Travis Sheperd
ME 751

Problem 1. [SimEngine3D track, MATLAB/Python/C] For the simple pendulum of Problem 1 in the previous assignment (Assignment 7), perform a dynamic analysis using a Quasi-Newton approach. The gravitational acceleration (not shown in the picture) is assumed to act in the opposite direction of the global z axis; its magnitude is $g = 9.81$ (in SI units). Use a BDF method of order 2, seeded at the beginning of the simulation by a BDF method of order 1. The simulation should run for $t \in [0, 10]$.

a) Provide a plot that displays the value of the torque as a function of time

b) In your report also include the following **7 plots**. The first three plots will display the x, y, and z coordinates of Body 1's point O' expressed in the G-RF as a function of time. The next three plots will show the angular velocity of Body 1 in the G-RF as a function of time. The last plot will display the 2-norm of the violation of the velocity constraint equations for the revolute joint between Body 1 and ground (you'll have five scalar velocity constraint equations, you need to compute the 2-norm of this vector; i.e., the violation of the velocity kinematic constraint equation)

c) Include in your report the amount of time it took your `simEngine3D` solver to finish the simulation. Indicate the step size you used to generate the results.

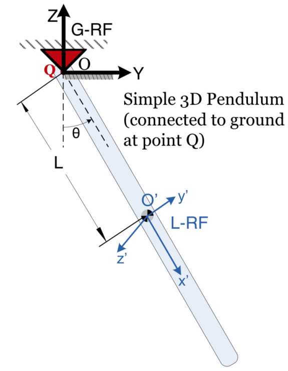
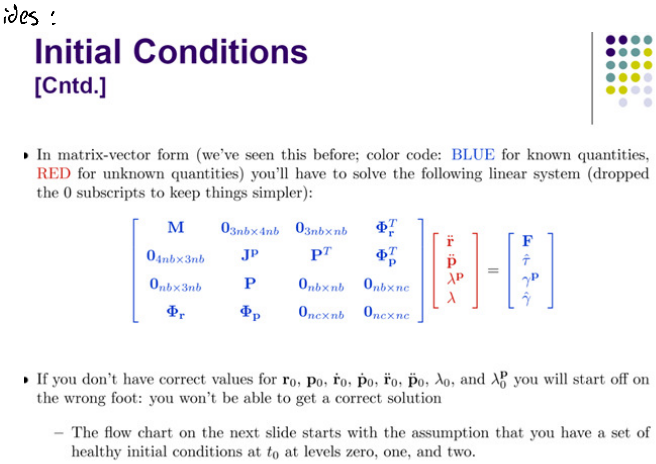


Figure 1: Pendulum with revolute joint.

Overview from slides:



Solving the Dynamics Problem: The Flow Chart, used for n=1, 2, 3, …

STAGE 0 : Prime new time step. Set: $\nu = 0$, $t_n = t_{n-1} + h$, $\ddot{\mathbf{r}}_n^{(0)} = \ddot{\mathbf{r}}_{n-1}$, $\ddot{\mathbf{p}}_n^{(0)} = \ddot{\mathbf{p}}_{n-1}$, $\lambda_n^{(0)} = \lambda_{n-1}$, $\lambda_n^{\mathbf{p}(0)} = \lambda_{n-1}^{\mathbf{p}}$

STAGE 1 : Compute position and velocity using BDF and most recent accelerations $\ddot{\mathbf{r}}_n^{(\nu)}$ and $\ddot{\mathbf{p}}_n^{(\nu)}$:

$$\mathbf{r}_n^{(\nu)} = \mathbf{C}_n^{\mathbf{r}}(l) + \beta_0^2 h^2 \ddot{\mathbf{r}}_n^{(\nu)} \qquad \dot{\mathbf{r}}_n^{(\nu)} = \mathbf{C}_n^{\dot{\mathbf{r}}}(l) + \beta_0 h \ddot{\mathbf{r}}_n^{(\nu)}$$

$$\mathbf{p}_n^{(\nu)} = \mathbf{C}_n^{\mathbf{p}}(l) + \beta_0^2 h^2 \ddot{\mathbf{p}}_n^{(\nu)} \qquad \dot{\mathbf{p}}_n^{(\nu)} = \mathbf{C}_n^{\dot{\mathbf{p}}}(l) + \beta_0 h \ddot{\mathbf{p}}_n^{(\nu)}$$

STAGE 2 : Compute the residual in the nonlinear system; i.e., evaluate $\mathbf{g}(\ddot{\mathbf{r}}_n^{(\nu)}, \ddot{\mathbf{p}}_n^{(\nu)}, \lambda_n^{\mathbf{P}(\nu)}), \lambda_n^{(\nu)}) \equiv \mathbf{g}_n^{(\nu)}$.

