Master's degree in Computer Engineering for Robotics and Smart Industry

Advanced control systems

Course assignments

Enrico Bonoldi - VR502852

February 2025

Contents

1 Introduction

This techical report is about the assignments of the **Advanced control systems** course.
The code is available at Github.

2 Robot structure and kinematics

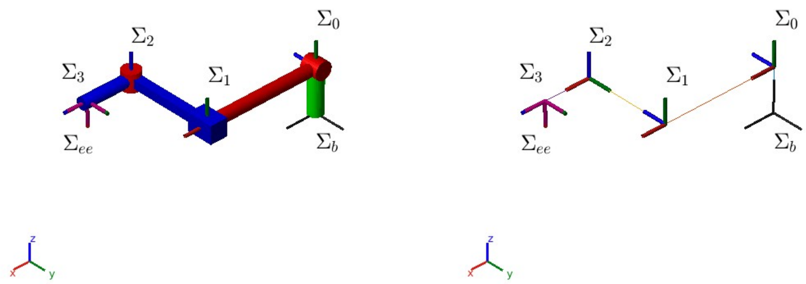


Figure 1: Robot visualization

$$\delta_{b-0} = 0.15 \tag{1}$$
$$\delta_{0-1} = 0.4 \tag{2}$$
$$\delta_{1-2} = 0.3 \tag{3}$$
$$\delta_{2-3} = 0.16 \tag{4}$$

2.1 DH table

a	α	d	θ
0	$\frac{\pi}{2}$	δ_{b-0}	0
δ_{0-1}	0	0	q_1
0	$-\frac{\pi}{2}$	$\delta_{1-2} + q_2$	0
δ_{2-3}	0	0	q_3

To have the same alignment at the end-effector as the one in from the robotic toolbox we must rotate around the y axis (of the frame σ_3) by $\frac{\pi}{2}$ rad.

2.2 Direct kinematics

$$T_i^{i-1} = \begin{bmatrix} \cos(\theta) & -\sin(\theta)\cos(\alpha) & \sin(\theta)\sin(\alpha) & a\cos(\theta) \\ \sin(\theta) & \cos(\theta)\cos(\alpha) & -\cos(\theta)\sin(\alpha) & a\sin(\theta) \\ 0 & \sin(\alpha) & \cos(\alpha) & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

2.3 Inverse kinematics

A closed for solution of the inverse kinematics problem can be found by considering only the cartesian coordinates of the end-effector (since we have only 3 DoF).

$$x_{ee} = 0.4\,\cos(q_1) + 0.16\,\cos(q_1)\cos(q_3) \tag{6}$$
$$y_{ee} = 0.16\,\sin(q_3) - q_2 - 0.3 \tag{7}$$
$$z_{ee} = 0.4\,\sin(q_1) + 0.16\,\cos(q_3)\sin(q_1) + 0.15 \tag{8}$$

TODO: calculations?

$$q_1 = \arctan\left(\frac{z_{ee} - 0.15}{x_{ee}}\right) \tag{9}$$
$$q_3 = \arccos\left(\frac{z_{ee} - 0.15 - 0.4\,\sin q_1}{0.16\,\sin(q_1)}\right) \tag{10}$$
$$q_2 = 0.16\,\sin(q_3) - 0.3 - y_{ee} \tag{11}$$

Figure(2) shows the case in which multiple valid solutions exists.

2.4 Jacobian matrices

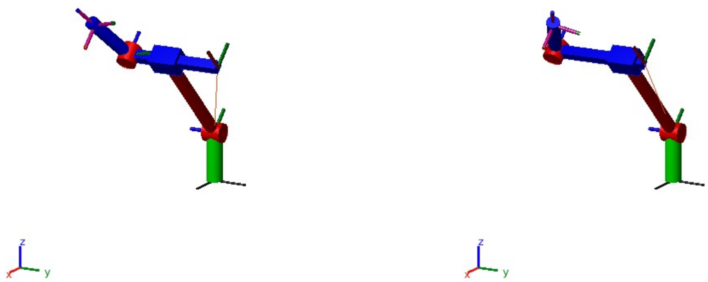
The Euler angles sequence chosen for the analytical Jacobian matrix is $\boldsymbol{ZYZ}$.

3 Manipulator dynamics

3.1 Potential energy

$$\mathcal{U}(q) = g\sin(q_1) + 0.7848\cos(q_3)\sin(q_1) + 4.4145 \tag{12}$$

Where g is the gravitational acceleration.



(a) Robotics system toolbox

(b) Inverse kinematics using equations (9)(10)(11)

Figure 2: Inverse kinematics solutions

3.2 Kinetic energy

$$
\begin{aligned}
\mathcal{T}(q,\dot{q}) = {} & 0.0063\,\dot{q_1}^2\,\cos(q_3)^2 + 0.0320\,\dot{q_1}^2\,\cos(q_3) + 0.2653\,\dot{q_1}^2 + \dot{q_2}^2 \\
& - 0.0800\,\dot{q_2}\,\dot{q_3}\,\cos(q_3) + 0.0128\,\dot{q_3}^2
\end{aligned}
\tag{13}
$$

3.3 Dynamic model

$$
B(q) = \begin{bmatrix} 0.064\,\cos(q_3) + 0.0126\,\cos(q_3)^2 + 0.5305 & 0 & 0 \\ 0 & 2 & -0.08\,cos(q_3) \\ 0 & -0.08\,cos(q_3) & 0.0256 \end{bmatrix}
\tag{14}
$$

$$
C(q,\dot{q}) = \begin{bmatrix} -\dot{q_3}\,(0.0063\,\sin(2\,q_3)) + 0.032\,\sin(q_3) & 0 & -\dot{q_1}\,(0.0063\,\sin(2\,q_3) + 0.032\,\sin(q_3)) \\ 0 & 0 & 0.08\,\dot{q_3}\,\sin(q_3) \\ \dot{q_1}\,(0.0063\,\sin(2\,q_3) + 0.032\,\sin(q_3)) & 0 & 0 \end{bmatrix}
\tag{15}
$$

$$
g(q) = \begin{bmatrix} 0.3924\,\cos(q_1)\,(2\,\cos(q_3) + 25) \\ 0 \\ -0.7848\,\sin(q_1)\,\sin(q_3) \end{bmatrix}
\tag{16}
$$

$$B(q)\,\ddot{q} + C(q,\dot{q})\,\dot{q} + g(q) = \tau \tag{17}$$

Equation(17) is a set of 3 nonlinear second-order differential equations.

3.4 RNE formulation

Using the Newton–Euler equations it is possible to set up an algorithm composed by a **forward recursion** step in which we propagate the links velocity and acceleration from the base link to the EE and a **backward recursion** step in which we propagate the forces from the EE to the base link.

