Soft Inverted Pendulum Robot - Modelling, Simulation and Control

Aditya Varadaraj, Lasitha Weerakoon

Dr. Nikhil Chopra’s Lab,
University of Maryland - College Park, MD

Model used and Parameters

Modelling scheme: Cosserat Rod Theory

Numerical Solver / Software Used: PyElastica (Python)

Initial Length of rod (L) = 0.13 mts.

Initial radius of the rod (R) = 0.01416 mts.

Density, ϱ = 1180 kg/m^3

Poisson Ratio, γ = 0.5

Young's Modulus, E = 3.79 MPa

Energy Dissipation, ν = 10

Simulation Time, t_f = 10 secs

Shear Modulus, G = E / (2*(1+γ))

Hinged Planar Case:
Stiffness (k) and Damping (β) Experimental Computation

$$\begin{bmatrix} q_0(t_0) & \dot{q}_0(t_0) \\ \vdots & \vdots \\ q_0(t_f) & \dot{q}_0(t_f) \end{bmatrix} \begin{bmatrix} k \\ \beta \end{bmatrix} = \begin{bmatrix} RHS(t_0) \\ \vdots \\ RHS(t_f) \end{bmatrix}$$

$$q = \begin{bmatrix} q_0 \\ \theta \end{bmatrix}$$

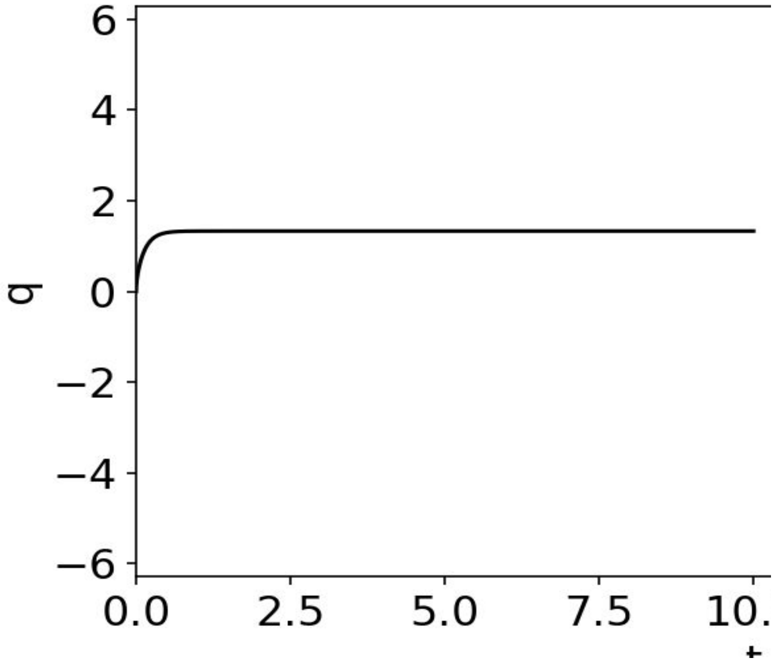
$$RHS(t_i) = \tau - (M(q)q(t_i) + C(q, \dot{q})\dot{q}(t_i))[0]$$

$$\begin{bmatrix} k \\ \beta \end{bmatrix} = \begin{bmatrix} q_0(t_0) & \dot{q}_0(t_0) \\ \vdots & \vdots \\ q_0(t_f) & \dot{q}_0(t_f) \end{bmatrix}^{\dagger} \begin{bmatrix} RHS(t_0) \\ \vdots \\ RHS(t_f) \end{bmatrix}$$

, where, $A^{\dagger}$ means pseudo-inverse of A

Why ν = 10 ?

As we can see from the figure, the damping seems to be enough to have a smooth convergence towards steady-state



Hinged Planar Case: Stiffness (k) and Damping (β) Experimental Computation

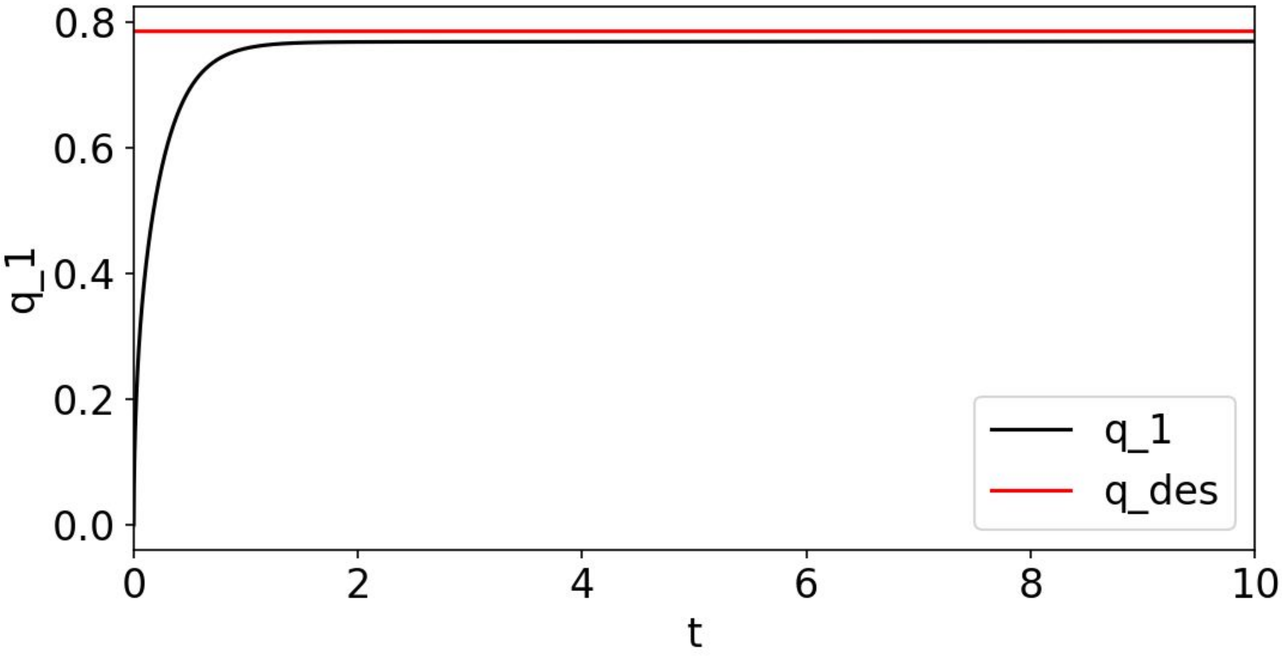
Input Torque (τ) (Nm)	Stiffness (k) (Nm/rad)	Damping (β) (Nsm/rad)	Final Bending Angle (q_0) (rad)
0.3125	0.942	0.07	0.33
0.625	0.942	0.07	0.67
0.9205	0.942	0.07	0.98
1.25	0.942	0.07	1.34
1.5	0.942	0.07	1.60
1.75	0.942	0.07	1.86

Thus, the obtained values for ν = 10 and E = 3.79 MPa conditions are:

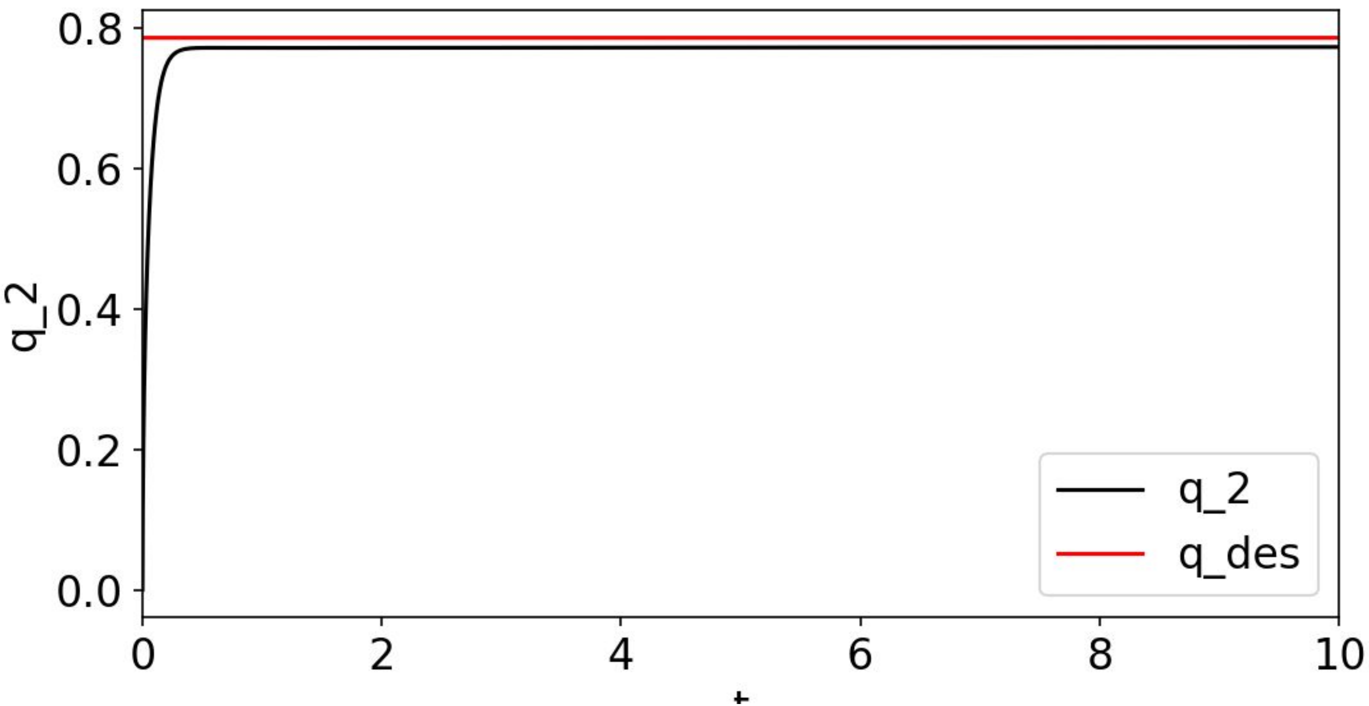
k = 0.942 Nm/rad and β = 0.07 Nsm/rad

Fixed End 3-segment case PD Joint-Space Kinematic Control:

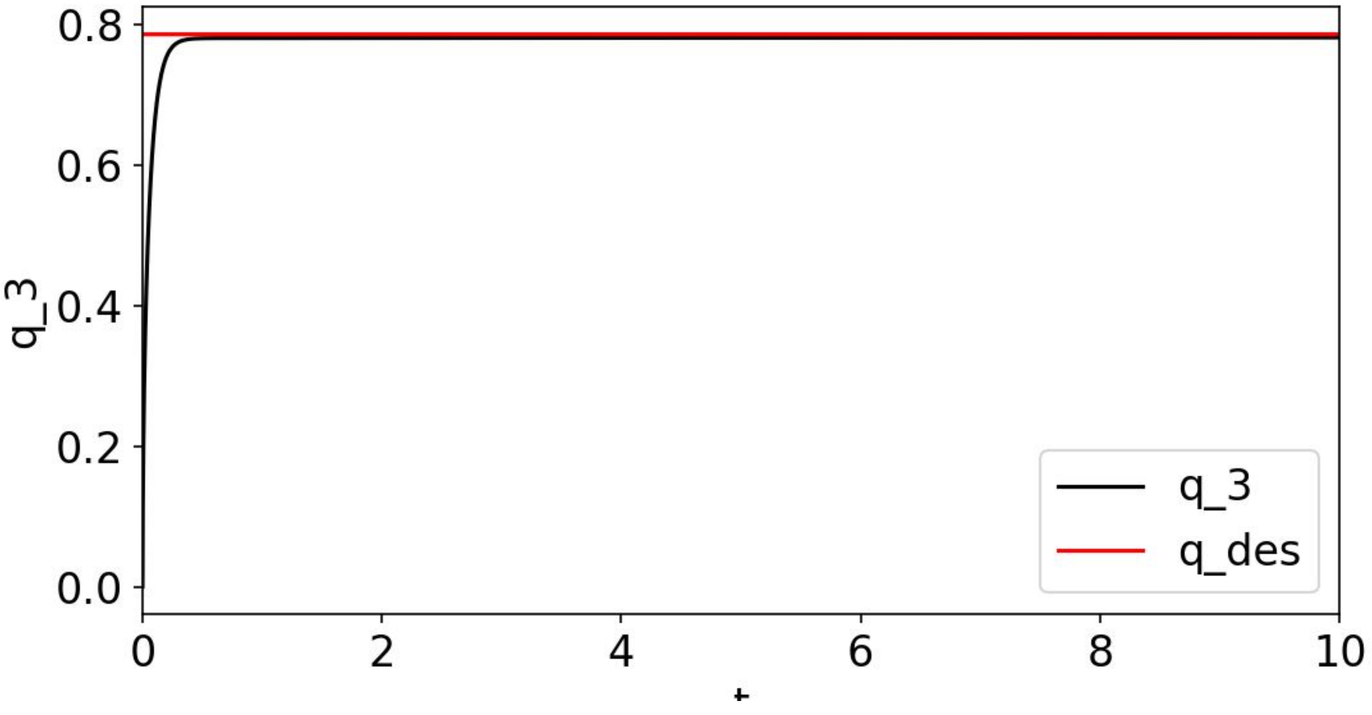
$q_{des1} = q_{des2} = q_{des3} = \pi/4$, Kp = 1.7275, Kd = 0.0075



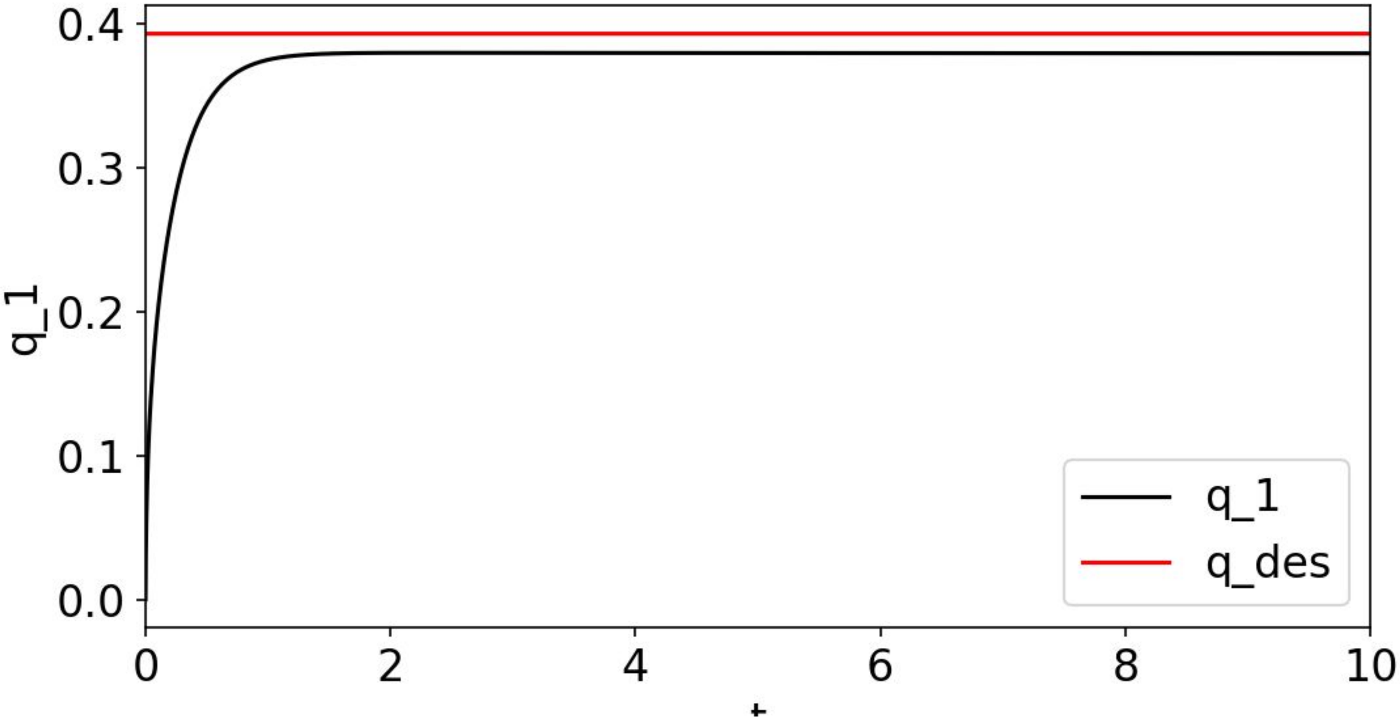
Fixed End 3-segment case PD Joint-Space Control: $q_{des1} = q_{des2} = q_{des3} = \pi/4$, Kp = 1.7275, Kd = 0.0075 (contd.)