STAGE 3 : Solve linear system $\mathbf{\Psi}^{(\nu)} \Delta \mathbf{z}^{(\nu)} = -\mathbf{g}_n^{(\nu)}$ to get correction $\Delta \mathbf{z}^{(\nu)}$. You'll use here an iteration matrix according to the Newton solver of your choice (NR, MN, or QN)

STAGE 4 : Improve the quality of the approximate solution: $\mathbf{z}^{(\nu+1)} = \mathbf{z}^{(\nu)} + \Delta \mathbf{z}^{(\nu)}$

STAGE 5 : Set $\nu = \nu + 1$. If the norm of the correction is small enough, go to STAGE 6. Otherwise, go to STAGE 1 (unless you already took too many iterations and feel like this is not going anywhere in which case you bail out; most likely you either have (a) too large of a step size h, (b) your problem has discontinuities, or (c) you have a bug (very unlikely ;-)

STAGE 6 : Accept the accelerations and lambdas computed in STAGE 4 as your solutions. Using the accelerations, do yet one more time STAGE 1 to get level zero and one variables that are in sync with the accelerations. Save the value of the time step t_n and the level zero, one, and two unknowns in an array since you want to plot the results at the end of simulation. At this point you just finished one integration step. Life is good. Go back to STAGE 0.

first solve for z at the initial time

$$\begin{bmatrix} \mathbf{M} & \mathbf{0}_{3nb\times 4nb} & \mathbf{0}_{3nb\times nb} & \mathbf{\Phi}_{\mathbf{r}}^T \\ \mathbf{0}_{4nb\times 3nb} & \mathbf{J}^{\mathbf{P}} & \mathbf{P}^T & \mathbf{\Phi}_{\mathbf{p}}^T \\ \mathbf{0}_{nb\times 3nb} & \mathbf{P} & \mathbf{0}_{nb\times nb} & \mathbf{0}_{nb\times nc} \\ \mathbf{\Phi}_{\mathbf{r}} & \mathbf{\Phi}_{\mathbf{p}} & \mathbf{0}_{nc\times nb} & \mathbf{0}_{nc\times nc} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{r}} \\ \ddot{\mathbf{p}} \\ \lambda^{\mathbf{P}} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \hat{\tau} \\ \gamma^{\mathbf{P}} \\ \hat{\gamma} \end{bmatrix}$$

next we need $r_n, \dot{r}_n, p_n, \dot{p}_n$ for the iterative soln

- Specifically, for the BDF method of order l and with coefficient β_0, we concluded last time that

$$\mathbf{r}_n = \mathbf{C}_n^{\mathbf{r}}(l) + \beta_0^2 h^2 \ddot{\mathbf{r}}_n \qquad \dot{\mathbf{r}}_n = \mathbf{C}_n^{\dot{\mathbf{r}}}(l) + \beta_0 h \ddot{\mathbf{r}}_n$$

$$\mathbf{p}_n = \mathbf{C}_n^{\mathbf{p}}(l) + \beta_0^2 h^2 \ddot{\mathbf{p}}_n \qquad \dot{\mathbf{p}}_n = \mathbf{C}_n^{\dot{\mathbf{p}}}(l) + \beta_0 h \ddot{\mathbf{p}}_n$$

BDF Methods: $\sum_{i=0}^{k} \alpha_i y_{n-i} = h\beta_0 f(t_n, y_n)$

	p	k	β_0	α_0	α_1	α_2	α_3	α_4	α_5	α_6
BDF 1 →	1	1	1	1	-1					
BDF 2 →	2	2	2/3	1	-4/3	1/3				
	3	3	6/11	1	-18/11	9/11	-2/11			
	4	4	12/25	1	-48/25	36/25	-16/25	3/25		
	5	5	60/137	1	-300/137	300/137	-200/137	75/137	-12/137	
	6	6	60/147	1	-360/147	450/147	-400/147	225/147	-72/147	10/147

$$\dot r_n = \overbrace{-\alpha_1 \dot r_{n-1} - \alpha_2 \dot r_{n-2}}^{c_n^{\dot r}(1)} - \beta_0 h (\ddot r_n)$$