TODO: add B,C*qdot,g matrices from the newton-euler approach

3.5 Dynamic model in the operational space

TODO: add just matrices T and J or all of them? ask the others

3.6 Bonus: Parameters estimation

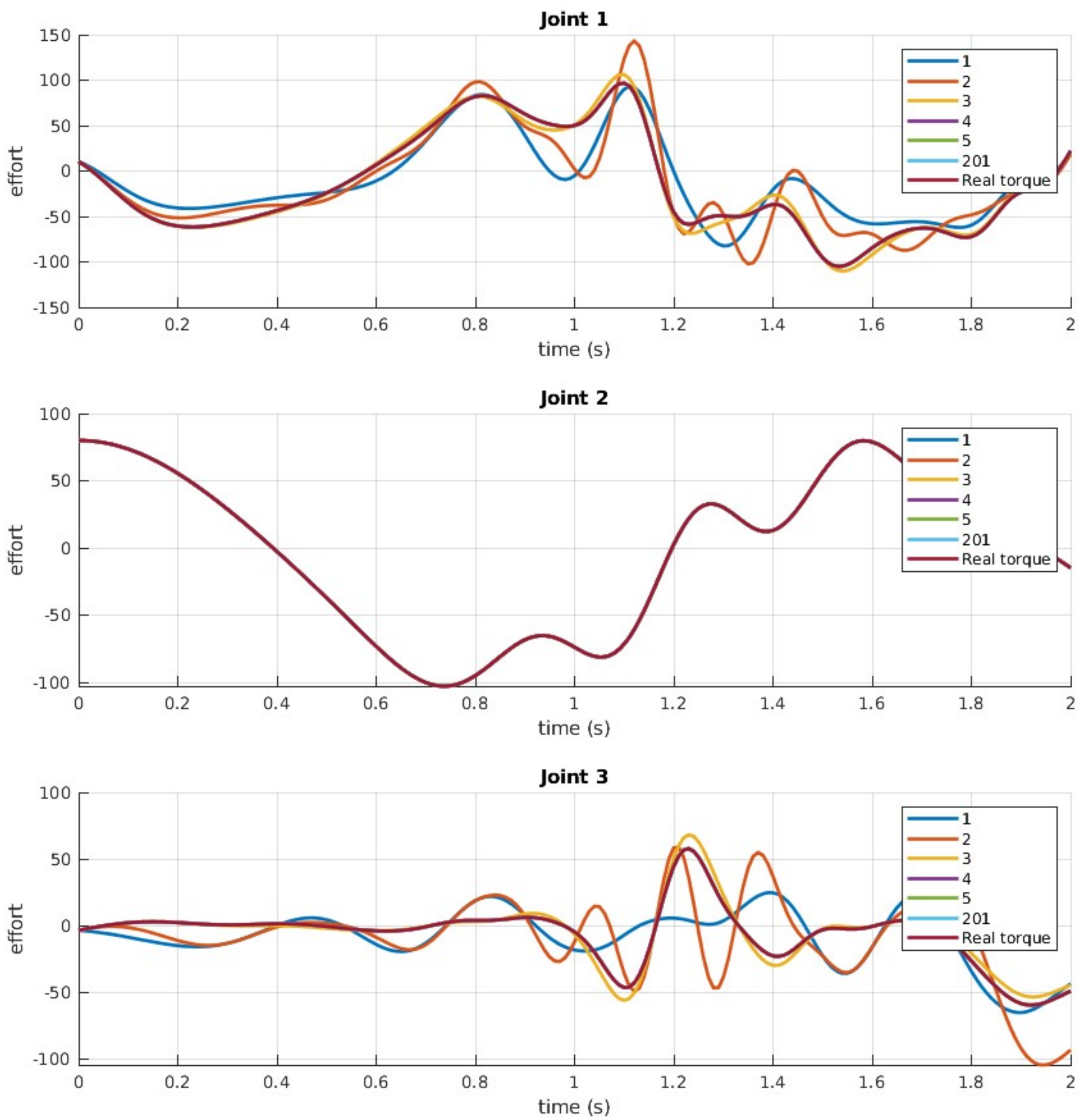


Figure 3: Parameters estimation
different simulation steps amount are taken into account