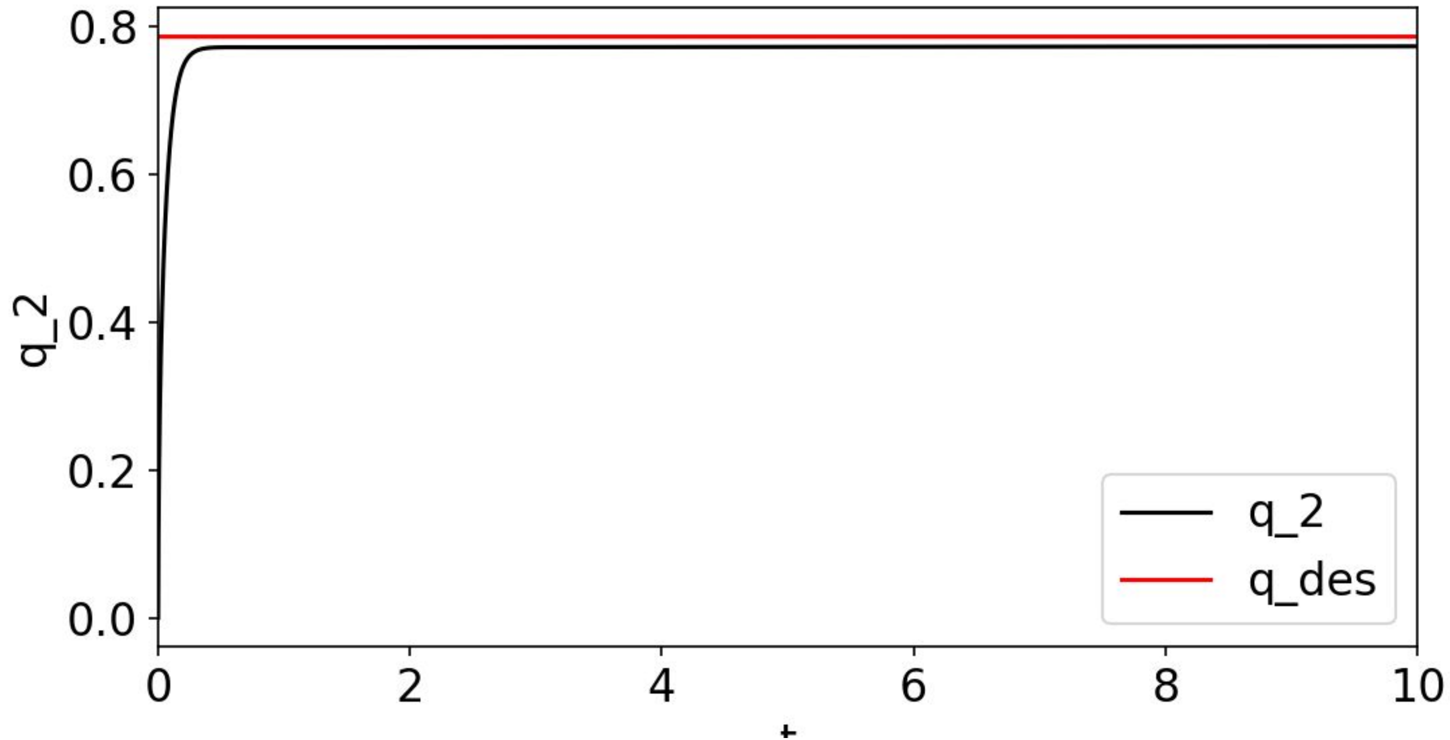
Fixed End 3-segment case PD Joint-Space Control: $q_{des1} = q_{des2} = q_{des3} = \pi/4$, Kp = 1.7275, Kd = 0.0075 (contd.)



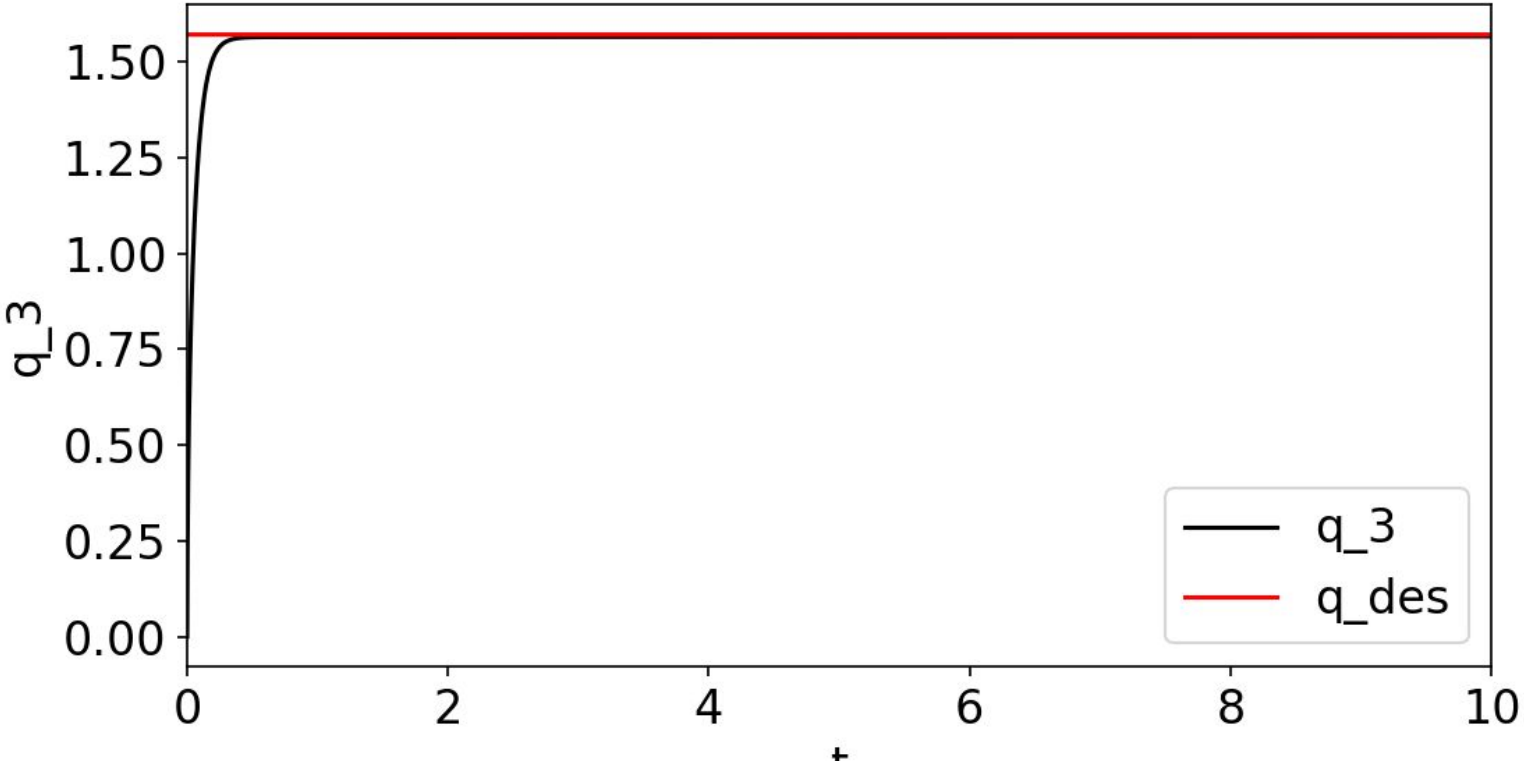
Fixed End 3-segment case PD Joint-Space Control: $q_{des1} = \pi/8$, $q_{des2} = \pi/4$, $q_{des3} = \pi/2$, Kp = 1.7275, Kd = 0.0075



(contd.)



(contd.)



