V I L L A N O V A U N I V E R S I T Y

ME 8204

Project Report

MODELING AND CONTROL OF LBR iiwa 7 R800

Turki Haj Mohamad

5/10/2018

Introduction

For this project LBR iiwa 7 R800 from KUKA Laboratories was chosen to be studied. The iiwa 7 is an adaptive manipulator that can work in a collaboration with human environment as shown in fig. 1. This light weight robot can be used for handling, machining, and assembly processing. In fig. 2 (a) the main robot assemblies are shown where (1) in-line wrist, (2) joint module and (3) base frame. The iiwa 7 consists of 7 joints (7 degree of freedom) which means it is a redundant robot. The robot axes are shown in Fig.2 (b).

Figure 1: LBR iiwa 7 R800-human collaboration.

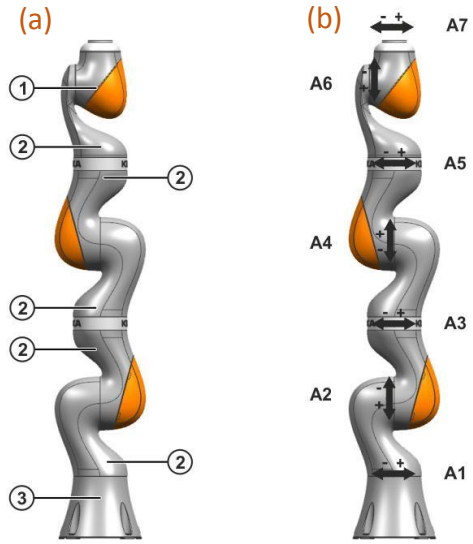


Figure 2: (a) LBR iiwa 7 R800 assemblies. (b) LBR iiwa 7 R800 7 axes.

Table 1 shows some basic data of the iiwa 7 such as the number of axes and controlled axes, the total robot weight, the maximum robot reach. The working envelope of iiwa 7 and the allowed angle ranges for each axis are shown in fig. 3.

Table 1: Technical data of LBR iiwa 7 R800

	LBR iiwa 7 R800
Number of axes	7
Number of controlled axes	7
Volume of working envelope	1.7 m^3
Weight	23.9 kg
Maximum reach	800 mm
Mounting position	Floor, Ceiling, Wall

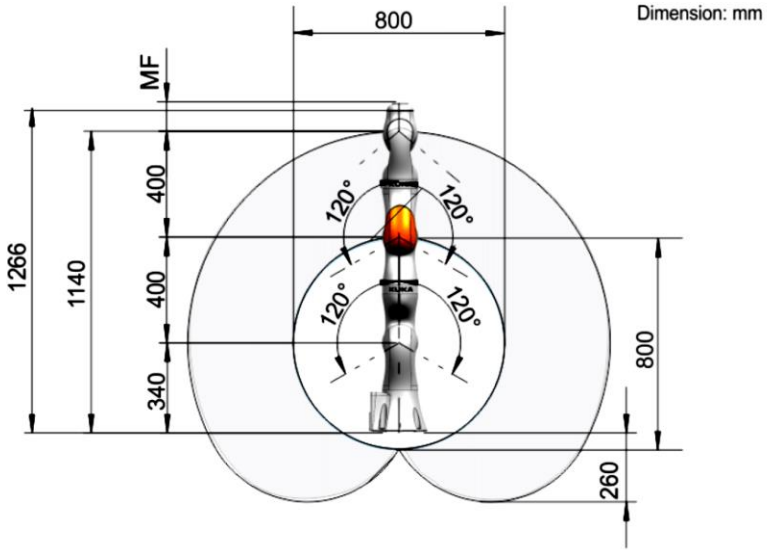


Figure 3: The working envelope of LBR iiwa 7 R800

This project is divided into three parts: 1) Kinematic and dynamic modeling 2) Trajectory planning 3) Control. In the first part, the DH table will be constructed then the transformation matrices will be determined. This step represent the forward kinematic step in which for any set of joint variable the end effector can be determined. The second step is deriving the equation of motion using Euler Lagrangian dynamics. The second stage of the project is trajectory planning. A circular trajectory was selected to be followed by the end effector with a fixed orientation. To do so, the inverse position solution has to be solved to determine the unknown joint variables. Because we have a redundant robot (7 degree of freedom) the pseudo Jacobian matrix have to be derived to solve for the inverse velocity and acceleration. Finally, a sliding controller is used to achieve the desired motion. Figure 4 summarize the project plan.

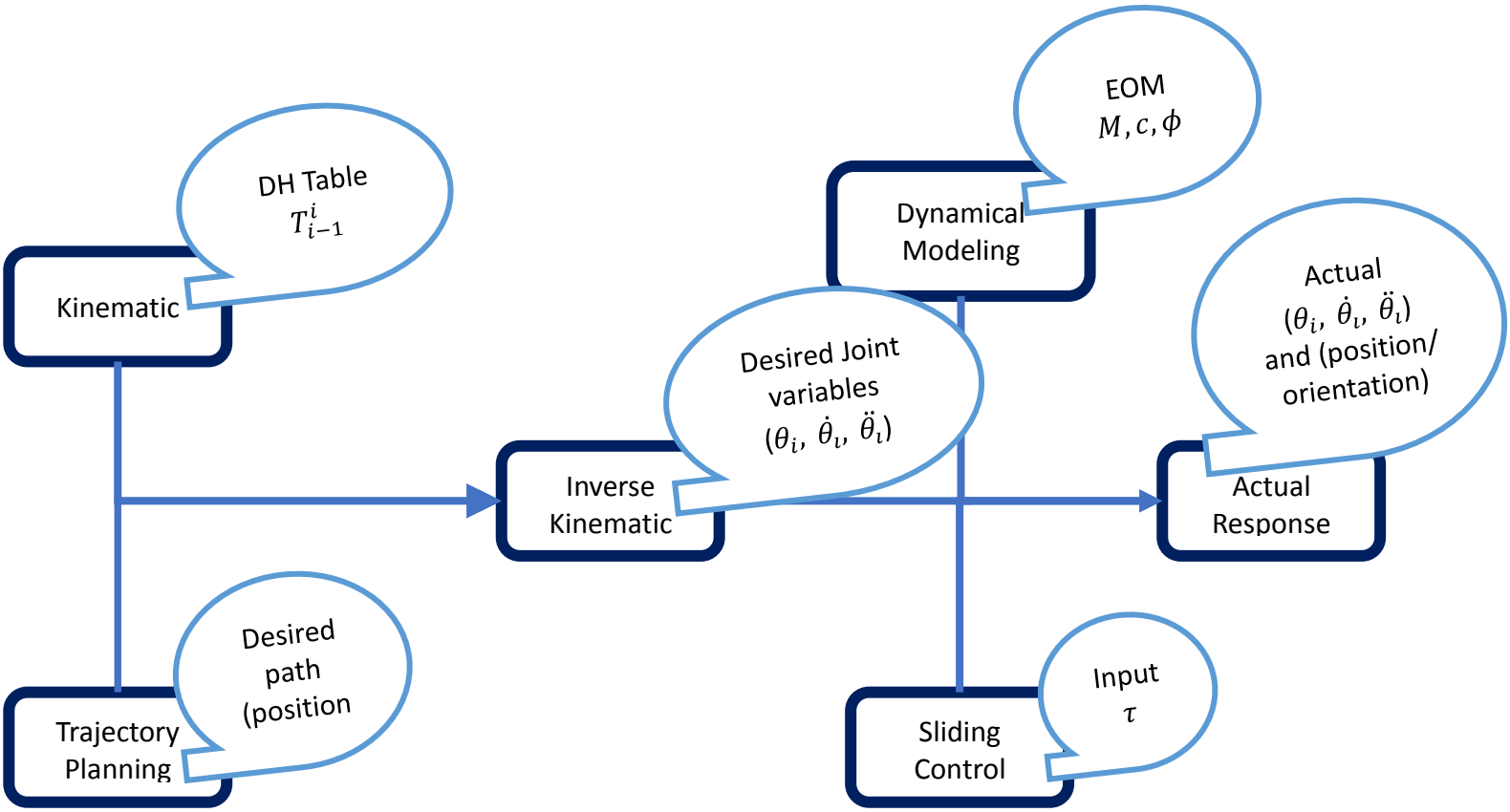


Figure 4: A summary of the project plan

Kinematic Model

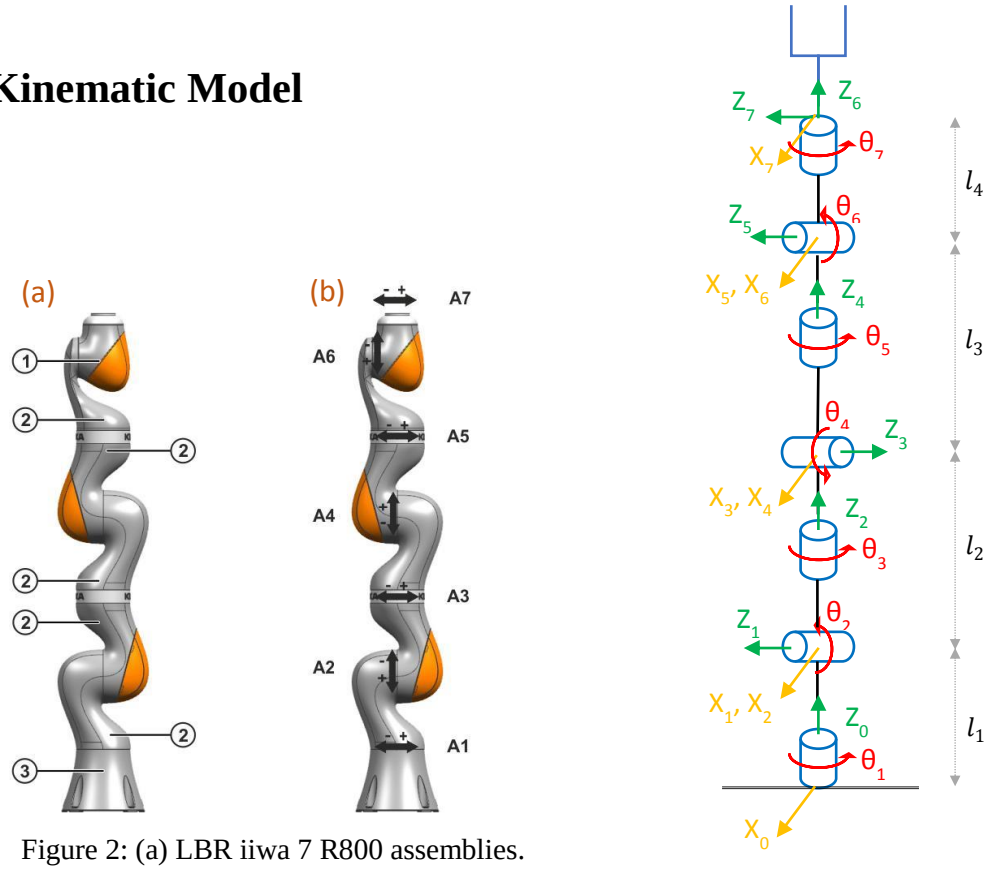


Figure 2: (a) LBR iiwa 7 R800 assemblies.
(b) LBR iiwa 7 R800 7 axes.