BDF2 $\Rightarrow$

$$r_n = -\alpha_1 r_{n-1} - \alpha_2 r_{n-2} - \beta_0 h (\dot r_n)$$

$$= \underbrace{-\alpha_1 r_{n-1} - \alpha_2 r_{n-2} - \beta_0 h(-\alpha_1 \dot r_{n-1} - \alpha_2 \dot r_{n-2})}_{c_n^{r}(1)} - \beta_0^2 h^2 \ddot r_n$$

same for p_n, $\dot p_n$ for BDF 1 $\alpha_2 \to 0$

$$\mathbf{g}(\ddot{\mathbf{r}}_n, \ddot{\mathbf{p}}_n, \lambda_n, \lambda_n^{\mathbf{P}}) = \begin{bmatrix} \mathbf{M}\ddot{\mathbf{r}}_n + \mathbf{\Phi}_{\mathbf{r}}^T(\mathbf{r}_n, \mathbf{p}_n, t_n)\lambda_n - \mathbf{F}(\dot{\mathbf{r}}_n, \dot{\mathbf{p}}_n, \mathbf{r}_n, \mathbf{p}_n, t_n) \\ \mathbf{J}^{\mathbf{P}}(\mathbf{p}_n)\ddot{\mathbf{p}}_n + \mathbf{\Phi}_{\mathbf{p}}^T(\mathbf{r}_n, \mathbf{p}_n)\lambda_n + \mathbf{P}^T(\mathbf{p}_n)\lambda_n^{\mathbf{P}} - \hat\tau(\dot{\mathbf{r}}_n, \dot{\mathbf{p}}_n, \mathbf{r}_n, \mathbf{p}_n, t_n) \\ \frac{1}{\beta_0^2 h^2}\, \mathbf{\Phi}^{\mathbf{P}}(\mathbf{p}_n) \\ \frac{1}{\beta_0^2 h^2}\, \mathbf{\Phi}(\mathbf{r}_n, \mathbf{p}_n, t_n) \end{bmatrix} = \mathbf{0}_{8nb+nc}$$

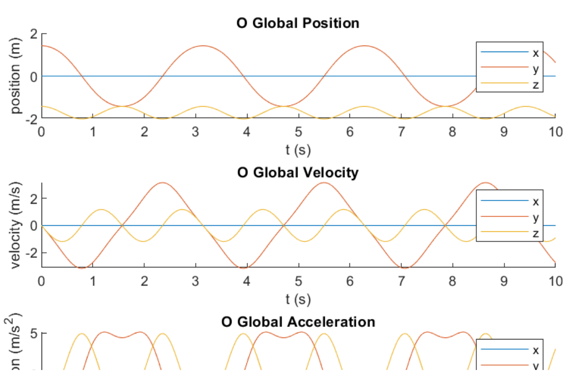
The Quasi-Newton Iteration Matrix

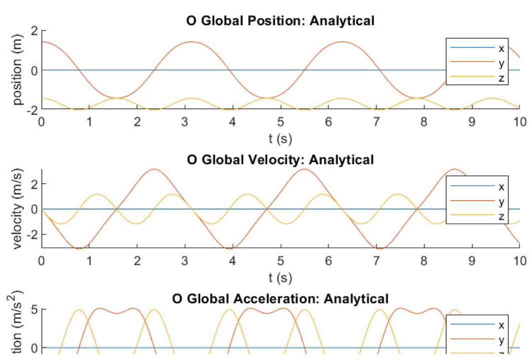
$$\Psi^{(\nu)} \approx \begin{bmatrix} \mathbf{M} & \mathbf{0}_{3nb\times 4nb} & \mathbf{0}_{3nb\times nb} & \mathbf{\Phi}_{\mathbf{r}}^T(\mathbf{r}_n, \mathbf{p}_n, t_n) \\ \mathbf{0}_{4nb\times 3nb} & \mathbf{J}^{\mathbf{P}}(\mathbf{p}_n) & \mathbf{P}^T & \mathbf{\Phi}_{\mathbf{p}}^T(\mathbf{r}_n, \mathbf{p}_n, t_n) \\ \mathbf{0}_{nb\times 3nb} & \mathbf{P} & \mathbf{0}_{nb\times nb} & \mathbf{0}_{nb\times nc} \\ \mathbf{\Phi}_{\mathbf{r}}(\mathbf{r}_n, \mathbf{p}_n, t_n) & \mathbf{\Phi}_{\mathbf{p}}(\mathbf{r}_n, \mathbf{p}_n, t_n) & \mathbf{0}_{nc\times nb} & \mathbf{0}_{nc\times nc} \end{bmatrix}$$