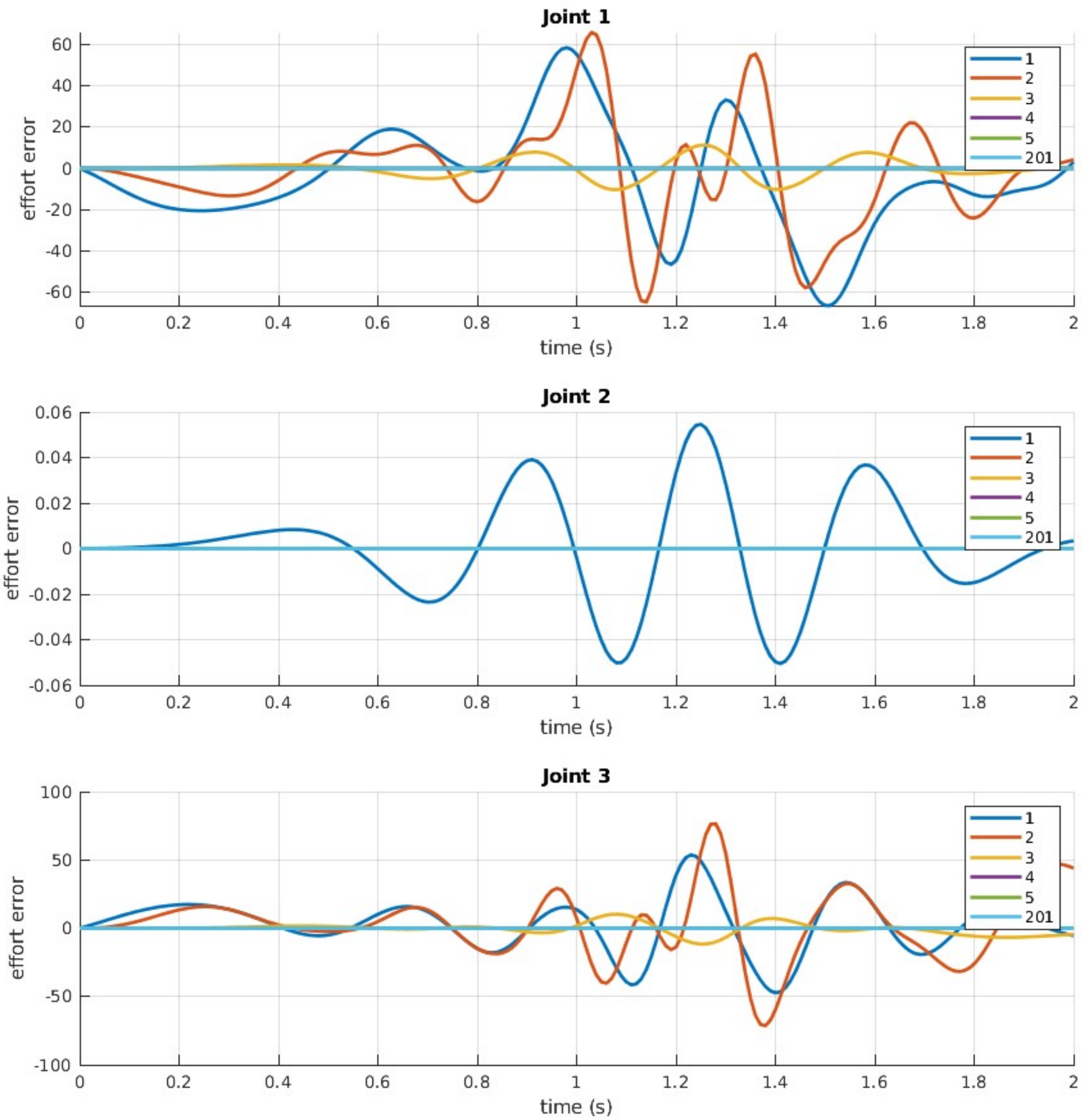


Figure 4: Parameters estimation error
different simulation steps amount are taken into account

Link # \ Property	Mass	Inertia tensor (w.r.t. CoM)
1	1	$\begin{bmatrix} 0.0002 & 0 & 0 \\ 0 & 0.0800 & 0 \\ 0 & 0 & 0.0800 \end{bmatrix}$
2	1	$\begin{bmatrix} 0.0076 & 0 & 0 \\ 0 & 0.0150 & 0 \\ 0 & 0 & 0.0076 \end{bmatrix}$
3	1	$\begin{bmatrix} 0.0002 & 0 & 0 \\ 0 & 0.0128 & 0 \\ 0 & 0 & 0.0128 \end{bmatrix}$

Table 1: Real robot parameters

Link # \ Property	Mass	Inertia tensor (w.r.t. CoM)
1	1	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0.0406 \end{bmatrix}$
2	1	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.0406 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
3	1	$\begin{bmatrix} 0.0140 & 0 & 0 \\ 0 & 0.0266 & 0 \\ 0 & 0 & 0.0128 \end{bmatrix}$