Forward Kinematic:

The Denavit-Hartenberg representation method is used in this project to derive the forward kinematics equations for the robot manipulator. This representation is based on describing the transformation relationship between link $(i-1)$ and link i by using four parmaeters θ_i, d_i, a_i and α_i. Table 1 show the DH table for the iiwa 7 robot manipulator.

Table 1: Denavit-Hartenberg representation for the iiwa robot manipulator.

Joint	**θ**	**d**	**a**	**α**
0 – 1	θ_1	l_1	0	$\pi/2$
1 – 2	θ_2	0	0	$-\pi/2$
2 – 3	θ_3	l_2	0	$-\pi/2$
3 – 4	θ_4	0	0	$\pi/2$
4 – 5	θ_5	l_3	0	$\pi/2$
5 – 6	θ_6	0	0	$-\pi/2$
6 – 7	θ_7	l_4	0	$\pi/2$

Transformation matrices between each link can be derived using the following relationship:

$$T_{n-1}^{n} = \begin{bmatrix} \cos\theta_n & -sin\theta_n cos\alpha_n & sin\theta_n sin\alpha_n & a_n cos\theta_n \\ sin\theta_n & \cos\theta_n cos\alpha_n & -\cos\theta_n sin\alpha_n & a_n sin\theta_n \\ 0 & sin\alpha_n & cos\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Using the above formula, the following transformation matrices are found:

$$T_0^1 = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_0^2 = \begin{bmatrix} c_1c_2 & -s_1 & -c_1s_2 & 0 \\ c_2s_1 & c_1 & -s_1s_2 & 0 \\ s_2 & 0 & c_2 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_0^3 = \begin{bmatrix} c_1c_2c_3 - s_1s_3 & c_1s_2 & -c_3s_1 - c_1c_2s_3 & -l_2c_1s_2 \\ c_1s_3 + c_2c_3s_1 & s_1\,s_2 & c_1\,c_3 - c_2s_1s_3 & -l_2s_1s_2 \\ c_3s_2 & -c_2 & -s_2s_3 & l_1 + l_2c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_0^4 = \begin{bmatrix} c_1s_2s_4 - c_4(s_1s_3 - c_1c_2c_3) & -c_3s_1 - c_1c_2s_3 & -s_4(s_1s_3 - c_1c_2c_3) - c_1c_4s_2 & -l_2c_1s_2 \\ c_4(c_1s_3 + c_2c_3s_1) + s_1s_2s_4 & c_1c_3 - c_2s_1s_3 & s_4(c_1s_3 + c_2c_3s_1) - c_4s_1s_2 & -l_2s_1s_2 \\ c_3c_4s_2 - c_2s_4 & -s_2s_3 & c_2c_4 + c_3s_2s_4 & l_1 + l_2c_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_0^5 = \begin{bmatrix} \mu_{11} & \mu_{12} & \mu_{13} & \mu_{14} \\ \mu_{21} & \mu_{22} & \mu_{23} & \mu_{24} \\ \mu_{31} & \mu_{32} & \mu_{33} & \mu_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mu_{11} = -c_5(c_4(s_1s_3 - c_1c_2c_3) - c_1s_2s_4) - s_5(c_3s_1 + c_1c_2s_3)$$
$$\mu_{12} = -s_4(s_1s_3 - c_1c_2c_3) - c_1c_4s_2$$
$$\mu_{13} = c_5(c_3s_1 + c_1c_2s_3) - s_5(c_4(s_1s_3 - c_1c_2c_3) - c_1s_2s_4)$$
$$\mu_{14} = -l_3(s_4(s_1s_3 - c_1c_2c_3) + c_1c_4s_2) - l_2c_1s_2$$
$$\mu_{21} = c_5(c_4(c_1s_3 + c_2c_3s_1) + s_1s_2s_4) + s_5(c_1c_3 - c_2s_1s_3)$$
$$\mu_{22} = s_4(c_1s_3 + c_2c_3s_1) - c_4s_1s_2$$
$$\mu_{23} = s_5(c_4(c_1s_3 + c_2c_3s_1) + s_1s_2s_4) - c_5(c_1c_3 - c_2s_1s_3)$$
$$\mu_{24} = l_3(s_4(c_1s_3 + c_2c_3s_1) - c_4s_1s_2) - l_2s_1s_2$$
$$\mu_{31} = -c_5(c_2s_4 - c_3c_4s_2) - s_2s_3s_5$$
$$\mu_{32} = c_2c_4 + c_3s_2s_4$$
$$\mu_{33} = c_5s_2s_3 - s_5(c_2s_4 - c_3c_4s_2)$$
$$\mu_{34} = l_1 + l_3(c_2c_4 + c_3s_2 * s_4) + L_2c_2$$

$$T_0^6 = \begin{bmatrix} \mu_{11} & \mu_{12} & \mu_{13} & \mu_{14} \\ \mu_{21} & \mu_{22} & \mu_{23} & \mu_{24} \\ \mu_{31} & \mu_{32} & \mu_{33} & \mu_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mu_{11} = -s_6(s_4(s_1s_3 - c_1c_2c_3) + c_1c_4s_2) - c_6(c_5(c_4(s_1s_3 - c_1c_2c_3) - c_1s_2s_4) + s_5(c_3s_1 + c_1c_2s_3))$$
$$\mu_{12} = s_5(c_4(s_1s_3 - c_1c_2c_3) - c_1s_2s_4) - c_5(c_3s_1 + c_1c_2s_3)$$
$$\mu_{13} = s_6(c_5(c_4(s_1s_3 - c_1c_2c_3) - c_1s_2s_4) + s_5(c_3s_1 + c_1c_2s_3)) - c_6(s_4(s_1s_3 - c_1c_2c_3) + c_1c_4s_2)$$
$$\mu_{14} = -l_3(s_4(s_1s_3 - c_1c_2c_3) + c_1c_4s_2) - l_2c_1s_2]$$
$$\mu_{21} = s_6(s_4(c_1s_3 + c_2c_3s_1) - c_4s_1s_2) + c_6(c_5(c_4(c_1s_3 + c_2c_3s_1) + s_1s_2s_4) + s_5(c_1c_3 - c_2s_1s_3))$$
$$\mu_{22} = c_5(c_1c_3 - c_2s_1s_3) - s_5(c_4(c_1s_3 + c_2c_3s_1) + s_1s_2s_4)$$
$$\mu_{23} = c_6(s_4(c_1s_3 + c_2c_3s_1) - c_4s_1s_2) - s_6(c_5(c_4(c_1s_3 + c_2c_3s_1) + s_1s_2s_4) + s_5(c_1c_3 - c_2s_1s_3))$$
$$\mu_{24} = l_3(s_4(c_1s_3 + c_2c_3s_1) - c_4s_1s_2) - l_2s_1s_2$$
$$\mu_{31} = s_6(c_2c_4 + c_3s_2s_4) - c_6(c_5(c_2s_4 - c_3c_4s_2) + s_2s_3s_5)$$
$$\mu_{32} = s_5(c_2s_4 - c_3c_4s_2) - c_5s_2s_3$$
$$\mu_{33} = s_6(c_5(c_2s_4 - c_3c_4s_2) + s_2s_3s_5) + c_6(c_2c_4 + c_3s_2s_4)$$
$$\mu_{34} = l_1 + l_3(c_2c_4 + c_3s_2s_4) + l_2c_2$$