Fixed End 3-segment case PD Joint-Space Control: Time-varying q_{des}, Kp = 3.3275, Kd = 0.0075

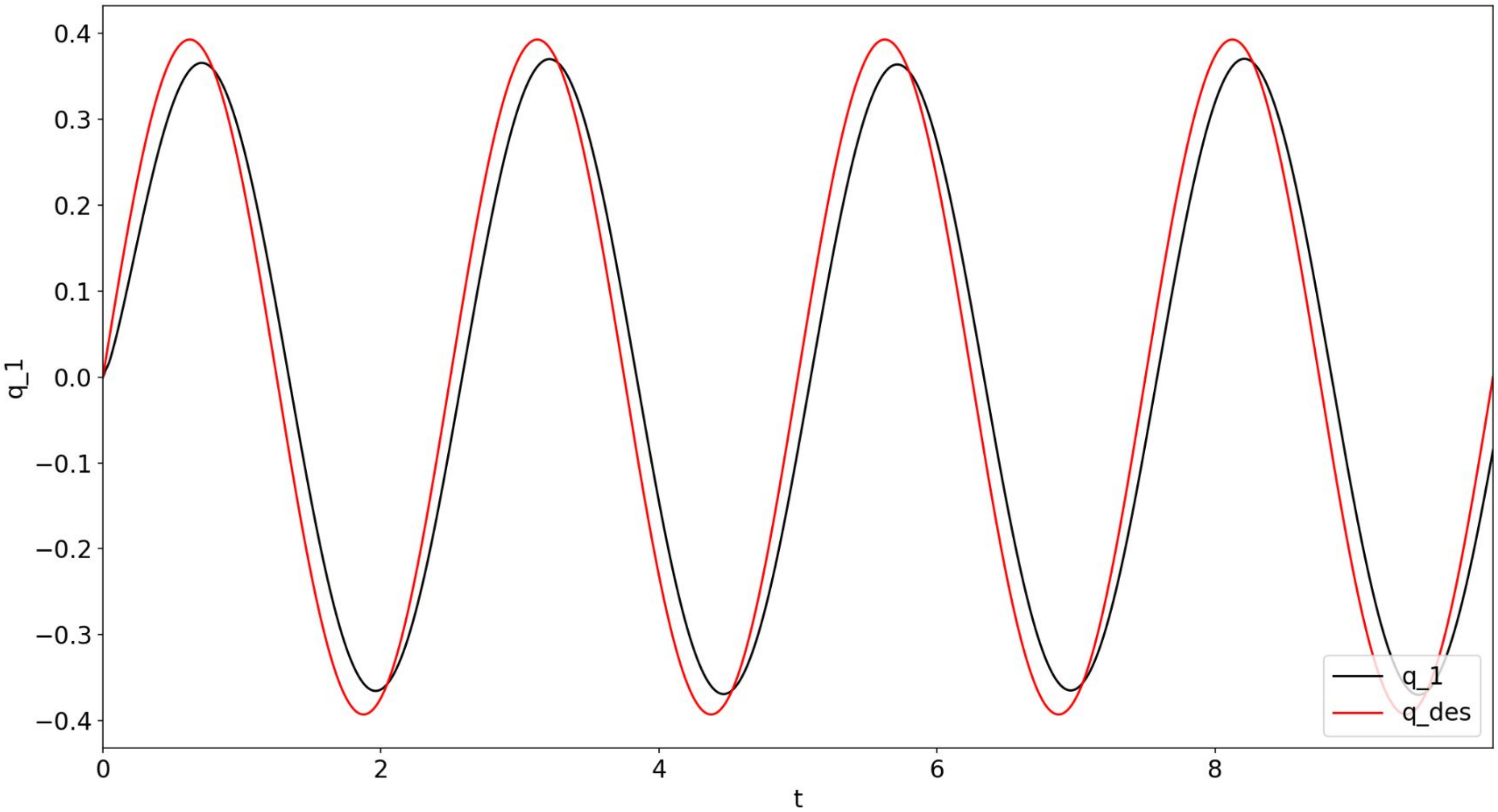
$$q_{des1} = \frac{\pi}{8} \sin\left(\frac{2\pi t}{2.5}\right)$$

$$q_{des2} = \frac{\pi}{4} \cos\left(\frac{2\pi t}{5}\right)$$

$$q_{des3} = \frac{\pi}{2} \sin\left(\frac{2\pi t}{4} + \frac{\pi}{9}\right)$$

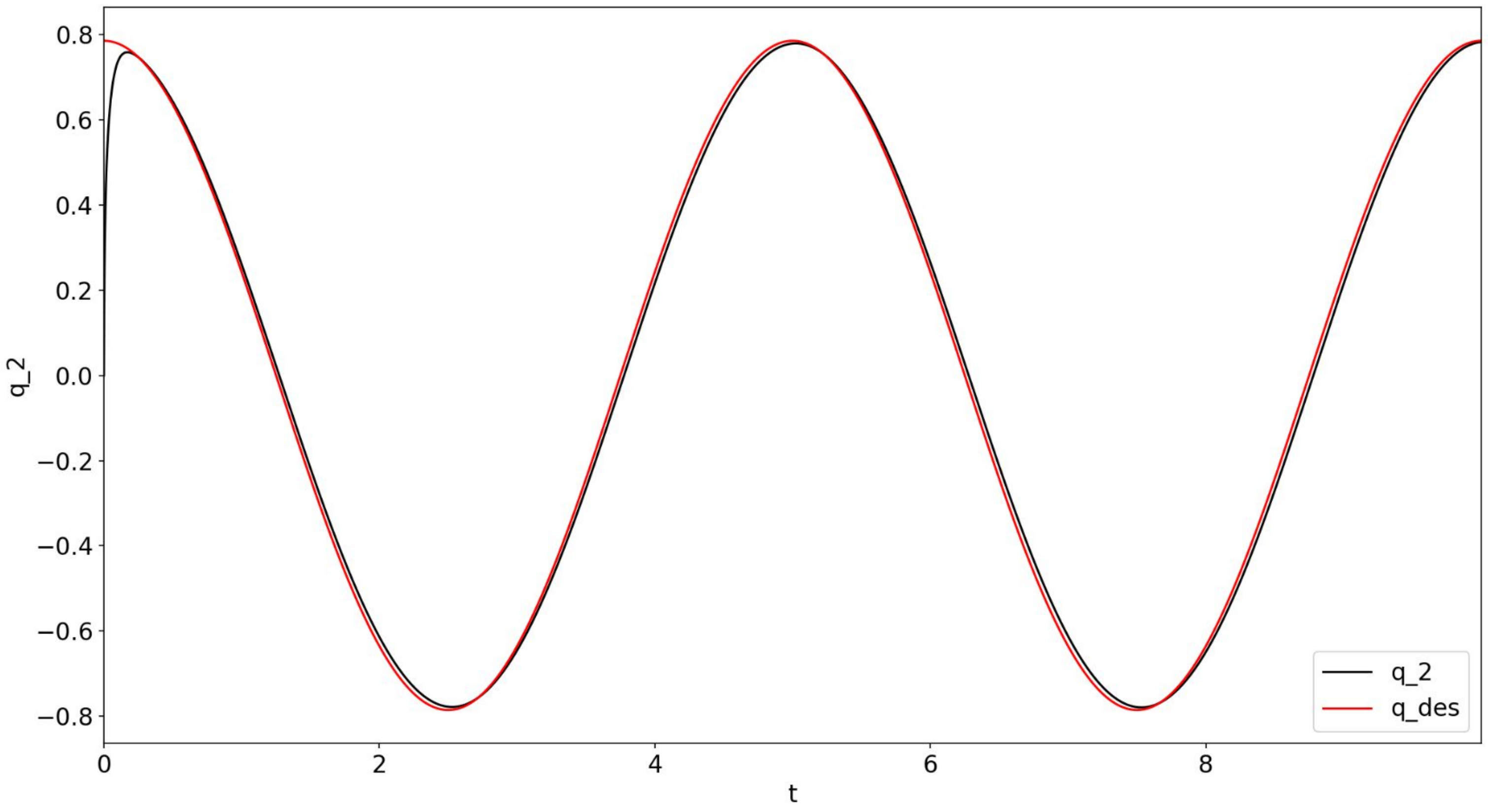
Torque = Kp*(q_{des} - q) - Kd*q_dot + k*q

(contd.)