Table 2: Estimated robot parameters

4 Control schemes

4.1 Joint Space PD control law with gravity compensation

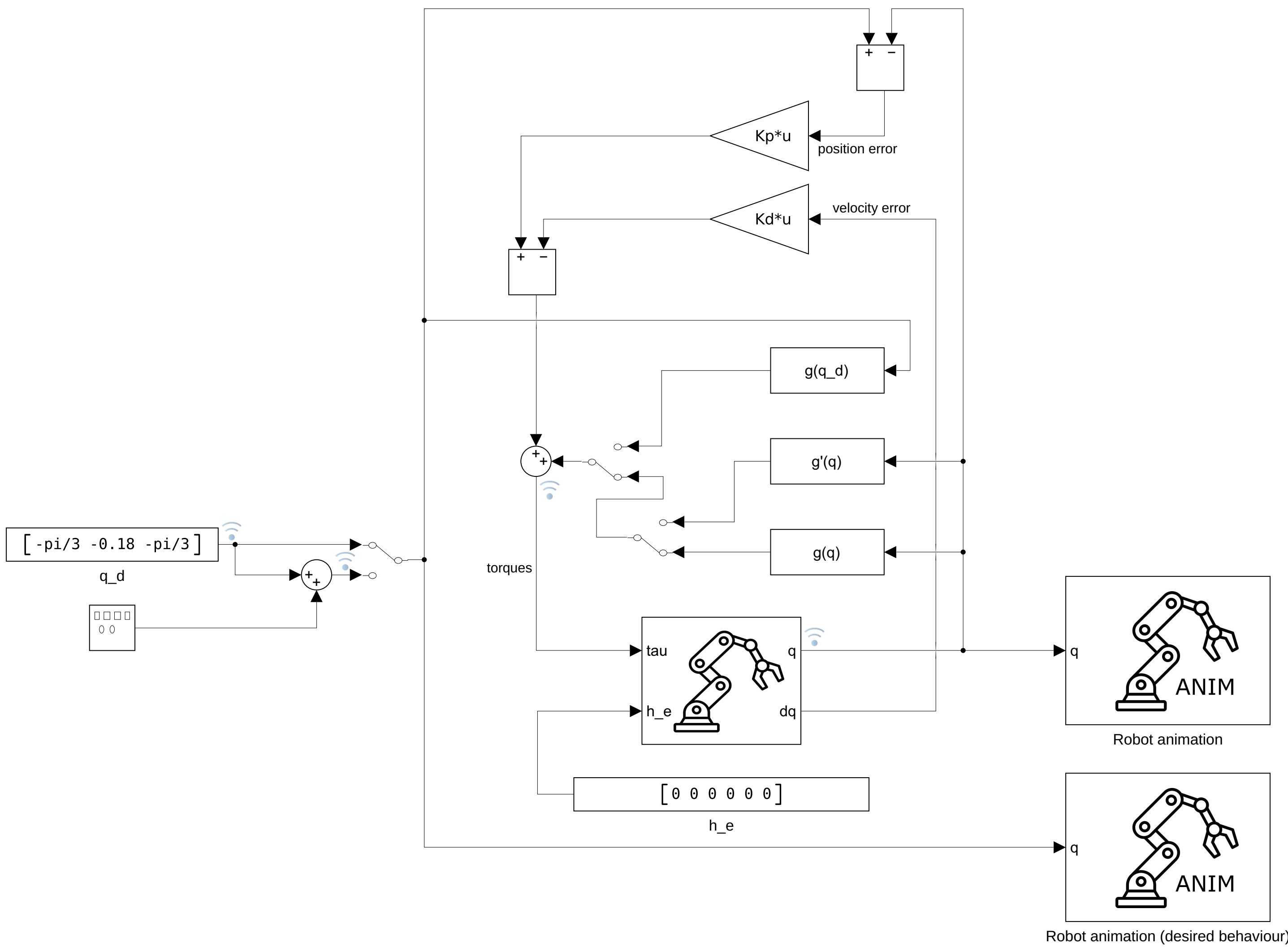


Figure 5: Joint Space PD control law with gravity compensation control scheme

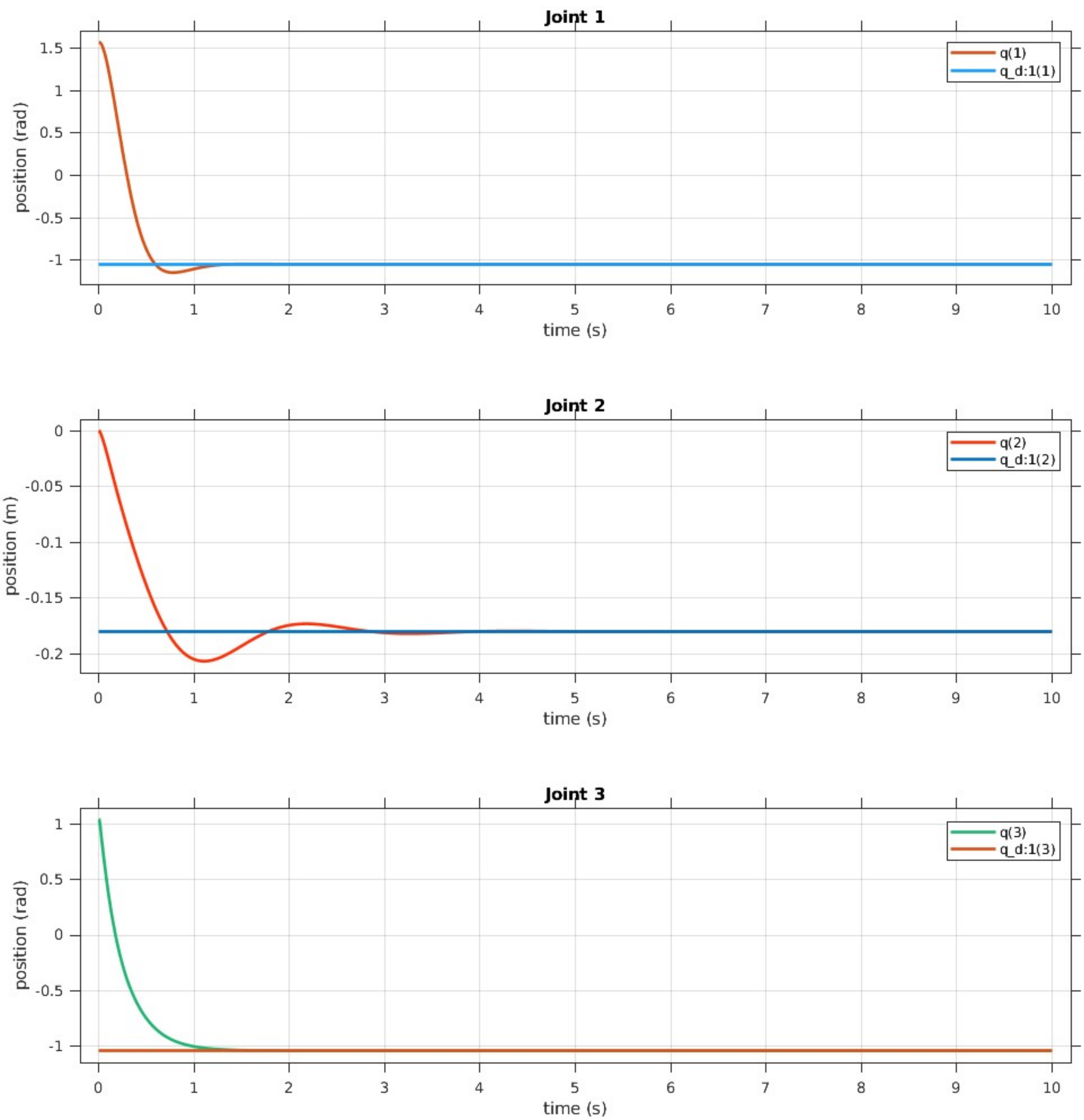


Figure 6: Joint Space PD control law with gravity compensation

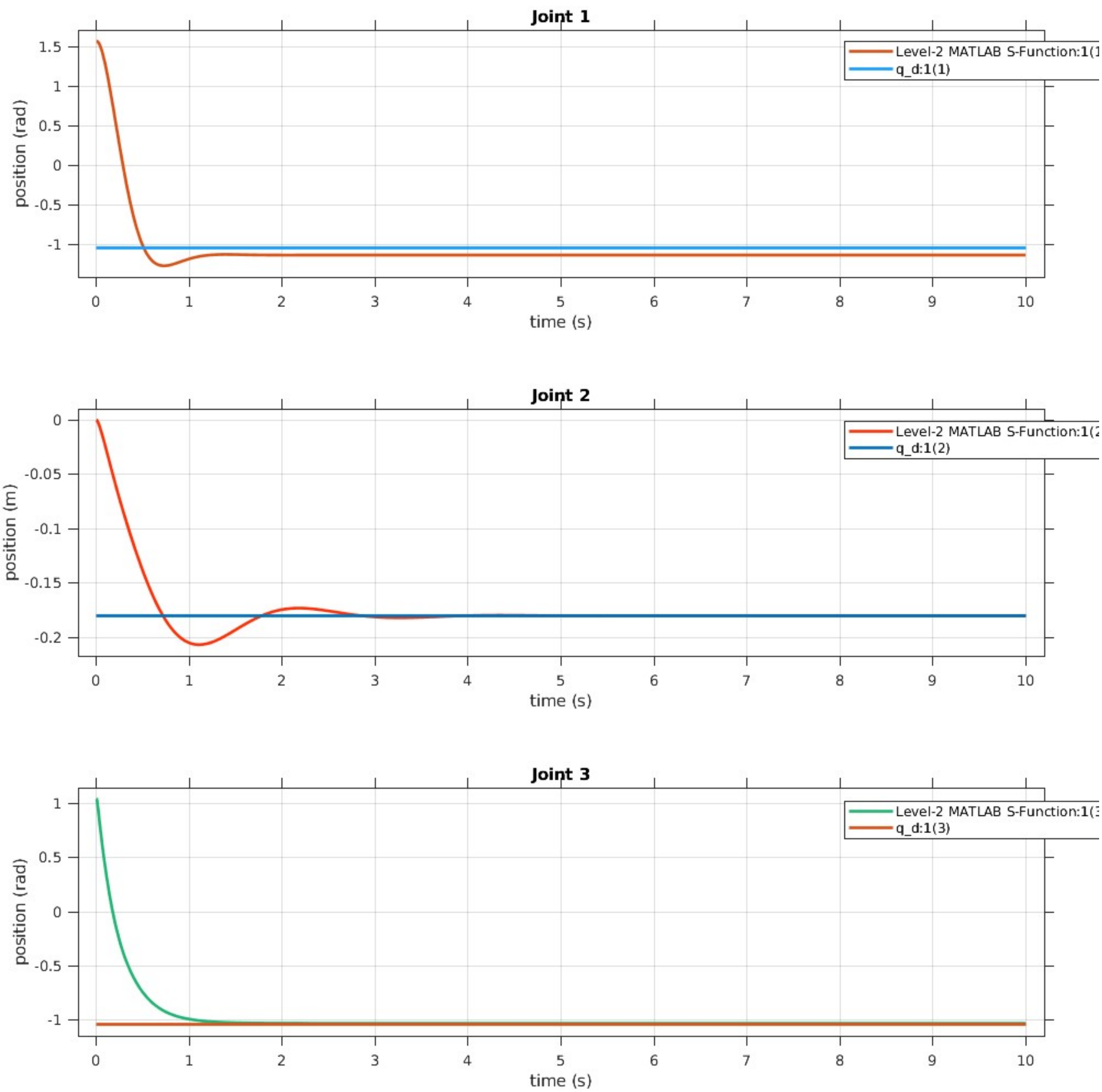


Figure 7: Joint Space PD control law with gravity compensation
compensated gravity term is $g'(q)$