$$T_0^7 = \begin{bmatrix} \mu_{11} & \mu_{12} & \mu_{13} & \mu_{14} \\ \mu_{21} & \mu_{22} & \mu_{23} & \mu_{24} \\ \mu_{31} & \mu_{32} & \mu_{33} & \mu_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mu_{11} = s_7(s_5(c_4(s_1s_3 - c_1c_2c_3) - c_1s_2s_4) - c_5(c_3s_1 + c_1c_2s_3)) - c_7(s_6(s_4(s_1s_3 - c_1c_2c_3) + c_1c_4s_2) + c_6(c_5(c_4(s_1s_3 - c_1c_2c_3) - c_1s_2s_4) + s_5(c_3s_1 + c_1c_2s_3)))$$
$$\mu_{12} = s_6(c_5(c_4(s_1s_3 - c_1c_2c_3) - c_1s_2s_4) + s_5(c_3s_1 + c_1c_2s_3)) - c_6(s_4(s_1s_3 - c_1c_2c_3) + c_1c_4s_2)$$
$$\mu_{13} = -c_7(s_5(c_4(s_1s_3 - c_1c_2c_3) - c_1s_2s_4) - c_5(c_3s_1 + c_1c_2s_3)) - s_7(s_6(s_4(s_1s_3 - c_1c_2c_3) + c_1c_4s_2) + c_6(c_5(c_4(s_1s_3 - c_1c_2c_3) - c_1s_2s_4) + s_5(c_3s_1 + c_1c_2s_3)))$$
$$\mu_{14} = -l_4(c_6(s_4(s_1s_3 - c_1c_2c_3) + c_1c_4s_2) - s_6(c_5(c_4(s_1s_3 - c_1c_2c_3) - c_1s_2s_4) + s_5(c_3s_1 + c_1c_2s_3))) - l_3(s_4(s_1s_3 - c_1c_2c_3) + c_1c_4s_2) - l_2c_1s_2]$$
$$\mu_{21} = c_7(s_6(s_4(c_1s_3 + c_2c_3s_1) - c_4s_1s_2) + c_6(c_5(c_4(c_1s_3 + c_2c_3s_1) + s_1s_2s_4) + s_5(c_1c_3 - c_2s_1s_3))) - s_7(s_5(c_4(c_1s_3 + c_2c_3s_1) + s_1s_2s_4) - c_5(c_1c_3 - c_2s_1s_3))$$
$$\mu_{22} = c_6(s_4(c_1s_3 + c_2c_3s_1) - c_4s_1s_2) - s_6(c_5(c_4(c_1s_3 + c_2c_3s_1) + s_1s_2s_4) + s_5(c_1c_3 - c_2s_1s_3))$$
$$\mu_{23} = c_7(s_5(c_4(c_1s_3 + c_2c_3s_1) + s_1s_2s_4) - c_5(c_1c_3 - c_2s_1s_3)) + s_7(s_6(s_4(c_1s_3 + c_2c_3s_1) - c_4s_1s_2) + c_6(c_5(c_4(c_1s_3 + c_2c_3s_1) + s_1s_2s_4) + s_5(c_1c_3 - c_2s_1s_3)))$$
$$\mu_{24} = l_4(c_6(s_4(c_1s_3 + c_2c_3s_1) - c_4s_1s_2) - s_6(c_5(c_4(c_1s_3 + c_2c_3s_1) + s_1s_2s_4) + s_5(c_1c_3 - c_2s_1s_3))) + l_3(s_4(c_1s_3 + c_2c_3s_1) - c_4s_1s_2) - l_2s_1s_2]$$
$$\mu_{31} = s_7(s_5(c_2s_4 - c_3c_4s_2) - c_5s_2s_3) - c_7(c_6(c_5(c_2s_4 - c_3c_4s_2) + s_2s_3s_5) - s_6(c_2c_4 + c_3s_2s_4))$$
$$\mu_{32} = s_6(c_5(c_2s_4 - c_3c_4s_2) + s_2s_3s_5) + c_6(c_2c_4 + c_3s_2s_4)$$
$$\mu_{33} = -c_7(s_5(c_2s_4 - c_3c_4s_2) - c_5s_2s_3) - s_7(c_6(c_5(c_2s_4 - c_3c_4s_2) + s_2s_3s_5) - s_6(c_2c_4 + c_3s_2s_4))$$
$$\mu_{34} = l_1 + l_3(c_2c_4 + c_3s_2s_4) + l_4(s_6(c_5(c_2s_4 - c_3c_4s_2) + s_2s_3s_5) + c_6(c_2c_4 + c_3s_2s_4)) + l_2c_2$$

Desired Trajectory

Selecting a desired trajectory is a tricky task. The desired trajectory should be in the workspace of the robot and also should have no singularity. The first desired trajectory was selected to be a circle of radius r and center(x_c, y_c, z_c). This desired circle is in a fixed $x = x_c$ plane. In this project, the following values were selected for the desired trajectory:

$$r = 2$$

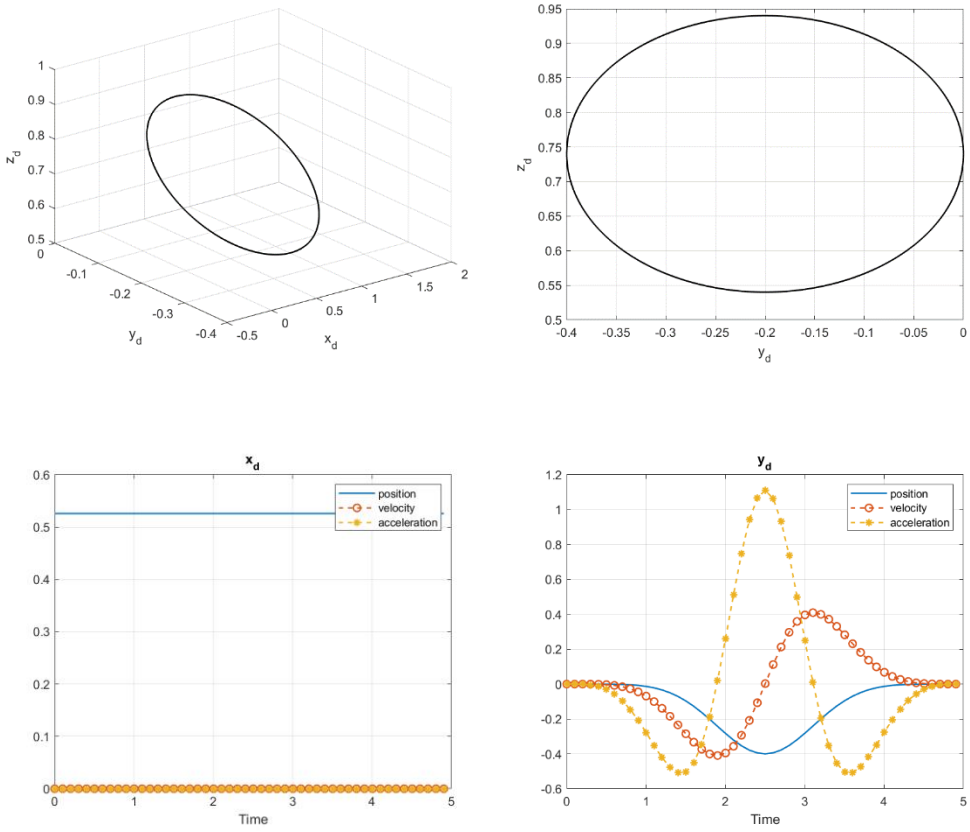
$$x_c = L_3 + L_4$$

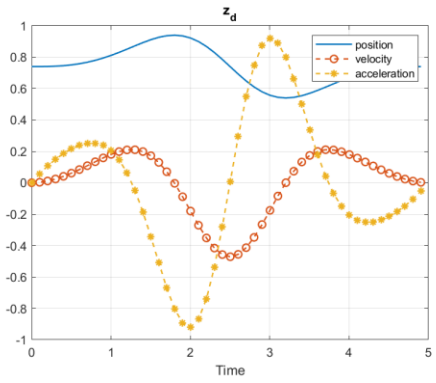
$$yc = -r$$

$$z_c = L_1 + L_2$$

$$R_{E(desired)} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix}$$

Figures below show the desired trajectory:





Inverse Kinematic

The inverse kinematics problem for a robot manipulator is a method of obtaining the required manipulator joint variable values for any desired endpoint position and orientation. One of the requirement to solve the inverse problem is the existence of the desired solution.

Inverse kinematics problems are tricky and indirect to solve. Redundant manipulators have an infinite number of configurations that lead to the same end-effector desired position and orientation. Some researchers solved the redundant manipulators problem by using iterative (numerical) methods. The following equation has to be solved to achieve the inverse solution.

$$F = \begin{bmatrix} \textbf{Actual position} - \textbf{Desired postion} \\ \textbf{Actual orientation} - \textbf{Desired oreintation} \end{bmatrix}$$

Where, F consists of 12 equations; 3 for position and 9 for orientation.

Assume the desired end effector pose at time i has the following formula:

$$T_{Ed}(t_i) = \begin{bmatrix} n_{xd} & o_{xd} & a_{xd} & x_d \\ n_{yd} & o_{yd} & a_{yd} & y_d \\ n_{zd} & o_{zd} & a_{zd} & z_d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

And the actual end effector pose at the same time has the following formula:

$$T_{Ea}(t_i) = \begin{bmatrix} n_{xa} & o_{xa} & a_{xa} & x_a \\ n_{ya} & o_{ya} & a_{ya} & y_a \\ n_{za} & o_{za} & a_{za} & z_a \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then,

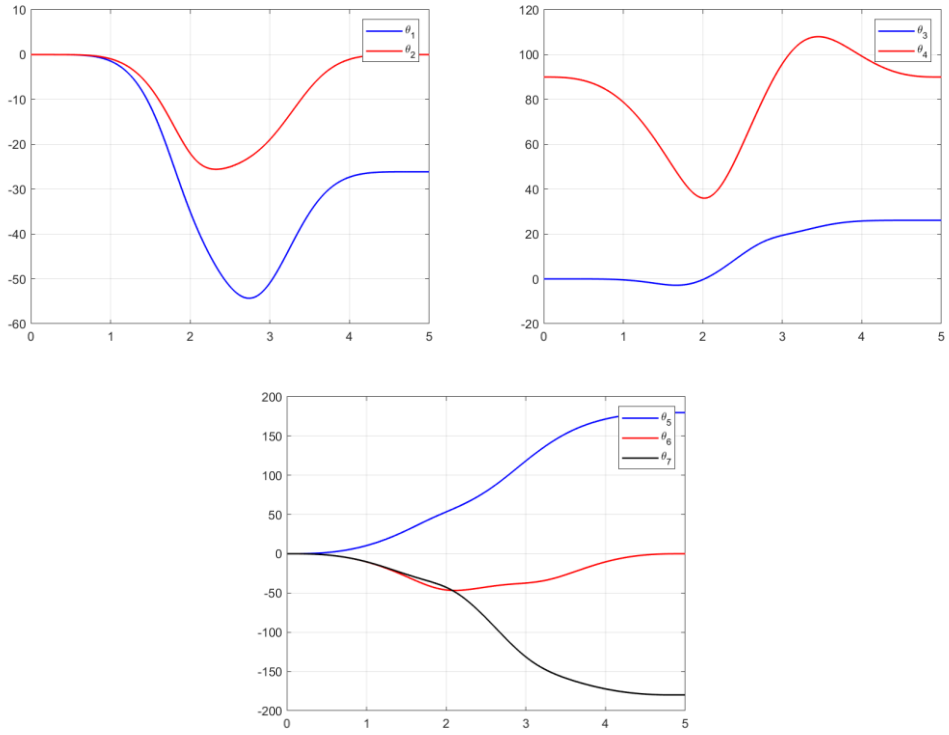
$$F = \begin{bmatrix} x_d - x_a \\ y_d - y_a \\ z_d - z_a \\ n_{xd} - n_{xa} \\ n_{yd} - n_{ya} \\ n_{zd} - n_{za} \\ o_{xd} - o_{xa} \\ o_{yd} - o_{ya} \\ o_{zd} - o_{za} \\ a_{xd} - a_{xa} \\ a_{yd} - a_{ya} \\ a_{zd} - a_{za} \end{bmatrix}$$