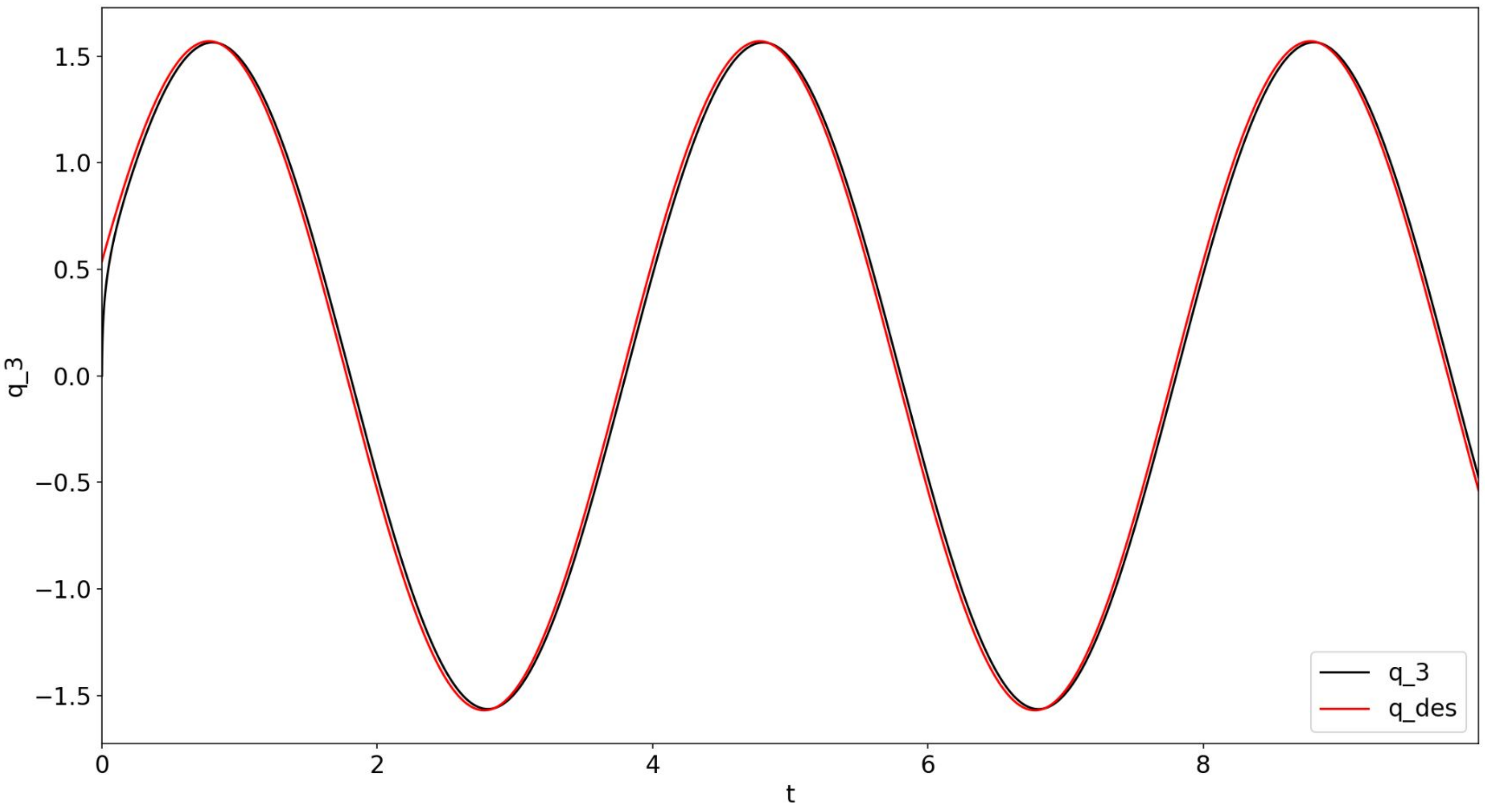


We can see that the max. error is around .0265 radians, i.e., around 1.5^0 which is acceptable.

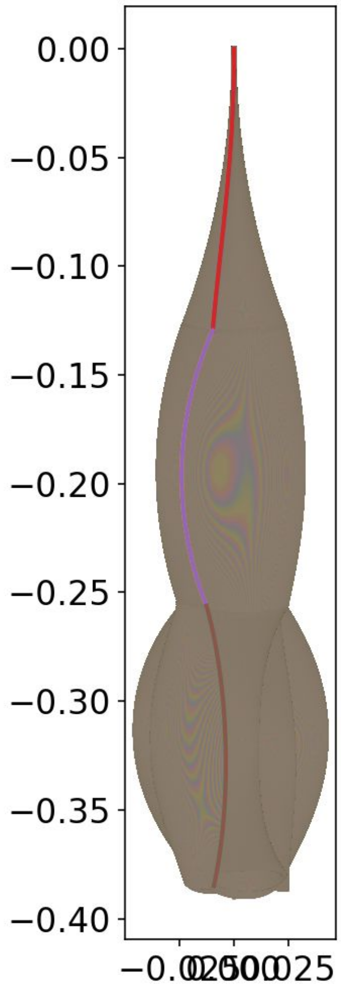
(contd.)



(contd.)