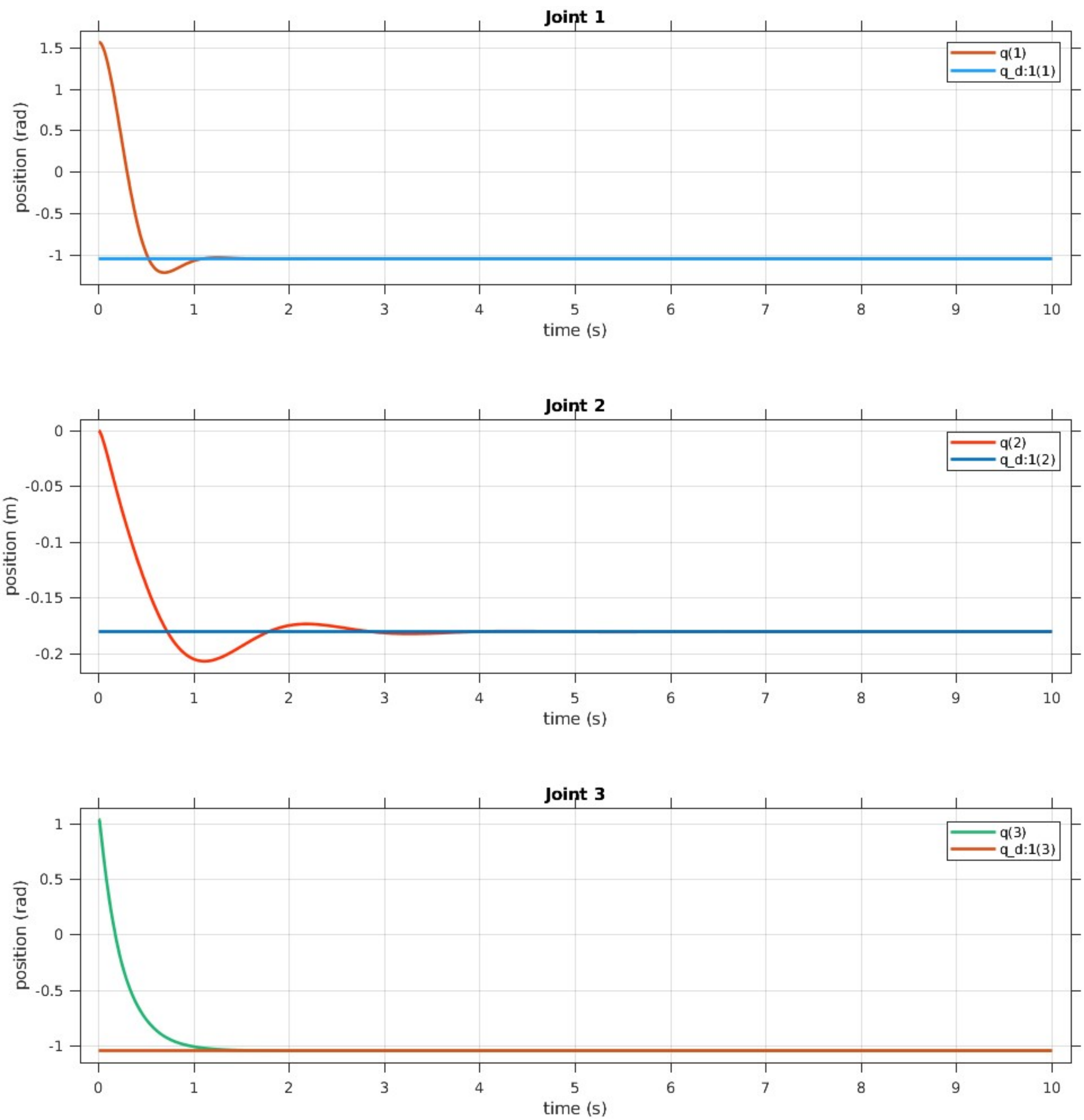


Figure 8: Joint Space PD control law with gravity compensation
compensated gravity term is $g(q_d)$