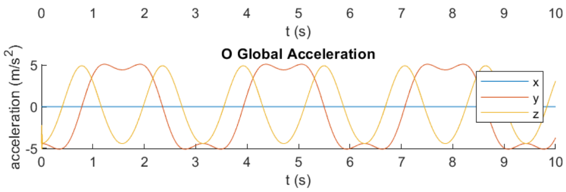
solve for Δz then z, and iterate

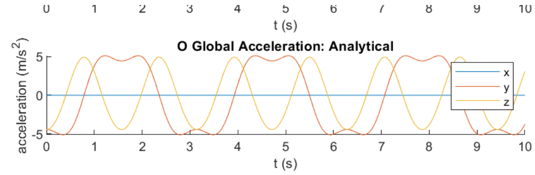
$$\begin{cases} \Psi^{(\nu)} \cdot \Delta \mathbf{z}^{(\nu)} = -\mathbf{g}(\mathbf{z}^{(\nu)}) \\ \mathbf{z}^{(\nu+1)} = \mathbf{z}^{(\nu)} + \Delta\mathbf{z}^{(\nu)} \end{cases} \qquad \text{where I used the notation} \qquad \mathbf{z} = \begin{bmatrix} \ddot{\mathbf{r}}_n \\ \ddot{\mathbf{p}}_n \\ \lambda_n^{\mathbf{P}} \\ \lambda_n \end{bmatrix}$$