(contd.) Positions

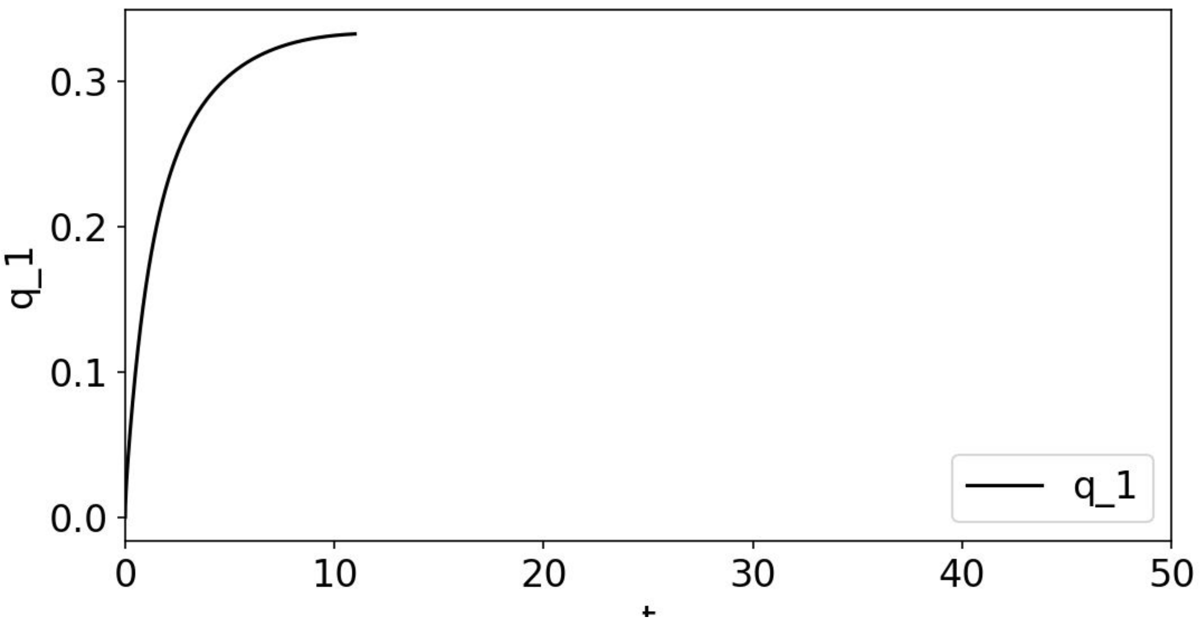


Task Space Control

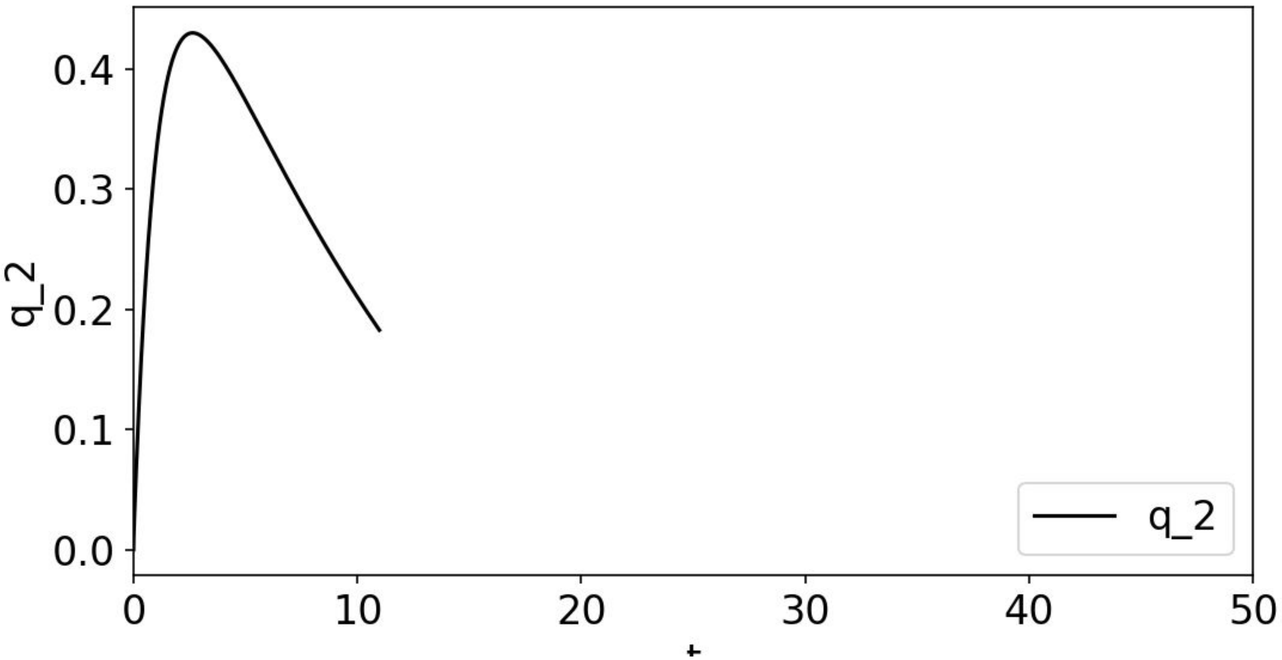
F = Kp*(X_{des} - X) + Kd*(X_{des}_dot- X_dot)

Torque = J^TF

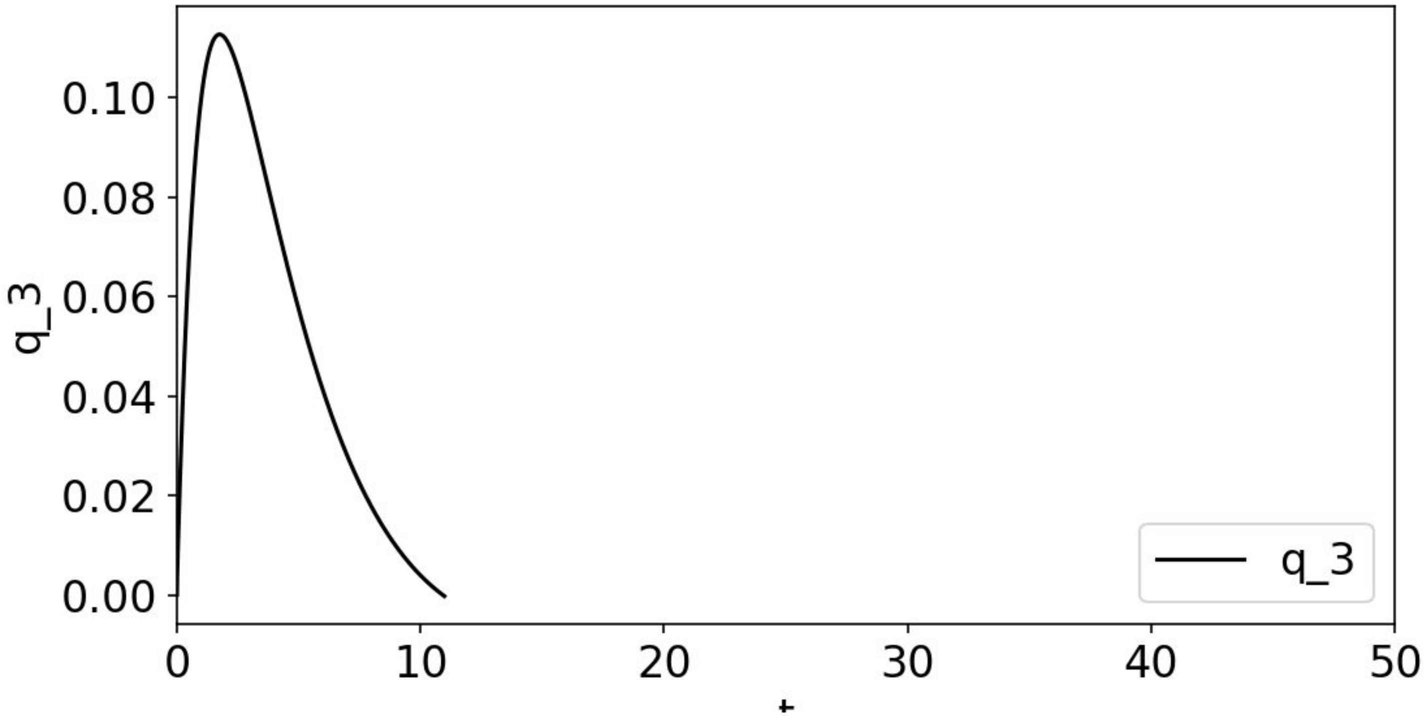
Task Space Control (Kp = 3.3275 and Kd = 0.075, nu=7): Case 1 (Constant torque when q1, q2 or q3 =0)



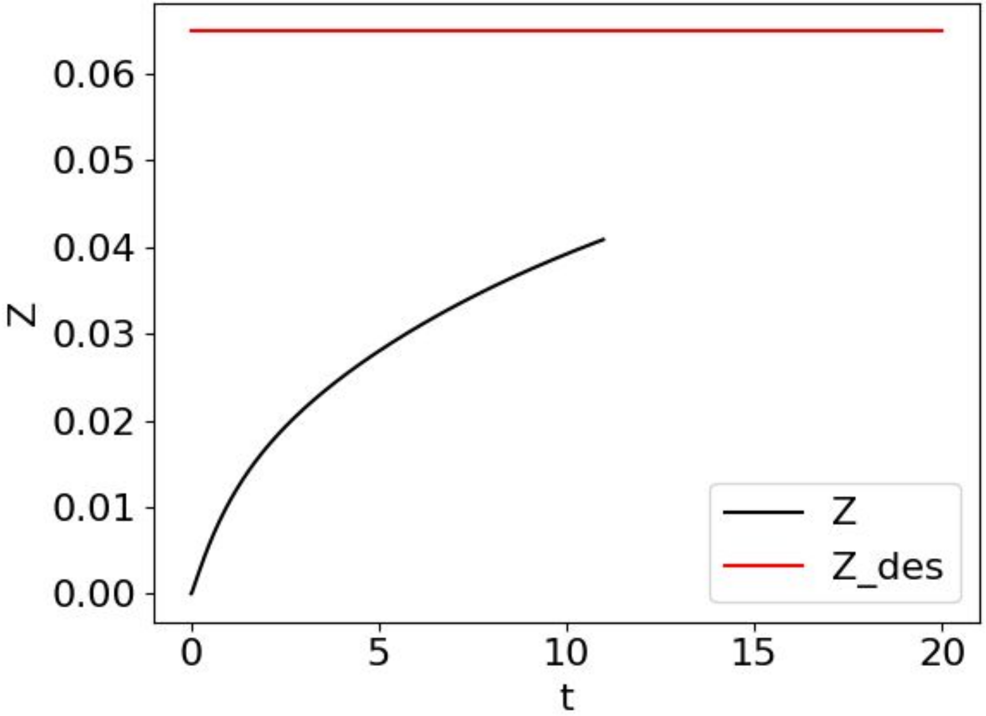
Task Space Control: Case 1 (contd.)



