \\ \mathbf{q}^{(k+1)} &= \mathbf{q}^{(k)} - \Delta\mathbf{q}^{(k+1)} \end{aligned}$$

solve for jacobian global => newton raphson to get q at each timestep

$$\left.\begin{aligned} \Phi_{\mathbf{q}}\ddot{\mathbf{q}} &= \gamma_{7nb} \\ \Phi_{\mathbf{q}}\dot{\mathbf{q}} &= \nu_{7nb} \end{aligned}\right\}$$ use these to solve for $\dot{q}, \ddot{q}$

```
e0_ini =
    0.6533
e1_ini =
    0.2706
e2_ini =
    0.5000
e3_ini =
    0.2706
r_ini =
         0
    1.4142
   -1.4142
```

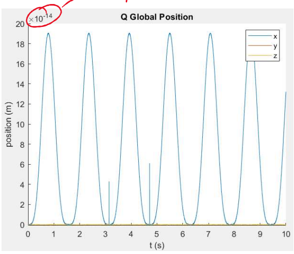
```
Phi
ans =
   1.0e-15 *
   -0.3053
         0
    0.6106
         0
         0
         0
   -0.1110
nu
ans =
     0
     0
     0
     0
     0
     0
     0
     0
```

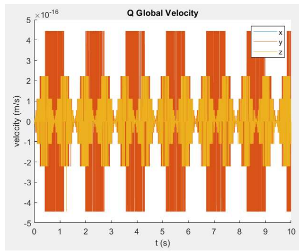
```
gamma
ans =
     0
     0
     0
     0
     0
2.2214
     0
Jacobian
ans =
         0         0         0    1.3066    0.5412   -1.3066   -0.5412
         0         0         0   -0.5412    1.3066    0.5412   -1.3066
    1.0000         0         0   -2.6131   -1.0824    2.6131    1.0824
         0    1.0000         0   -1.0824   -2.6131   -1.0824   -2.6131
         0         0    1.0000    2.6131   -1.0824    2.6131   -1.0824
         0         0         0   -0.5412   -1.3066   -0.5412   -1.3066
         0         0         0    1.3066    0.5412    1.3066    0.5412
```

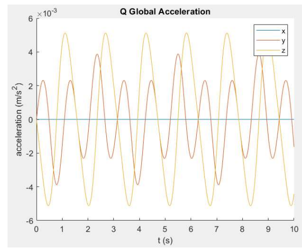
Problem 3. [MATLAB/Python/C] This problem builds on Problem 2. The schematic of the mechanism is shown in Figure 1. The rigid body is subjected to a motion specified as $f(t) = \frac{\pi}{4}\cos(2t)$.

You will need 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 $\Delta t = 10^{-3}$. Add **six plots** to your report including location/velocity/acceleration of point O'/Q in the G-RF as a function of time, and discuss the results that you obtained for point Q. The program that will be executed for this problem should be named "`simEngine3D-A6P3`" followed by the filename extension of your programming language.

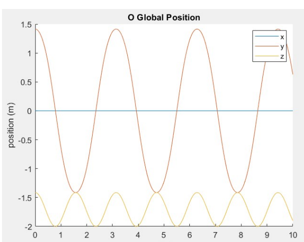
computed zero

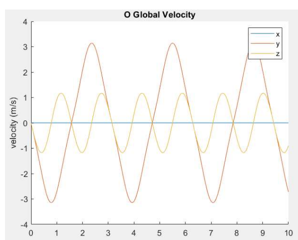


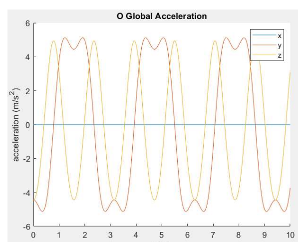




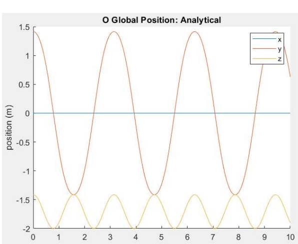
these show approx. computed zeros or noise w/in approx. order of the tolerance => they are ≈ 0. Note: the acceleration term is slightly larger => could have bug

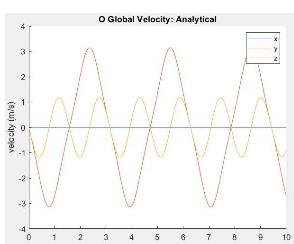


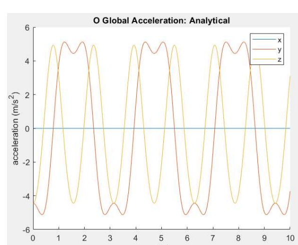




solved very close to analytical







analytical