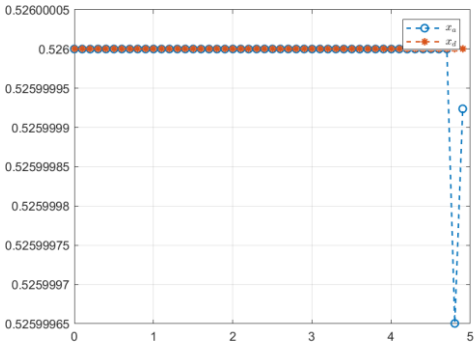
To solve the inverse problem the following iterative formula was used

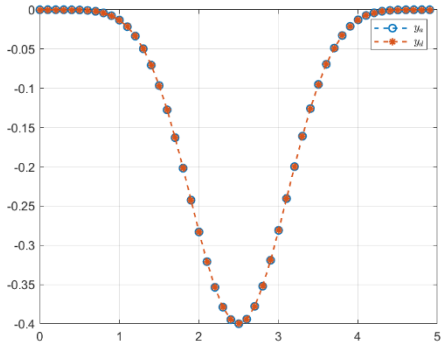
$$\theta_{i+1} = Fsolve\ (F, \theta_i)$$

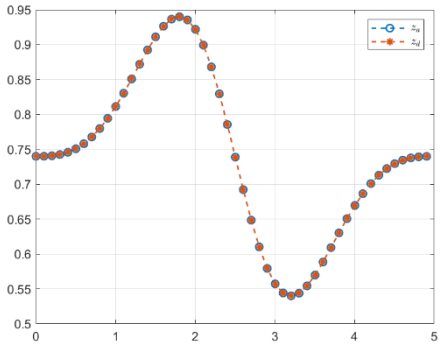
Where $Fsolve$ is a MATLAB function solve for $F = 0$ given θ_i as initial point.

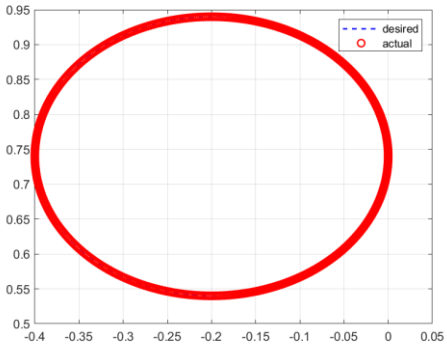
The following results were found using the above formula:

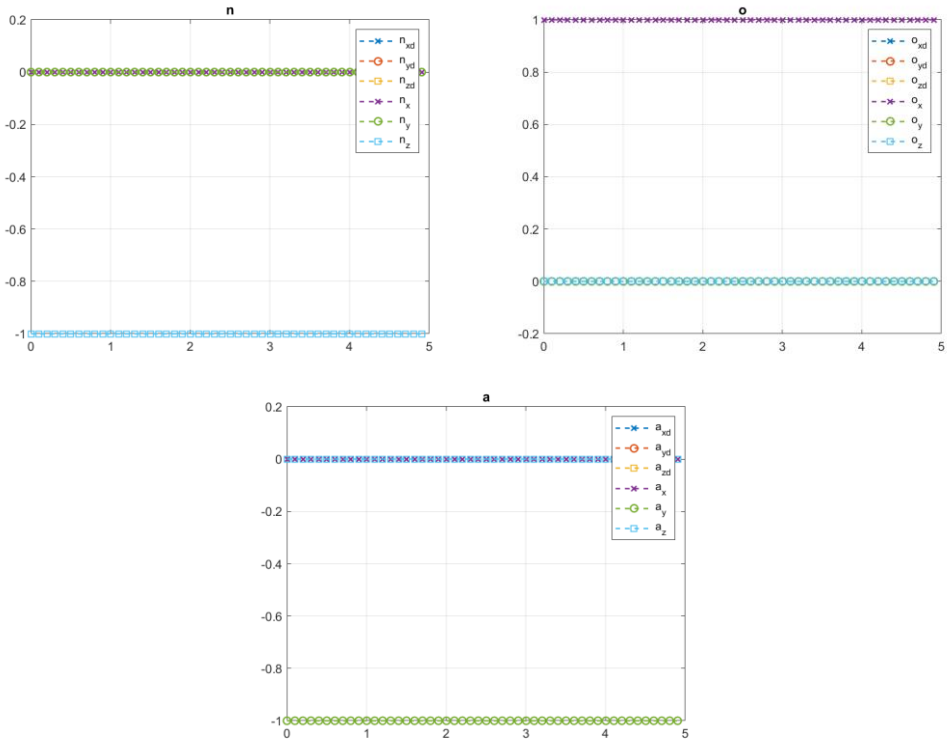



x_a
x_d
0.52600005
0.526
0.52599995
0.5259999
0.52599985
0.5259998
0.52599975
0.5259997
0.52599965
0
1
2
3
4
5

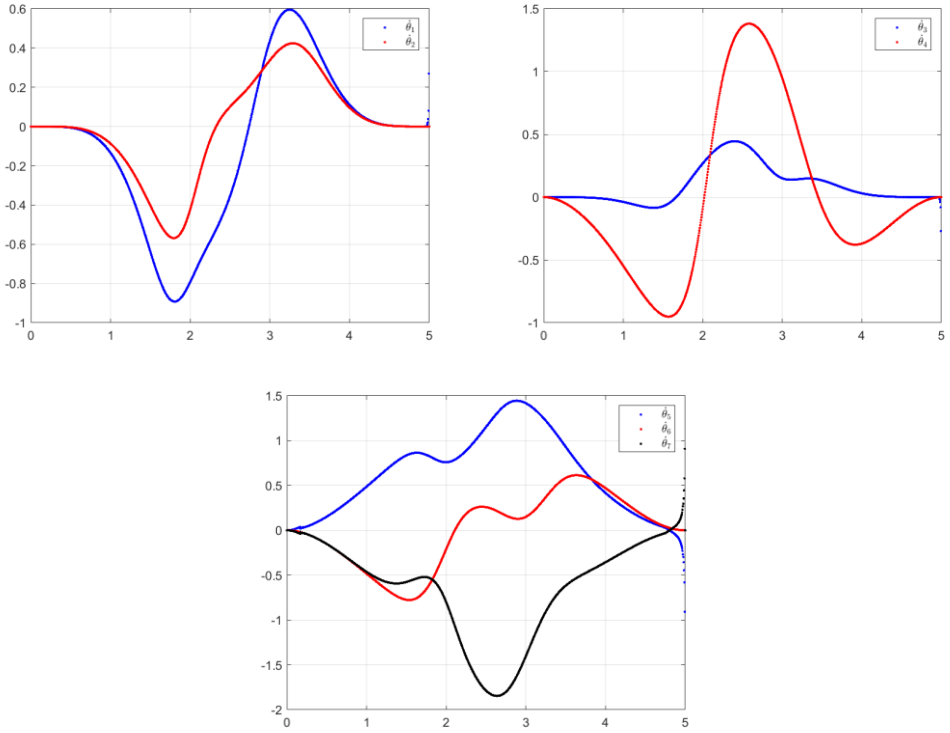

y_a
y_d
0
-0.05
-0.1
-0.15
-0.2
-0.25
-0.3
-0.35
-0.4
0
1
2
3
4
5

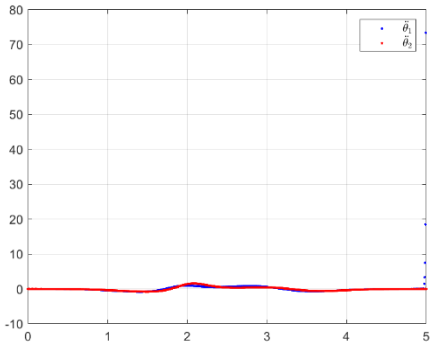

z_a
z_d
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0
1
2
3
4
5

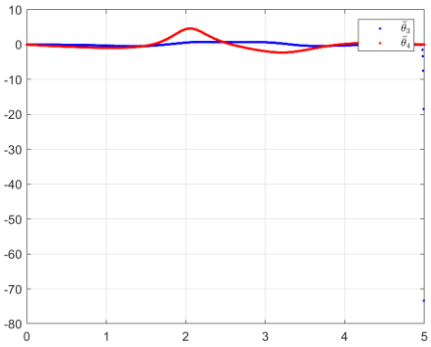

desired
actual
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
-0.4
-0.35
-0.3
-0.25
-0.2
-0.15
-0.1
-0.05
0
0.05

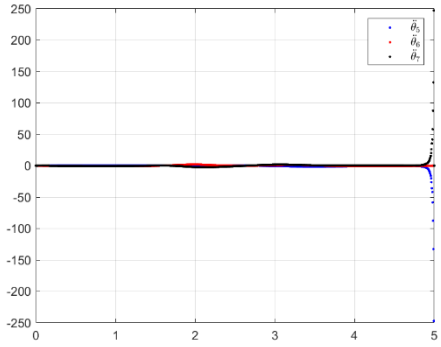


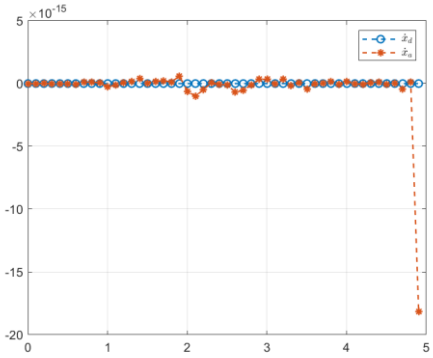
As it can be seen the trajectory is identical to the desired path. Next step is solving for the inverse velocity and acceleration. Results are shown in the following figures:

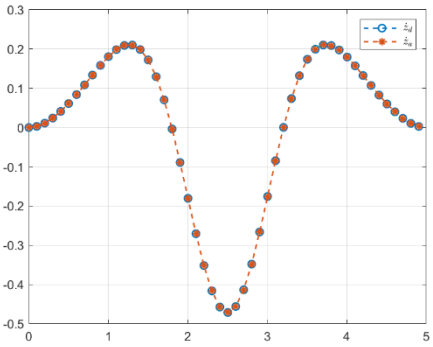


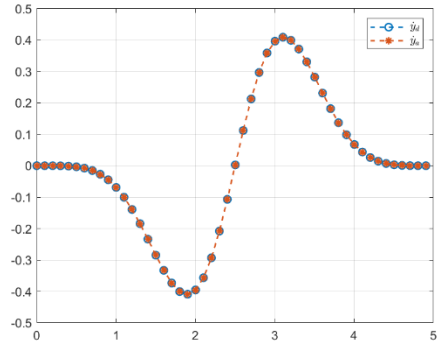

$\tilde{\theta}_1$
$\tilde{\theta}_2$

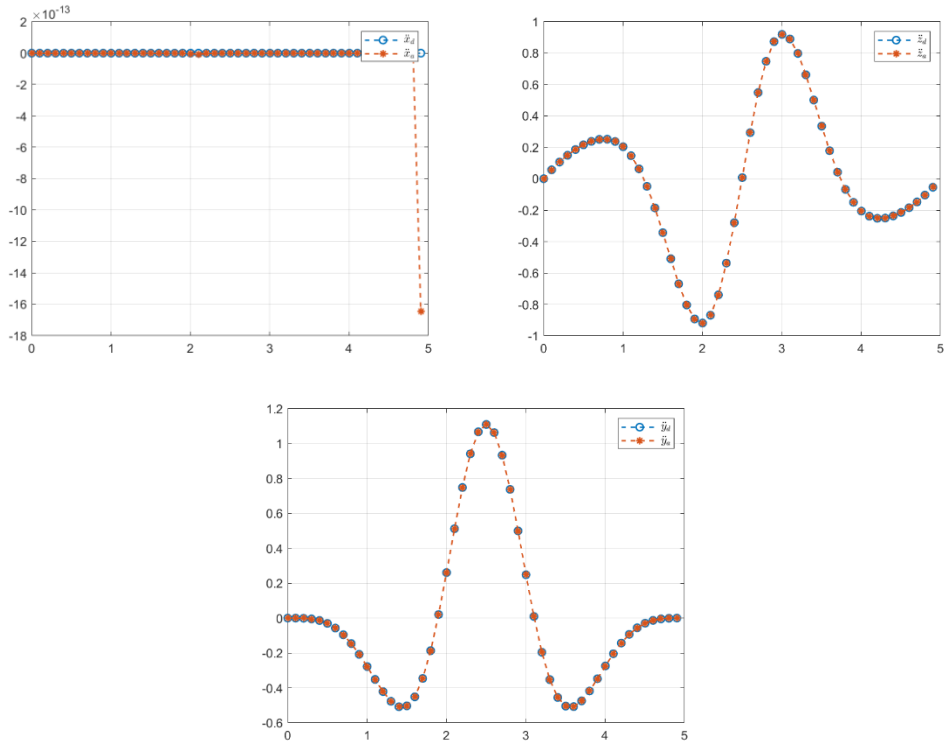

$\tilde{\theta}_3$
$\tilde{\theta}_4$


$\tilde{\theta}_5$
$\tilde{\theta}_6$
$\tilde{\theta}_7$


$\times 10^{-15}$
$\tilde{x}_d$
$\tilde{x}_a$


$\tilde{z}_d$
$\tilde{z}_a$


$\tilde{y}_d$
$\tilde{y}_a$



It can be seen that the solution has singularities, this can be checked by using $rank$ or $cond$ MATLAB functions. To avoid singularities another trajectory is proposed using the following formulas:

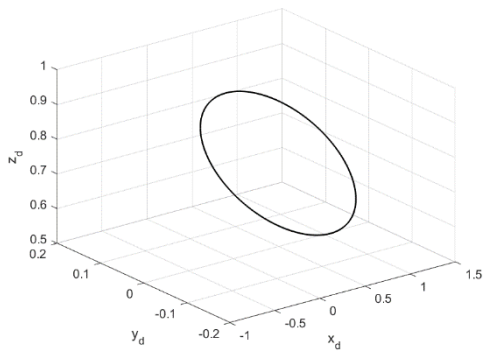
$$r = (L_3 + L_4)\sin(\beta)$$

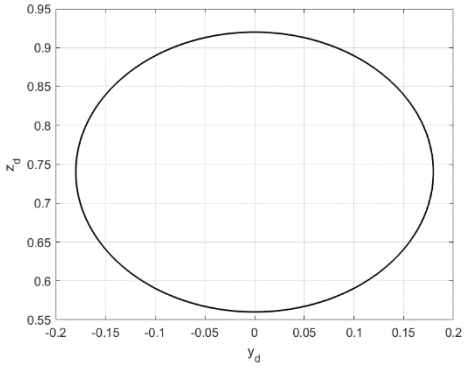
$$x_c = (L_3 + L_4)\cos(\beta)$$

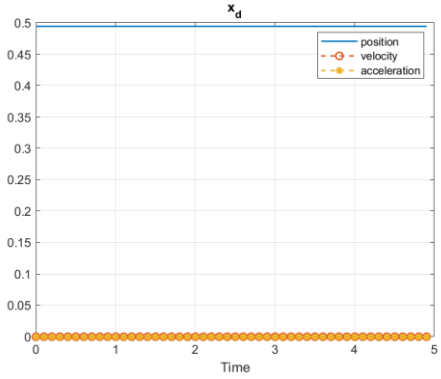
$$yc = 0$$

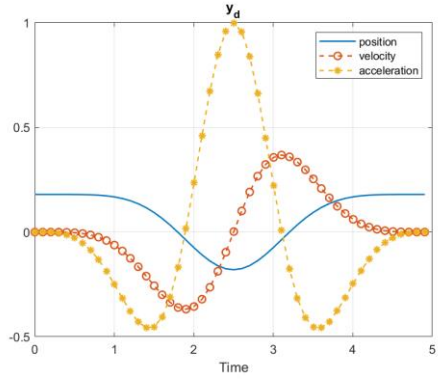
$$z_c = L_1 + L_2$$

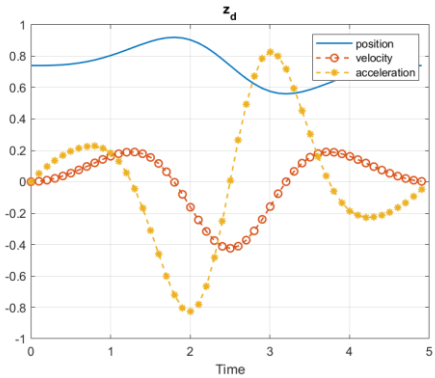
$$R_{E(desired)} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix}$$



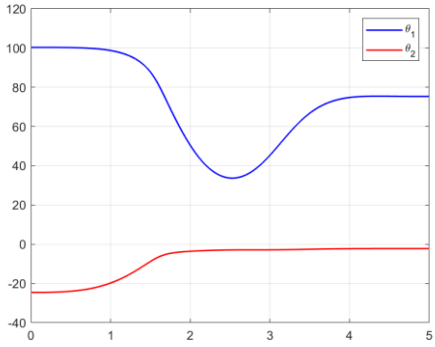


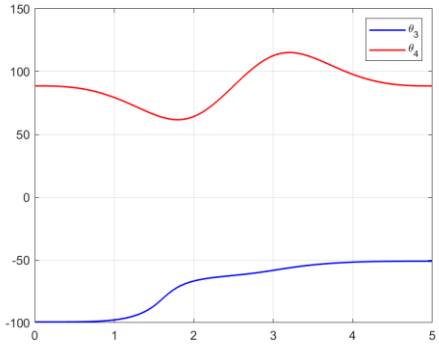


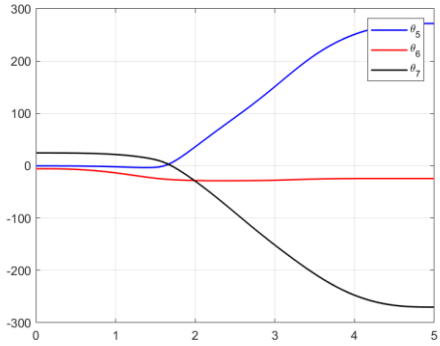


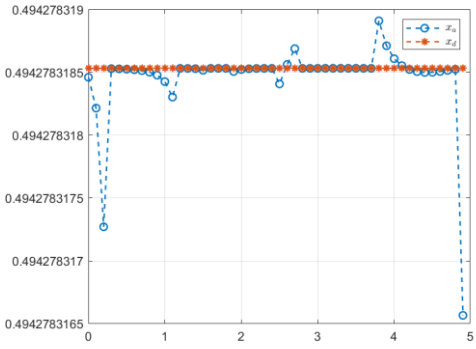


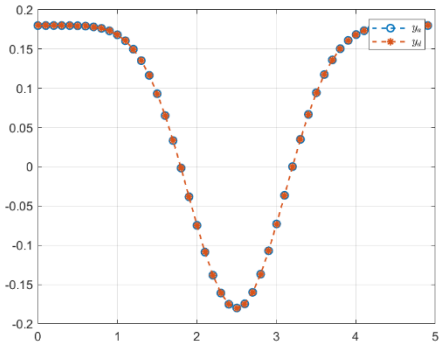
The inverse solution look like:

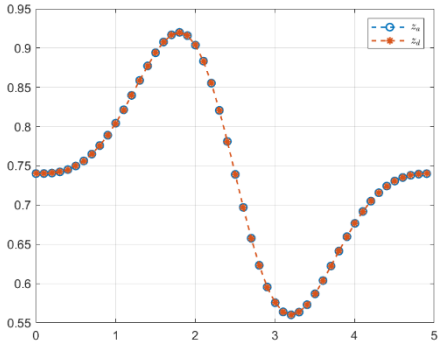

θ_1
θ_2

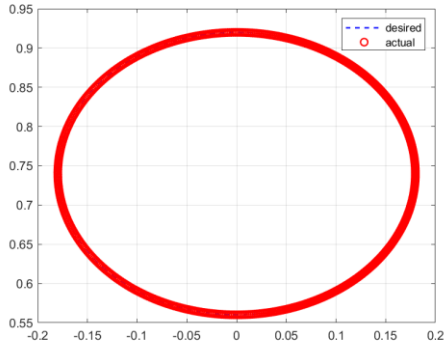

θ_3
θ_4

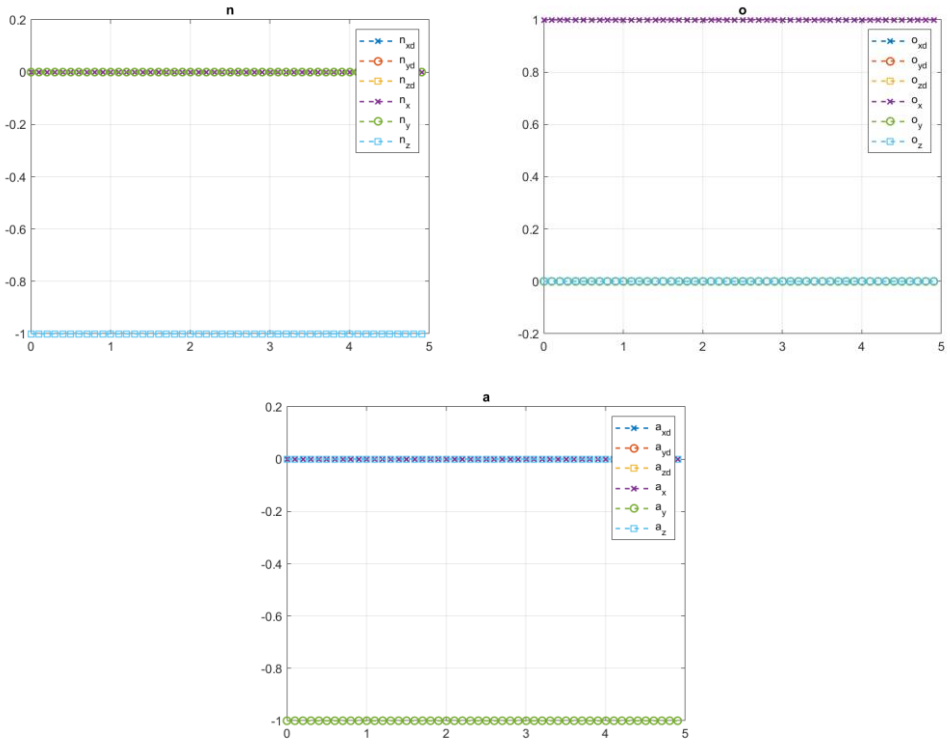

θ_5
θ_6
θ_7


x_a
x_d

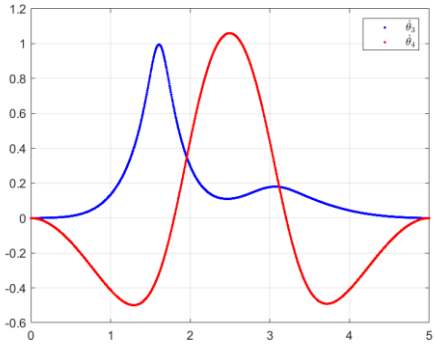

y_a
y_d

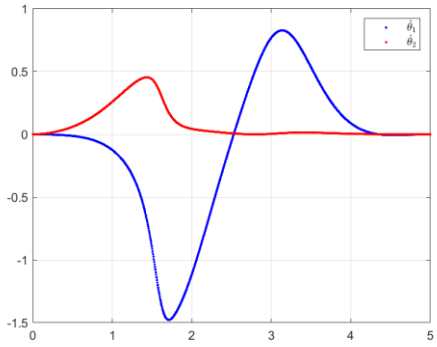

z_a
z_d

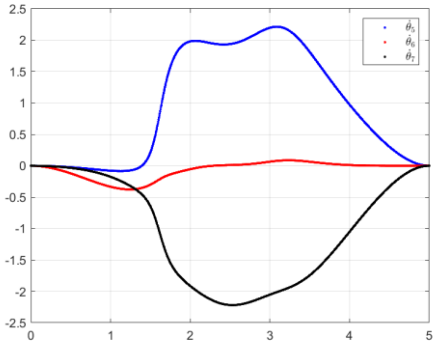

desired
actual

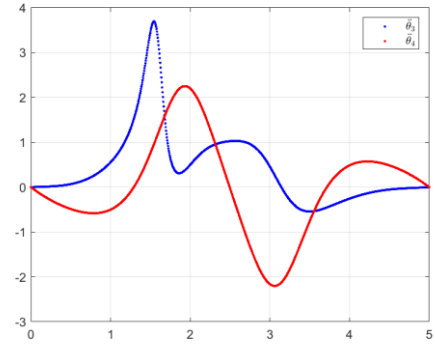


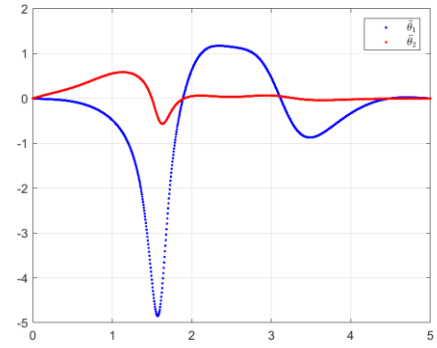
The inverse velocity and acceleration are shown in the following figures:

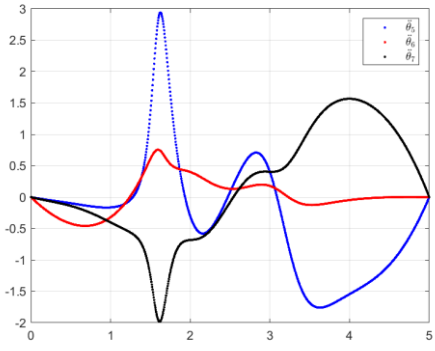

$\bar{\theta}_3$
$\bar{\theta}_4$

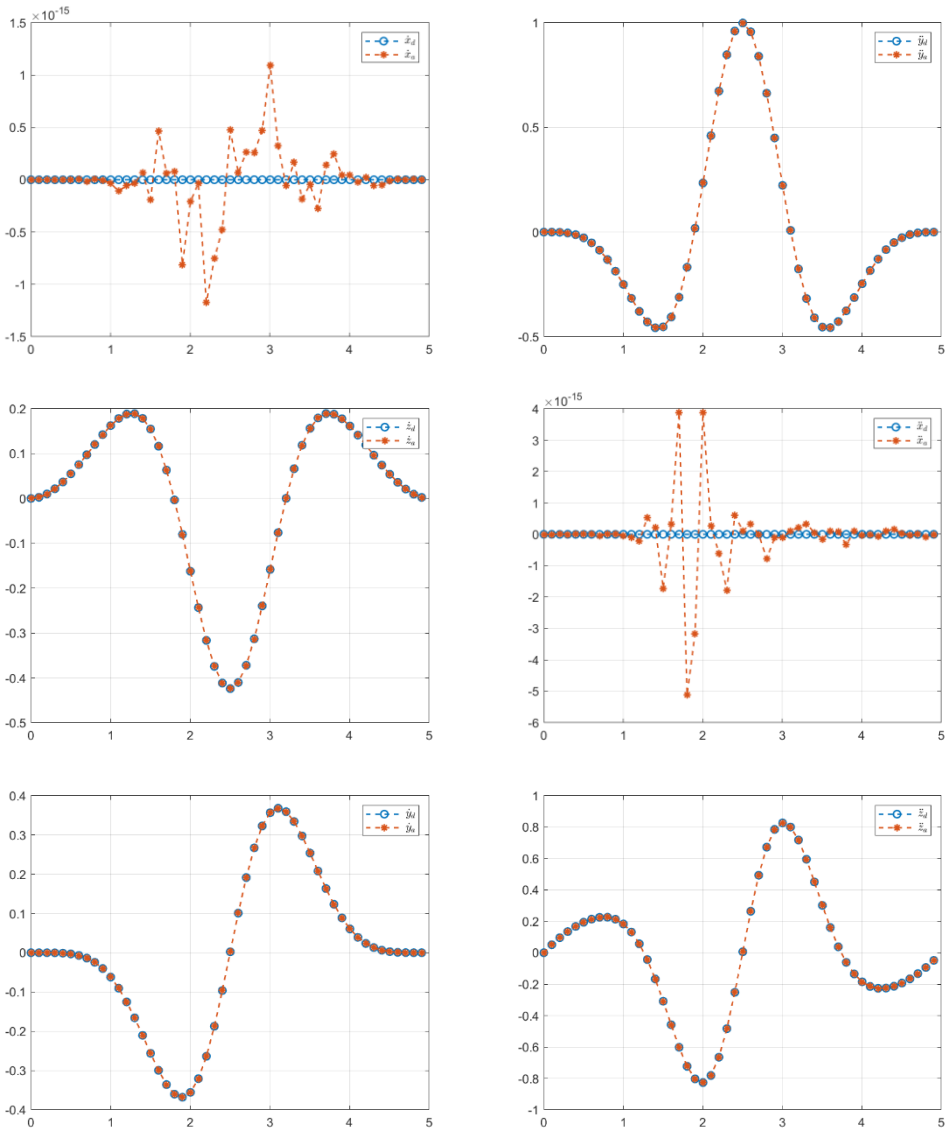

$\bar{\theta}_1$
$\bar{\theta}_2$


$\bar{\theta}_5$
$\bar{\theta}_6$
$\bar{\theta}_7$

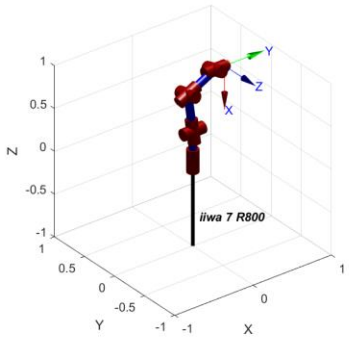

$\bar{\theta}_3$
$\bar{\theta}_4$

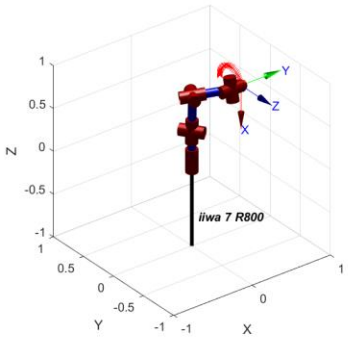

$\bar{\theta}_1$
$\bar{\theta}_2$

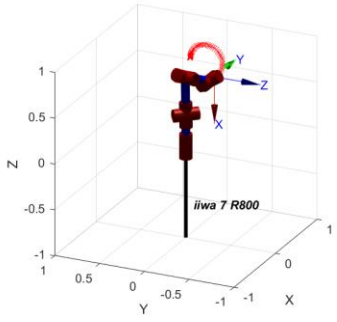

$\bar{\theta}_5$
$\bar{\theta}_6$
$\bar{\theta}_7$

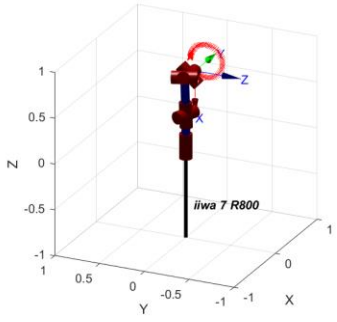


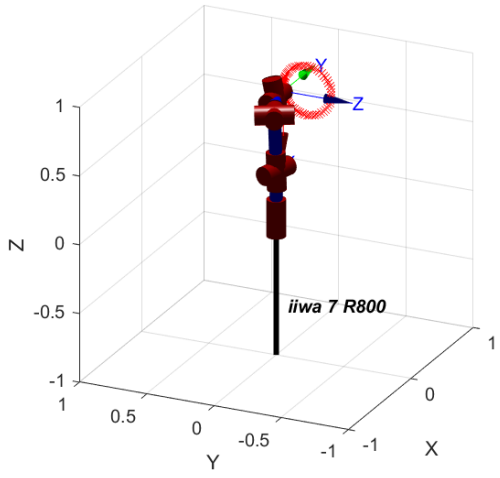
It can be seen that there are no singularities in this particular path. The study will use this path in the following sections. Figures below show a simulation of the robot using the desired trajectory.


iiwa 7 R800
X
Y
Z


iiwa 7 R800
X
Y
Z


iiwa 7 R800
X
Y
Z


iiwa 7 R800
X
Y
Z


iiwa 7 R800
X
Y
Z

Dynamical modeling

The equation of motion (EOM) can be derived as:

$$M\ddot{q} + c + \phi + B\dot{q} = \tau$$

- The inertial parameter of the links for the iiwa 7 model obtained from KUKA website.
- Off diagonal elements in the mass moment of inertia matrix are ignored (rigid body symmetric assumption)
- Center of masses were assumed to be in the center of each link
- No friction

Because of the long expressions the EOM are not shown here.

	Link						
Parameter	1	2	3	4	5	6	7
m	3.4525	3.4821	4.0562	3.4822	2.1633	2.3466	3.1290
r_x	0.0	0.0	0.0	0.0	0.0	0.0	0.0
r_y	0.0	0.0	0.0	0.0	0.0	0.0	0.0
r_z	$L\_1/2$	0.0	$L\_2/2$	0.0	$L\_3/2$	0.0	$L\_4/2$
I_{xx}	0.02180	0.02076	0.03200	0.02170	0.01280	0.00650	0.01460
I_{yy}	0.00770	0.02179	0.00970	0.02070	0.00580	0.00620	0.01470
I_{zz}	0.02080	0.00779	0.03040	0.00770	0.01110	0.00450	0.00290

Sliding Control

$$\tau = c + \phi + B\dot{q} + M(\ddot{q}_d - \lambda(\dot{q} - \dot{q}_d) - k\, sat(s/\epsilon))$$

Where, $k = F + \eta$

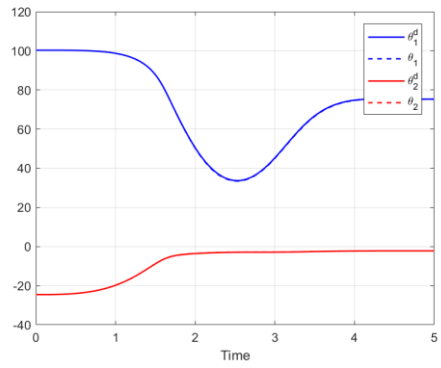
Three parameter were changed to achieve the desired trajectory

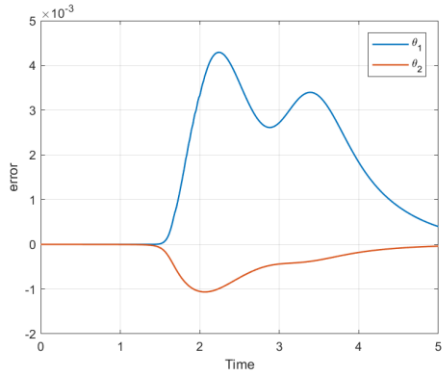
- ϵ: boundary layer thickness
- λ: speed of convergence
- η: control effort

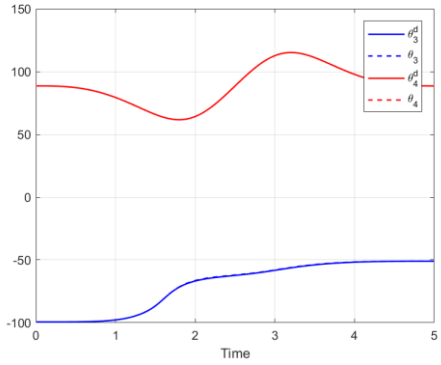
Case 1:

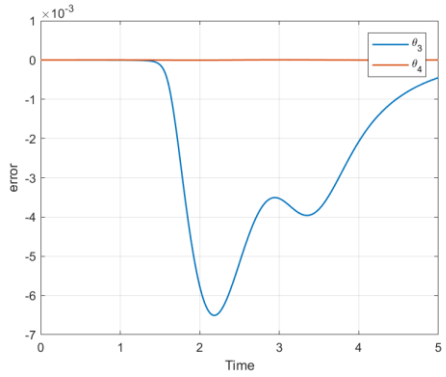
$$\epsilon = 0.01 \;\; \lambda = 1.5 \;\; \eta = 4$$

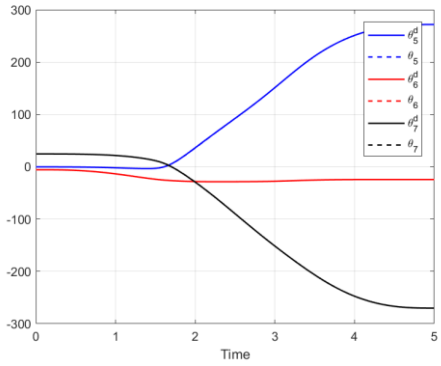
The following figures show the actual response compared with the desired response. Also, the error for each response is presented.