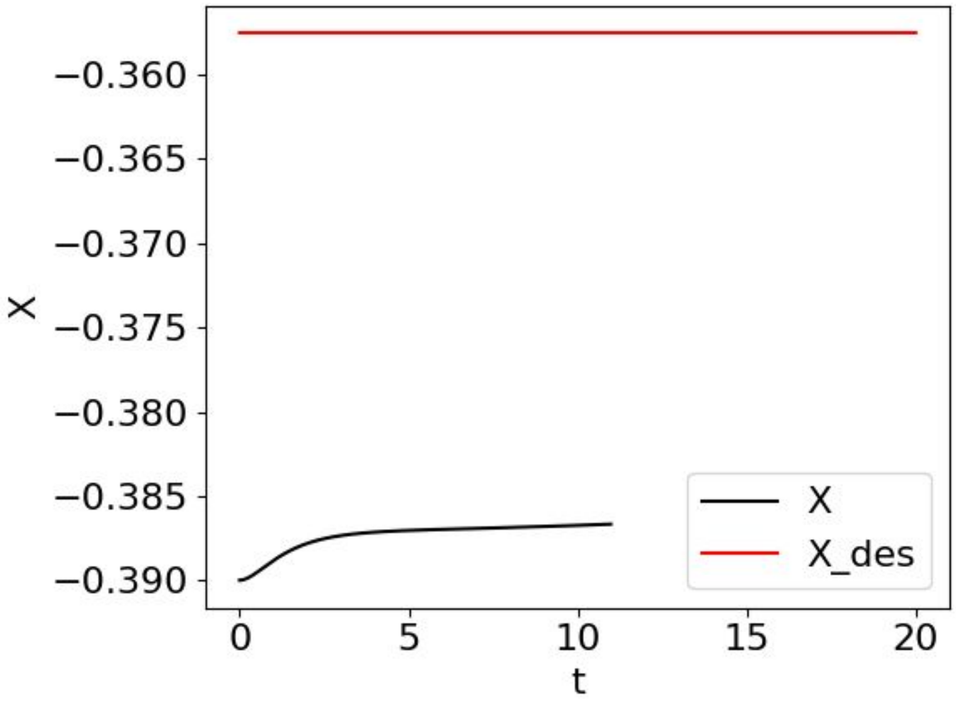
Task Space Control: Case 1 (contd.)



Task Space Control: Case 1 (contd.)



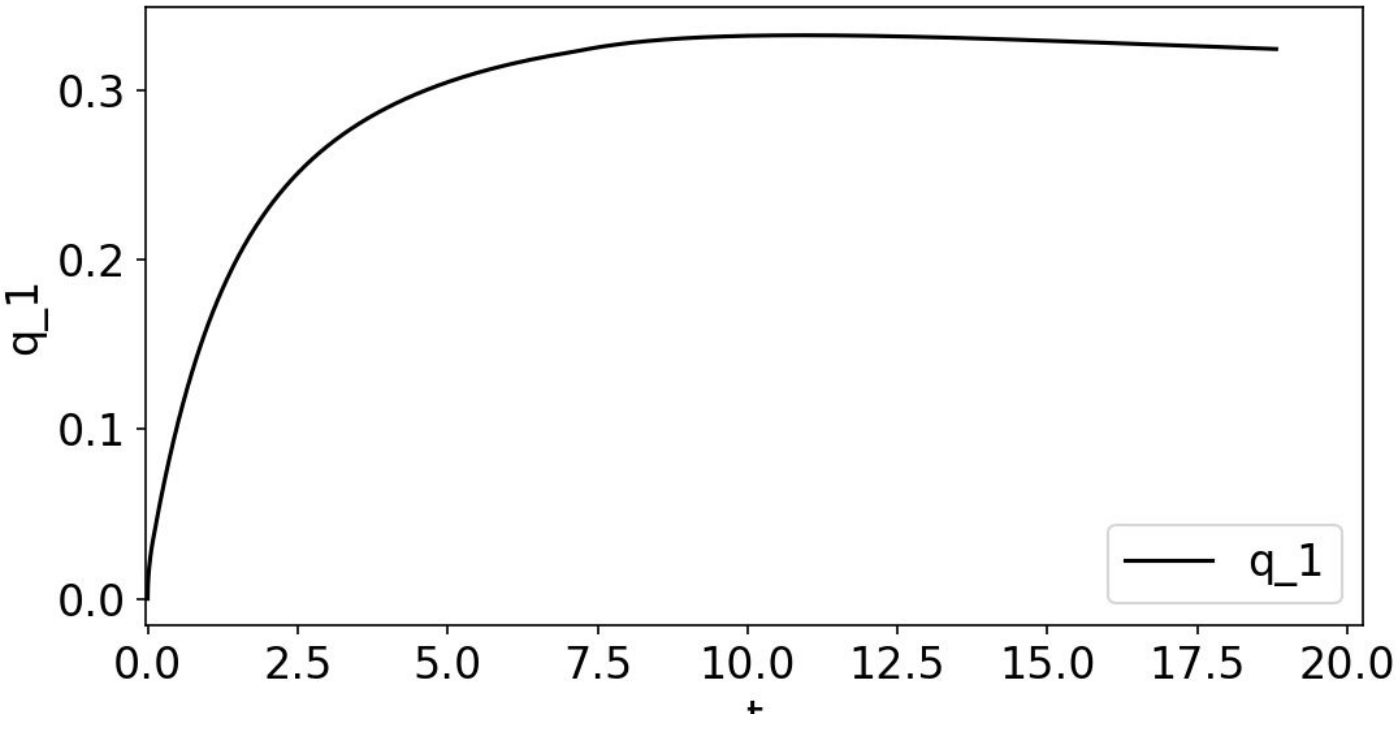
Task Space Control : Case 1 (contd.)



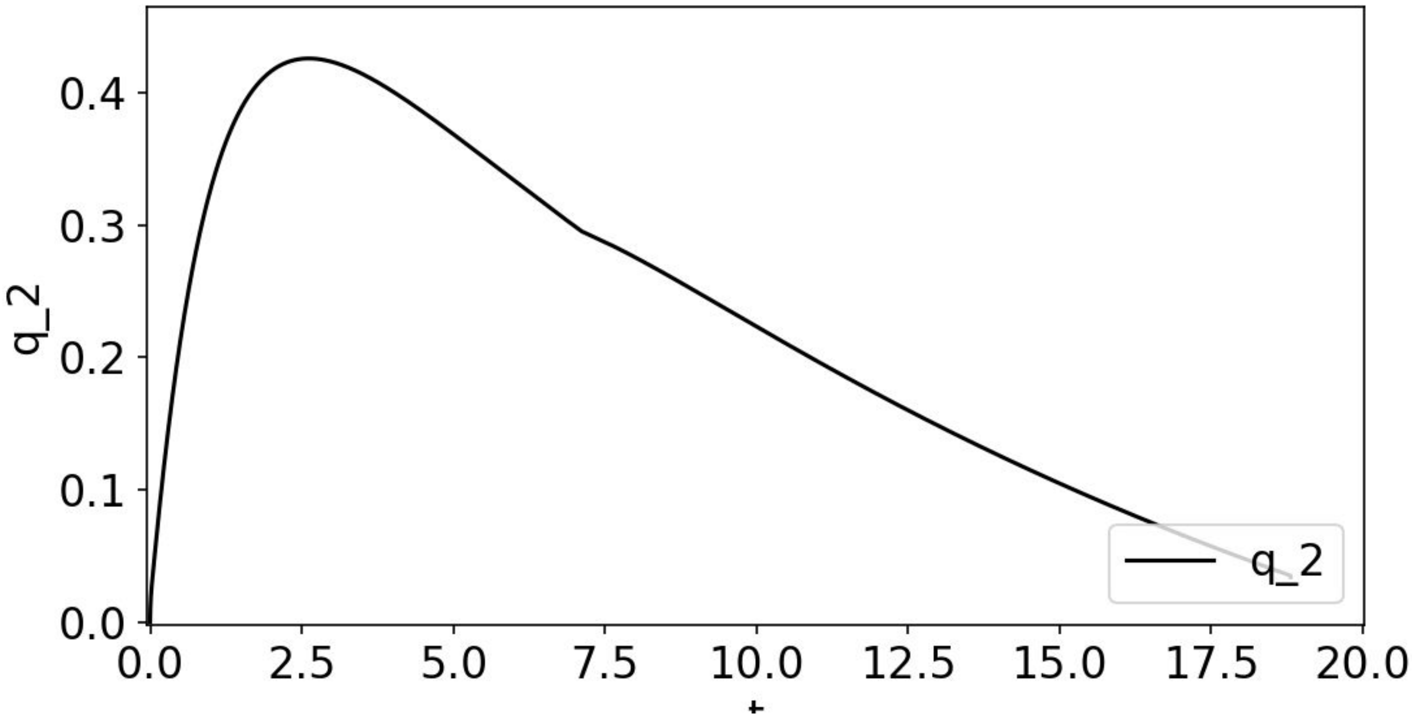
Case 2: Using Margin of 2 degrees

When q is inside the margin, skip to torque at one of the boundaries/margins as per direction of q_dot.