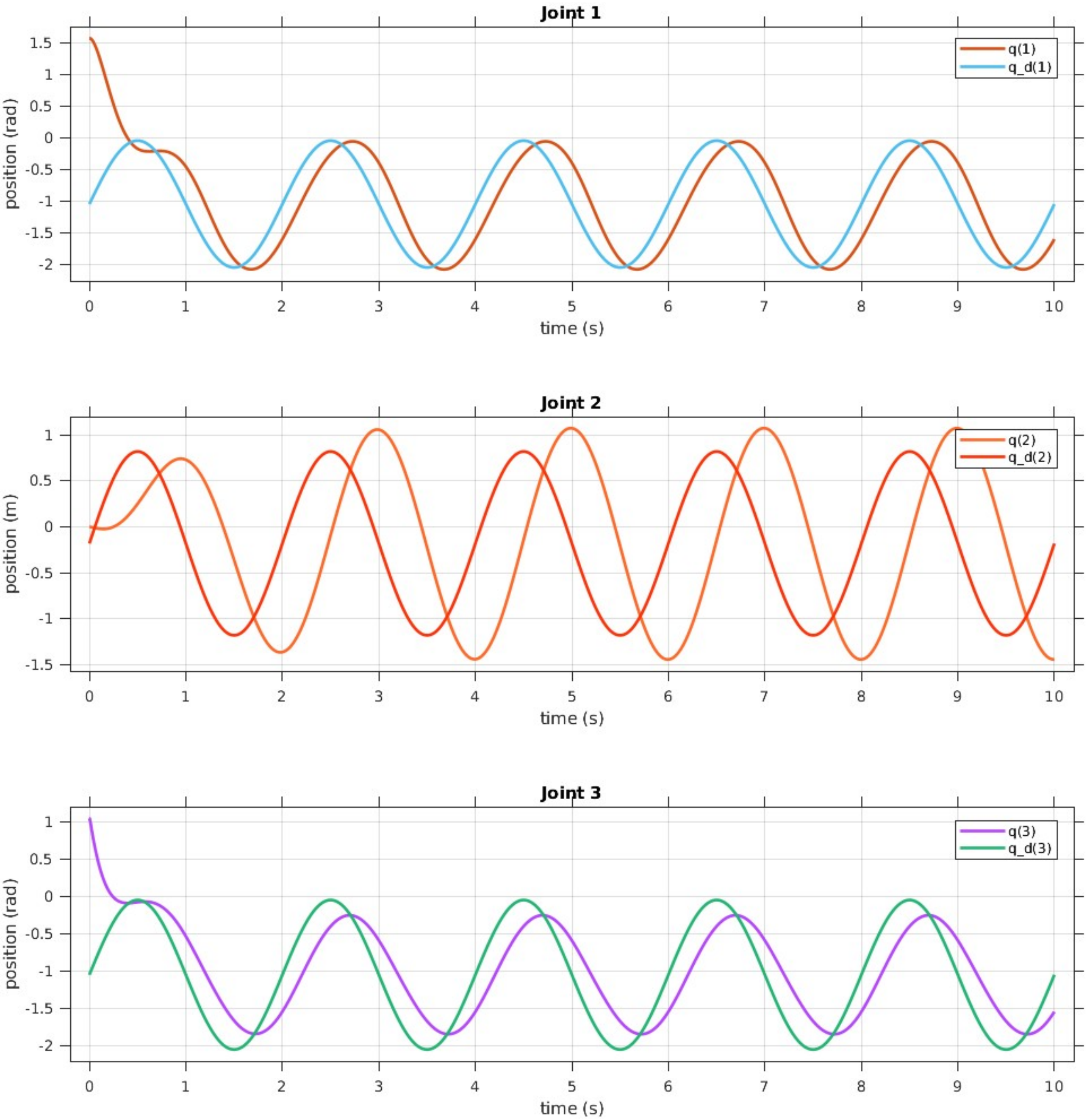


Figure 9: Joint Space PD control law with gravity compensation
tracking task

4.2 Joint Space Inverse Dynamics control law

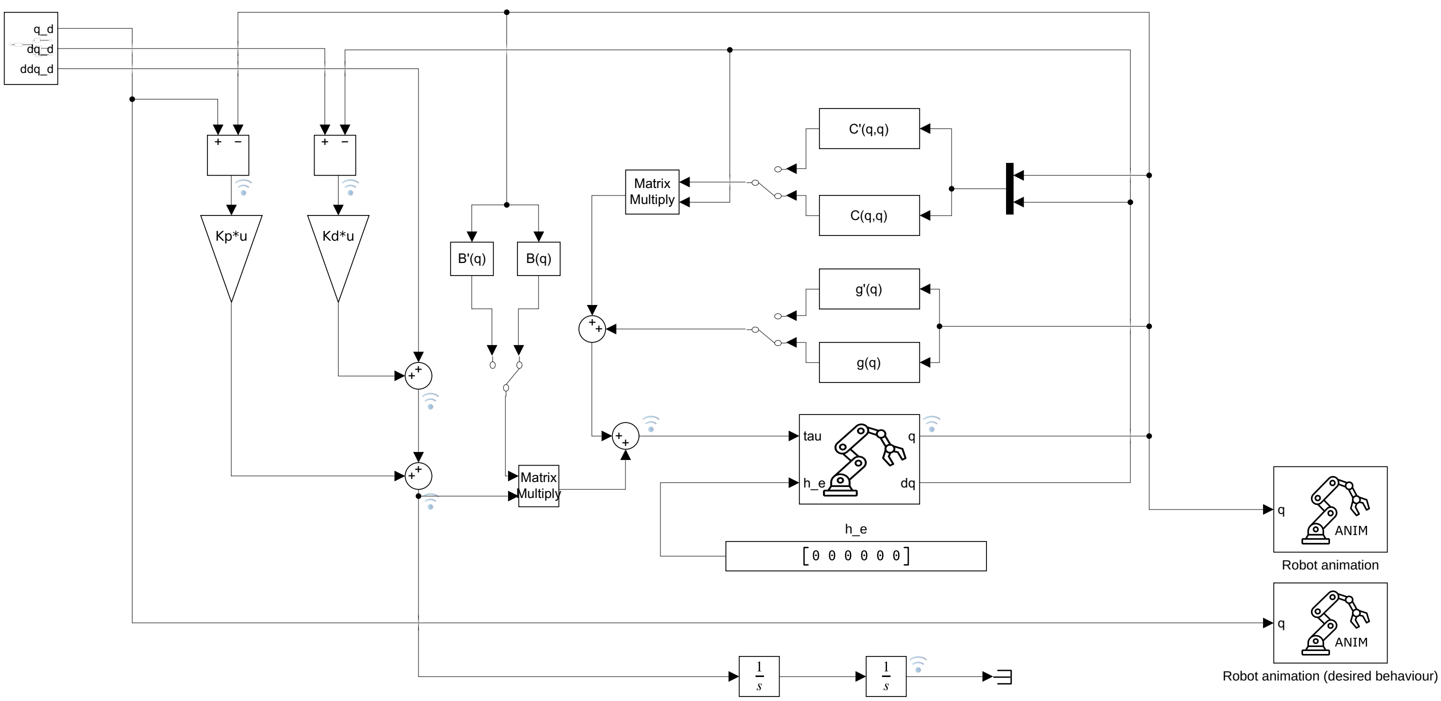


Figure 10: Joint Space Inverse Dynamics control law control scheme