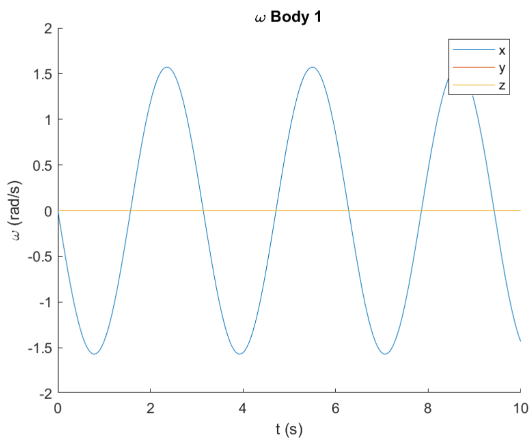
good correspondence w/ exact

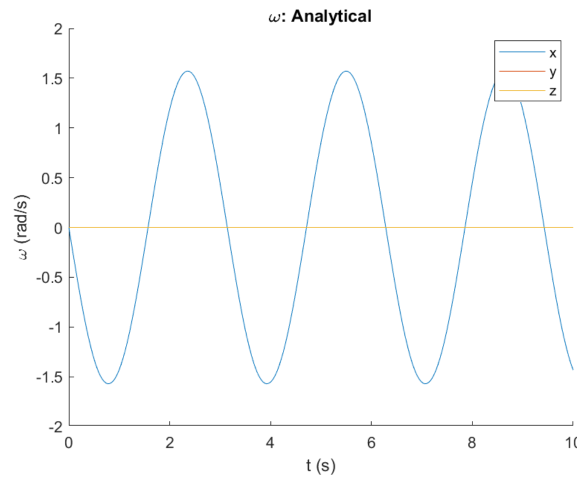


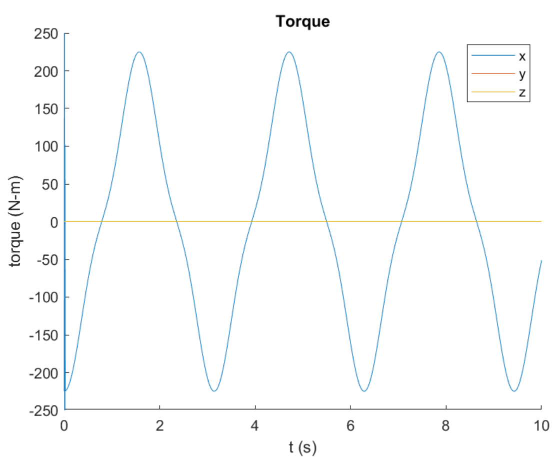


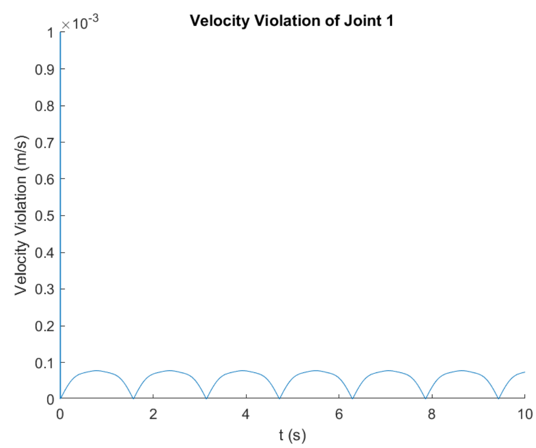












H = 0.005,   TOL = 1e-4,   Elapsed time is 58.087916 seconds.

Problem 2. [SimEngine3D track, MATLAB/Python/C] This problem builds on the simple pendulum of the previous problem. The schematic of the mechanism is shown in Figure 2.

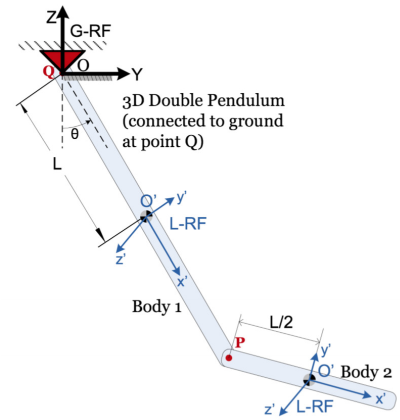


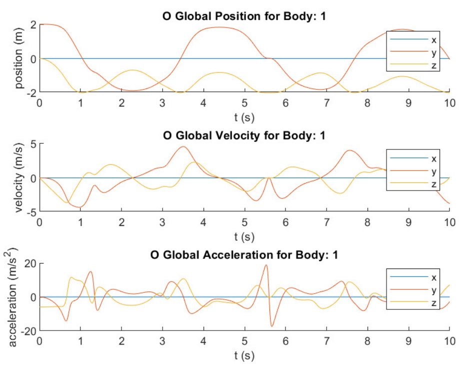
Figure 2: 3D Double Pendulum with revolute joint.

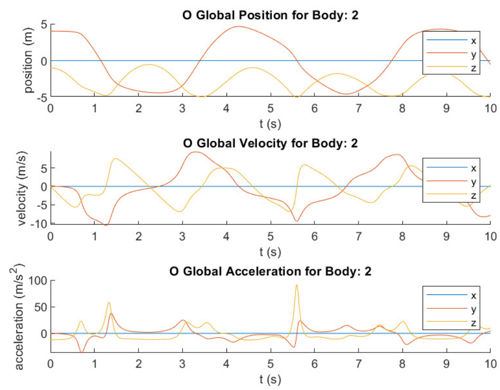
You will have to carry out a Dynamics Analysis for the mechanism for 10 seconds of its evolution using a BDF method of order 1 in conjunction with a Quasi-Newton method for solving the resulting discretization nonlinear system of equations. For initial conditions consider the first pendulum (Body 1) to be horizontal; i.e. according to Figure 2, $\theta = \pi/2$, while the second pendulum (Body 2) is hanging down, making a $\pi/2$ angle with the first pendulum. Both bodies have zero velocity at time $t = 0$. The bodies move in a gravitational field.

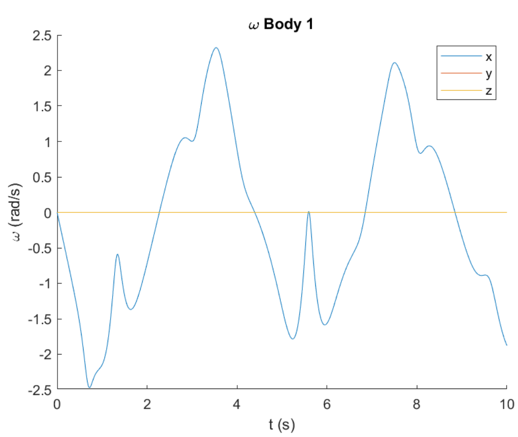
a) In your report include **13 plots**. The first three plots will display the x, y, and z coordinates of Body 1's point O' expressed in the G-RF as a function of time. The next three plots will show the angular velocity of Body 1 in the G-RF as a function of time. The next six plots will display the same information for Body 2's point O'. The last plot will display the 2-norm of the violation of the velocity constraint equations for the revolute joint between Body 1 and Body 2 (you'll have five scalar velocity constraint equations, you need to compute the 2-norm of this vector; i.e., the violation of the velocity kinematic constraint equation)