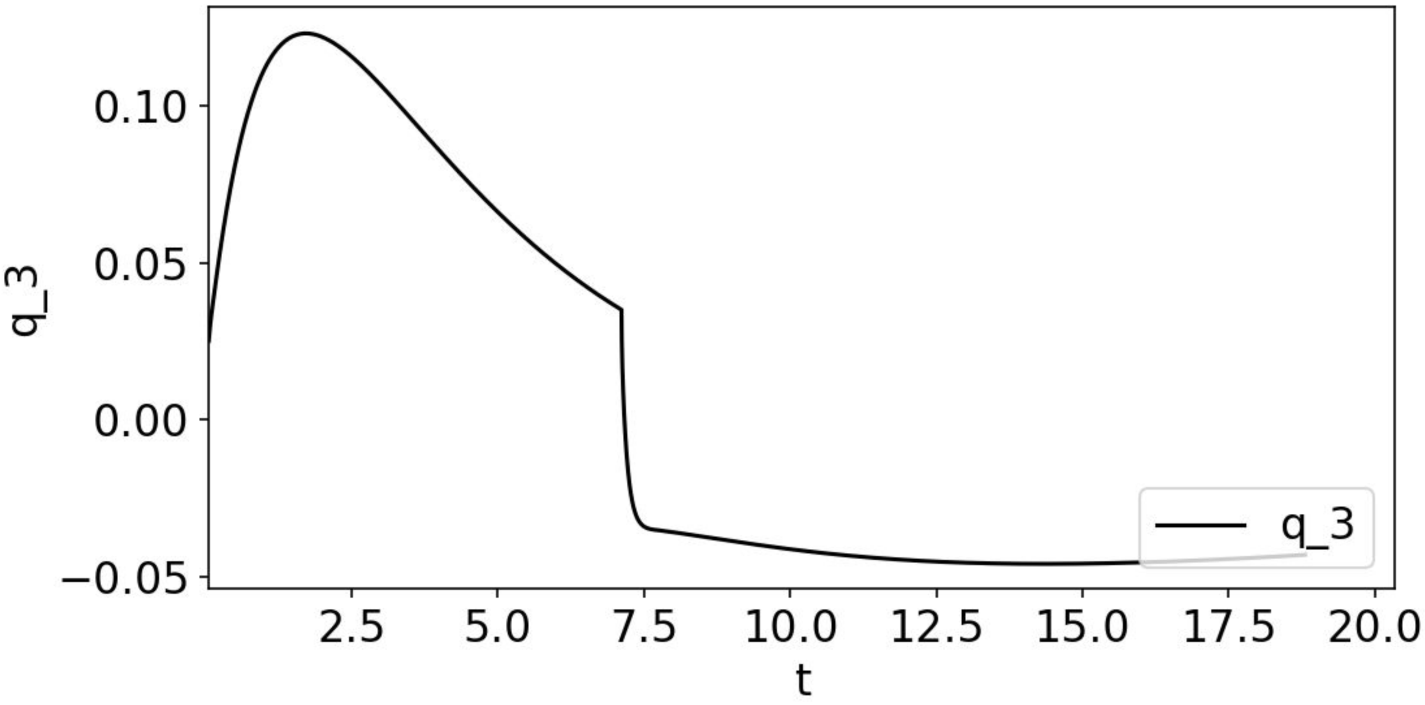
Task Space Control : Case 2 (Using Margin of 2 degrees)



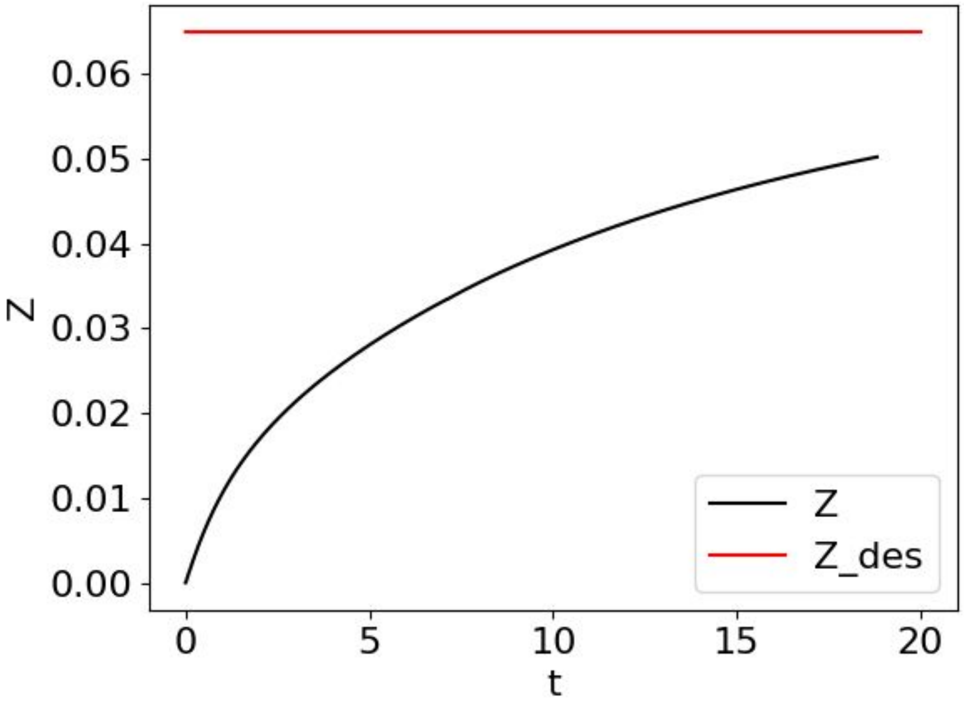
Task Space Control : Case 2 (contd.)



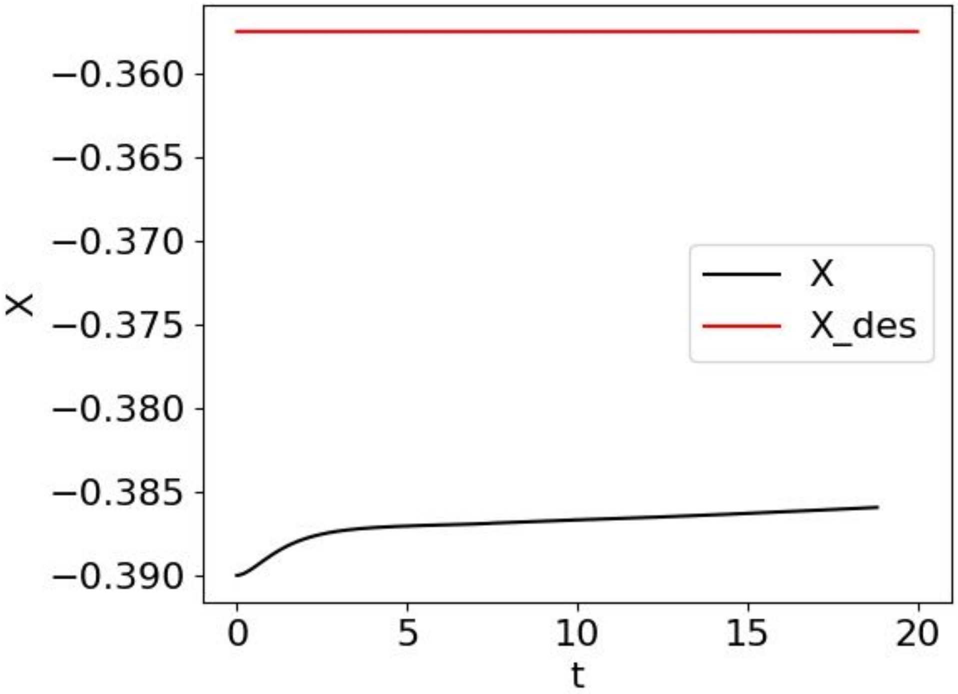
Task Space Control: Case 2 (contd.)



Task Space Control: Case 2 (contd.)



Task Space Control: Case 2 (contd.)



Conclusion

- Joint Space PD Control for Fixed planar case works well with reasonable accuracy.
- Task Space PD Control is not working too well. Giving “Nan”, i.e., “not defined” angle values after a specific amount of time.
- As we can see, around 18 secs, the system goes haywire. Not able to solve this problem yet.
- From q graphs, we can clearly see the skip when in margin of 2 degree around 0. Thus, that logic is working.