

```

1 # %% Peer Graded Assignment C2 Module 3
2 # Goal: modify IKinbody function to report intermediate iterates of the Newton-
  Raphson
3 # Each reported iteration must include:
4 # 1) i = iteration number
5 # 2) Theta = Joint vector
6 # 3) T = End effector configuration
7 # 4) V = Error twist
8 # 5) w = Angular error
9 # 6) v = Linear error
10
11 import numpy as np
12 import modern_robotics as mr
13
14 # -----MR Lib Code Section -----
  ----
15
16 def IKinBodyIterates(Blist, M, T, thetalist0, eomg, ev):
17
18     thetalist = np.array(thetalist0).copy()
19     i = 0
20
21     maxiterations = 60
22     Vb = mr.se3ToVec(mr.MatrixLog6(np.dot(mr.TransInv(mr.FKinBody(M, Blist,
23                                                         thetalist)), T)))
24     err = np.linalg.norm([Vb[0], Vb[1], Vb[2]]) > eomg \
25         or np.linalg.norm([Vb[3], Vb[4], Vb[5]]) > ev
26     log_lines = [] # adding empty list
27     theta_history = [] # adding empty list
28     while err and i < maxiterations:
29         thetalist = thetalist \
30             + np.dot(np.linalg.pinv(mr.JacobianBody(Blist,
31                                                         thetalist)), Vb)
32         i = i + 1
33         Vb \
34             = mr.se3ToVec(mr.MatrixLog6(np.dot(mr.TransInv(mr.FKinBody(M, Blist,
35                                                         thetalist)), T)))
36         err = np.linalg.norm([Vb[0], Vb[1], Vb[2]]) > eomg \
37             or np.linalg.norm([Vb[3], Vb[4], Vb[5]]) > ev
38
39 # -----MR Lib Code Section -----
  ----
40
41     # End effector config for each iteration
42     T_theta = mr.FKinBody(M, Blist, thetalist)
43
44     # logging iteration results
45     log_lines.append(f"Iteration: {i}")
46     log_lines.append(f"SE(3) end - effector config: {T_theta}")
47     log_lines.append(f"error twist V_b: {Vb}")
48     log_lines.append(
49         f"Angular error magnitude ||omega_b||: {np.linalg.norm([Vb[0], Vb[1],
  Vb[2]])}")
50     log_lines.append(
51         f"Linear error magnitude ||v_b||: {np.linalg.norm([Vb[3], Vb[4], Vb[5]])}")
52     \n")
53
54     # Log theta history as .csvs
55     theta_history.append(thetalist)

```

```
56 # Save iteration results as .txt
57 f = open("ik_log.txt", "w")
58 for line in log_lines:
59     f.write(line + "\n")
60 f.close()
61
62 # Save theta history as .csv
63 import csv
64 f = open("theta_history.csv", "w")
65 for row in theta_history:
66     line = ",".join(str(value) for value in row)
67     f.write(line + "\n")
68 f.close()
69
70 return ("theta_history.csv", not err, "ik_log.txt")
71
72 # %%
73
```