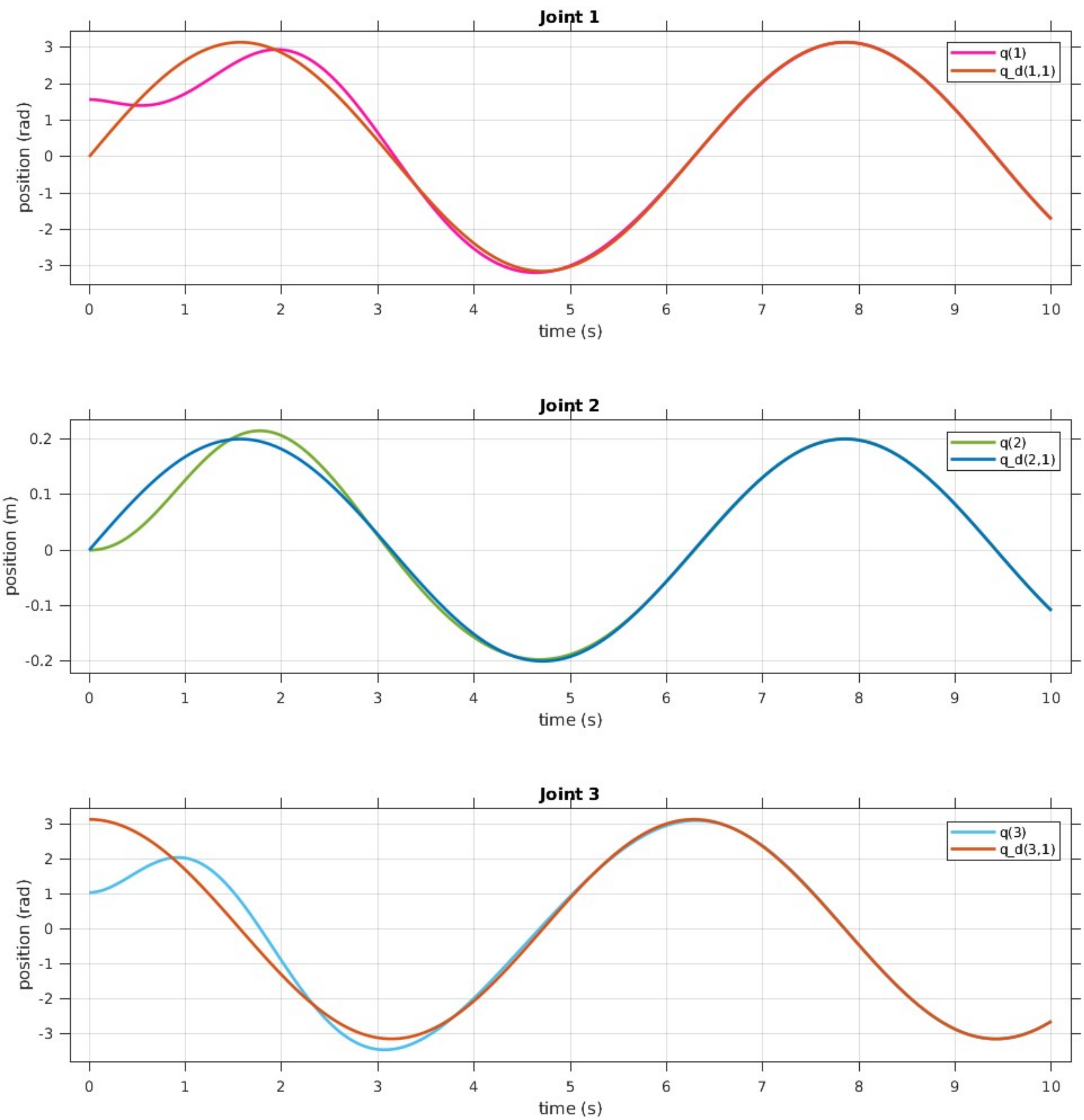


Figure 11: Joint Space Inverse Dynamics control law

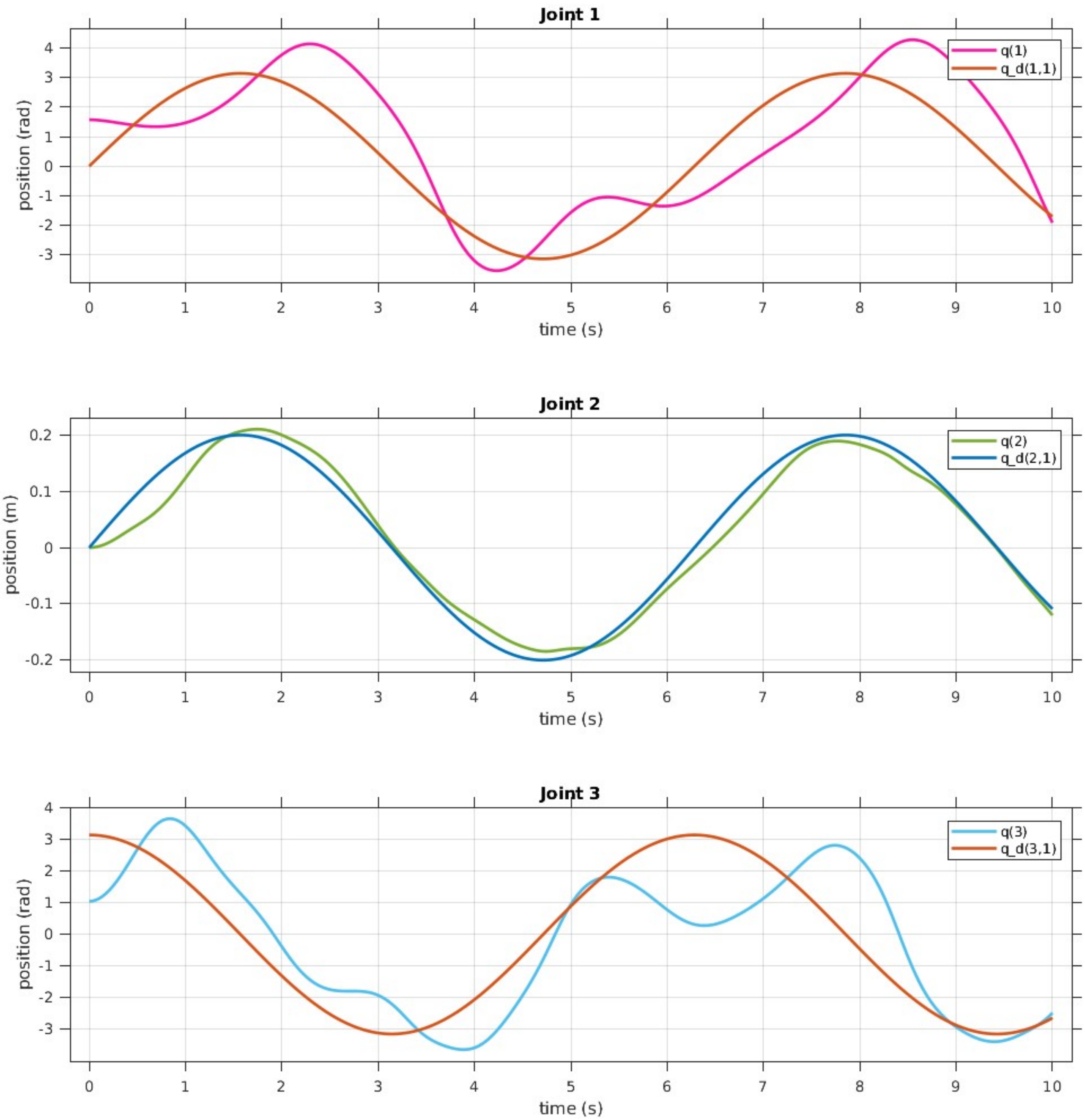


Figure 12: Joint Space Inverse Dynamics control law
dynamics matrices are off

4.3 Operations Space PD control law with gravity compensation