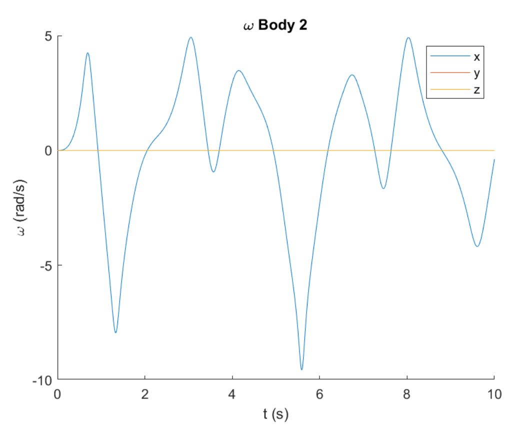
b) Include in your report the amount of time it took your `simEngine3D` solver to finish the simulation. Indicate the step size you used to generate the results (make sure the results are converged and not garbage)

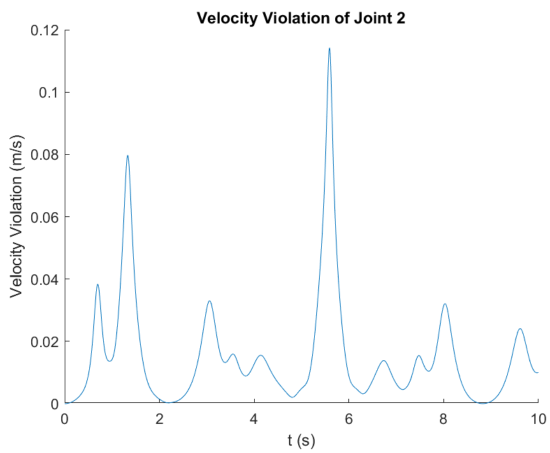
for BDF 1 only





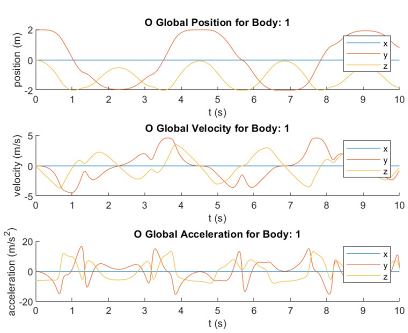


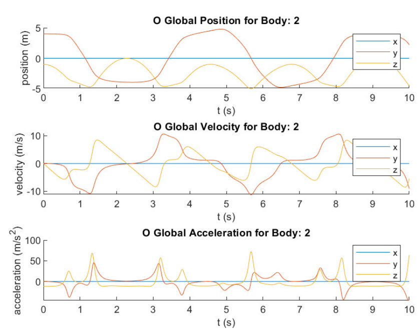


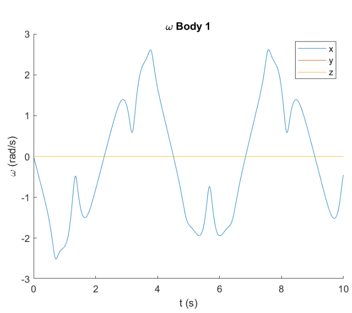


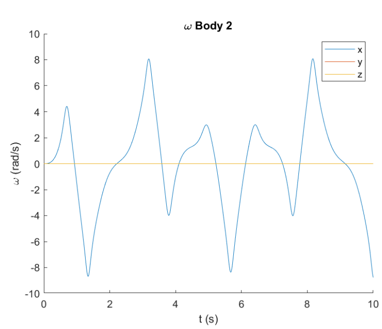
H = 0.005,   TOL = 1e-3,   Elapsed time is 92.642865 seconds.

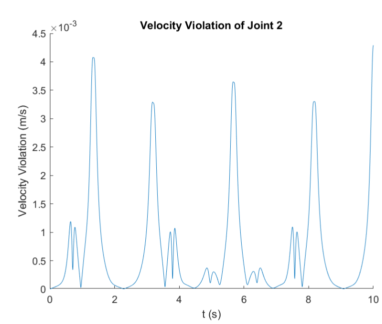
compare to BDF 2 for same h & tol











H = 0.005,   TOL = 1e-3,   Elapsed time is 86.506332 seconds.