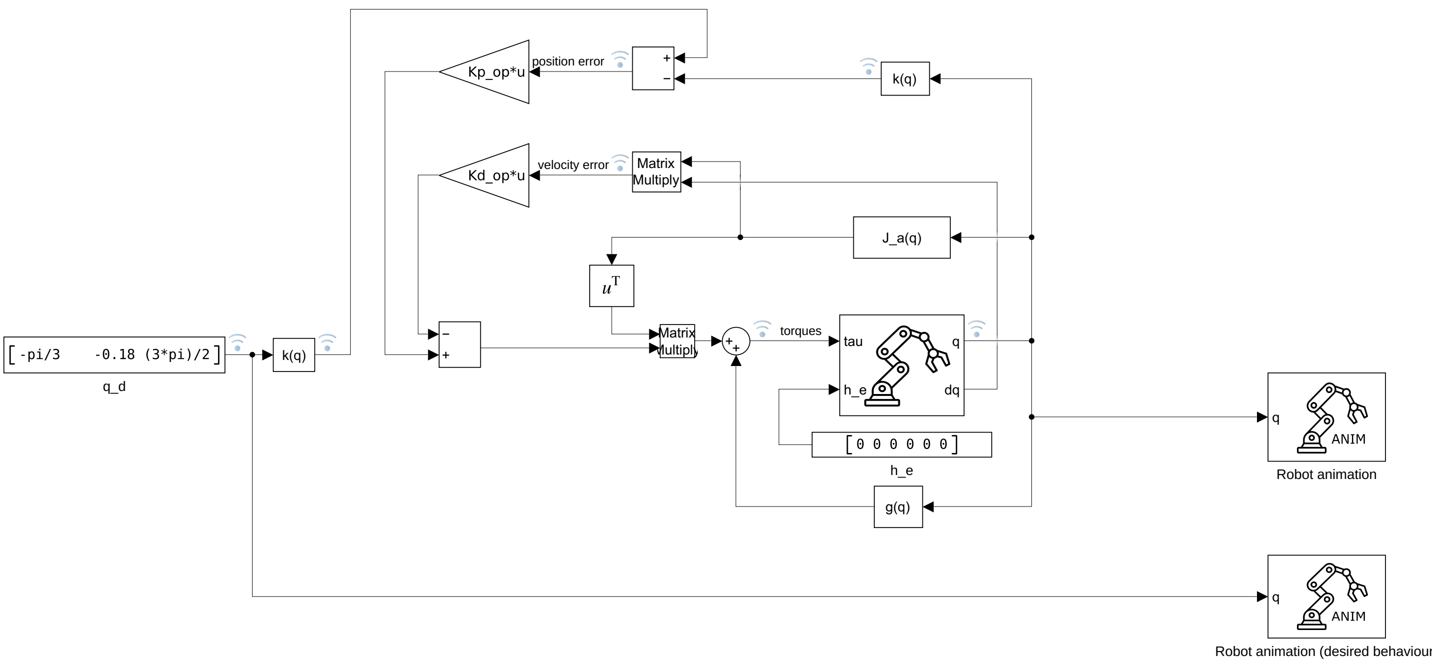


Figure 13: Operational Space PD control law with gravity compensation scheme

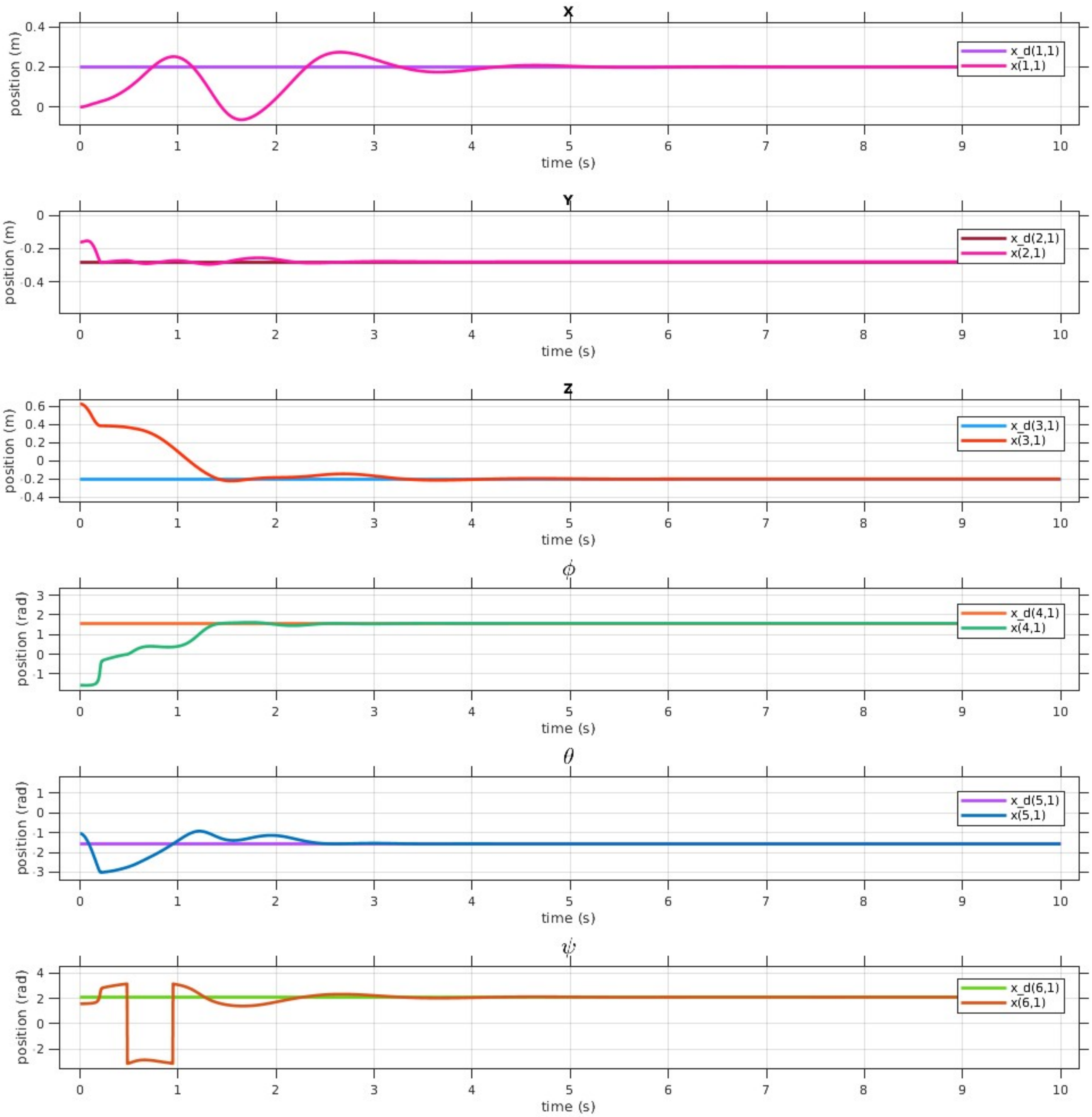


Figure 14: Operational Space PD control law with gravity compensation