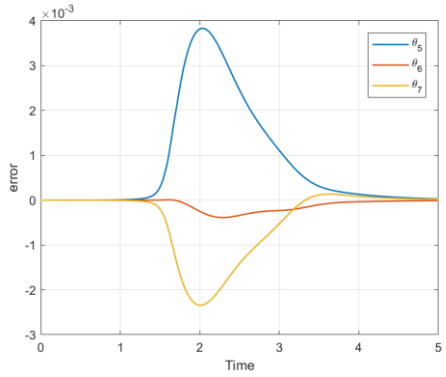


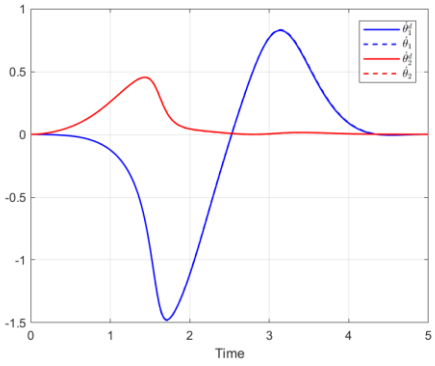


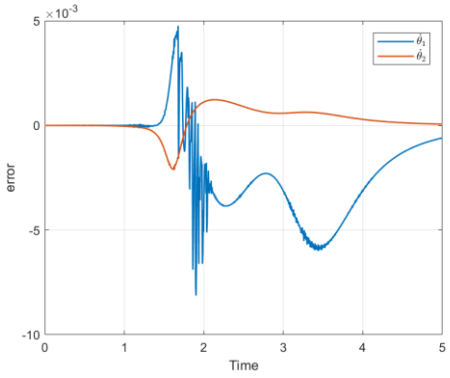


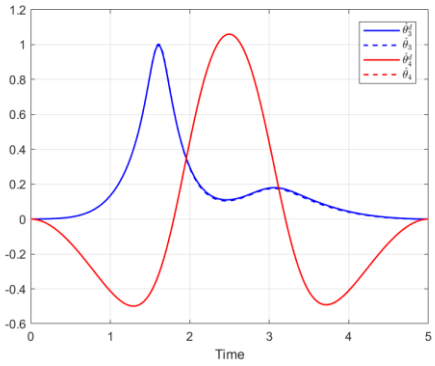


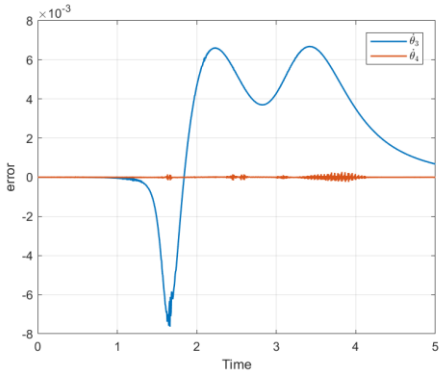


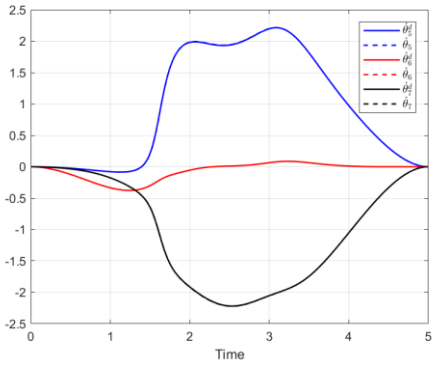


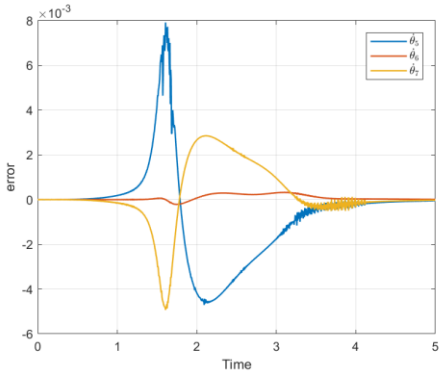

Time


error
Time

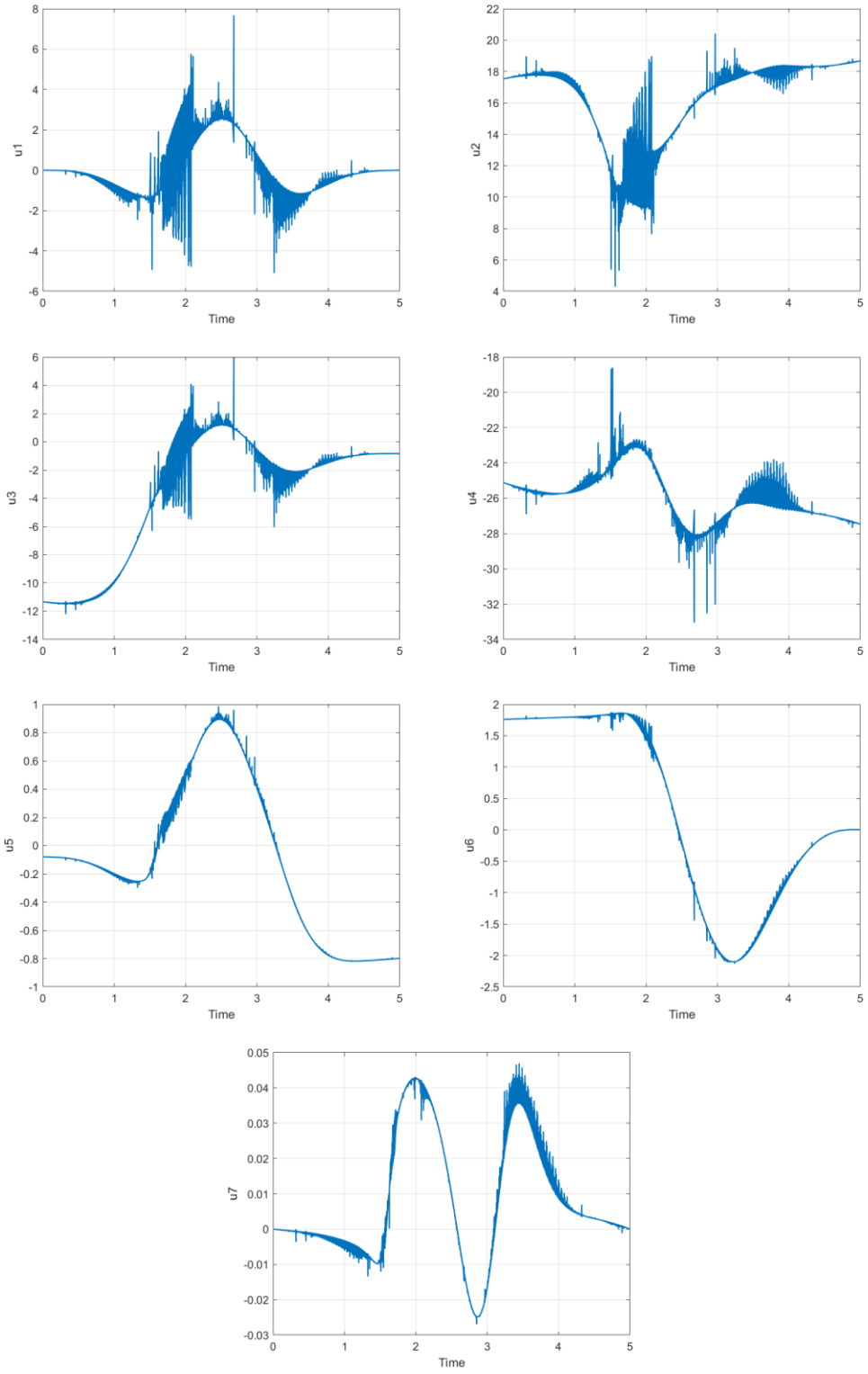

Time


error
Time

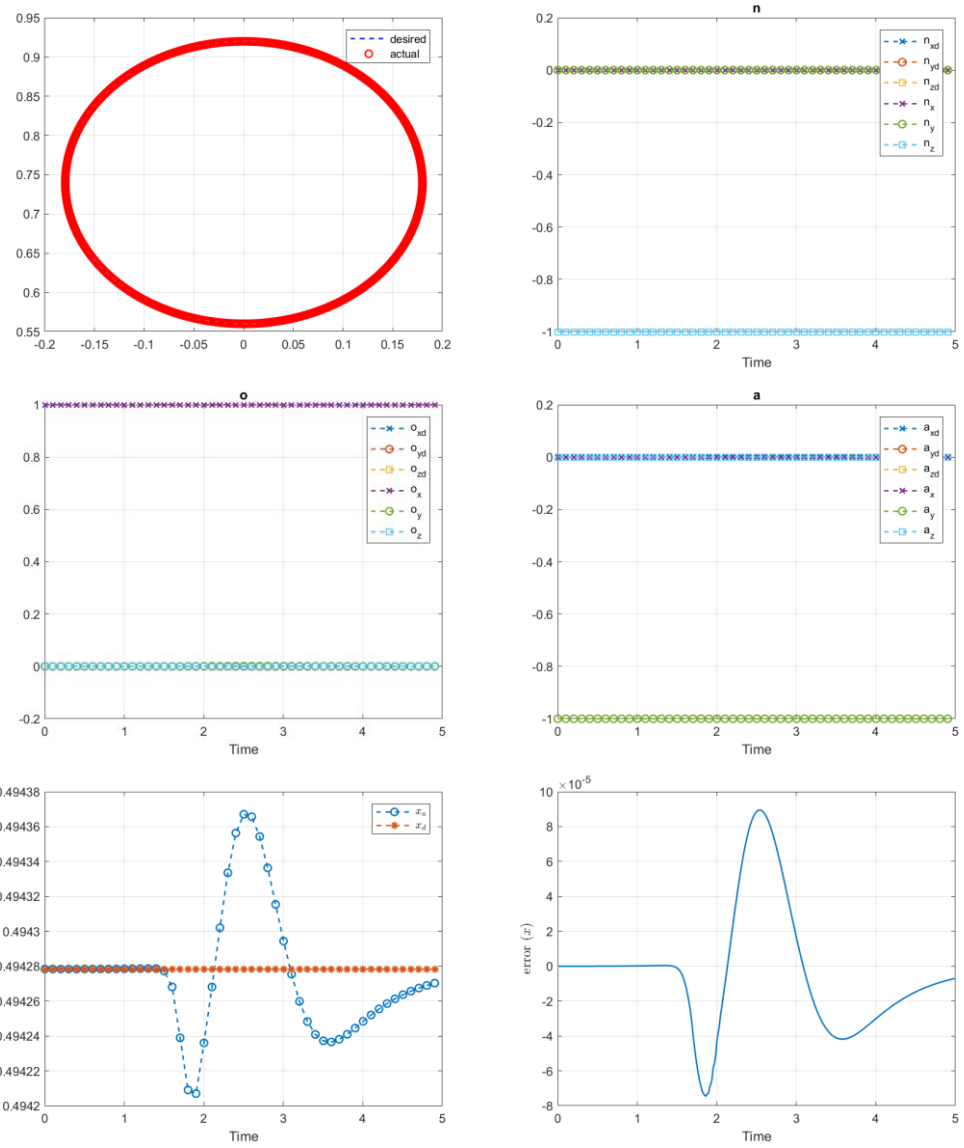

Time

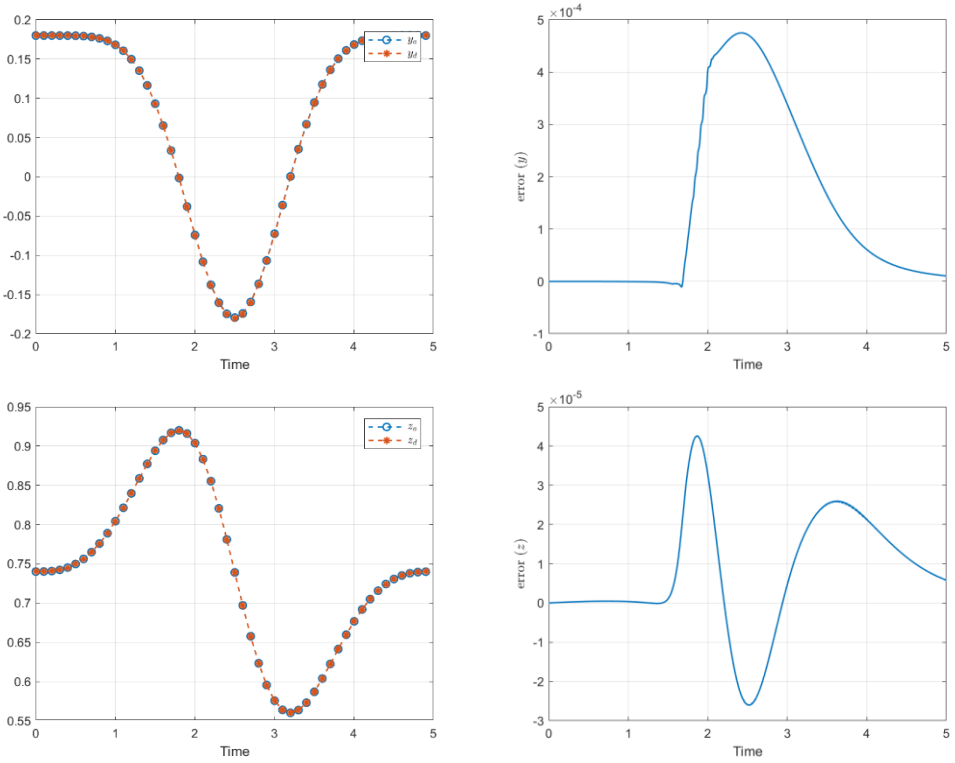

error
Time

The input of the sliding controller is presented the figures below. The chattering problem is obvious, this can be solved by increasing ϵ which is presented in the following cases.



Finally, a comparison between the actual and the desired position and orientation of the end effector are shown below:

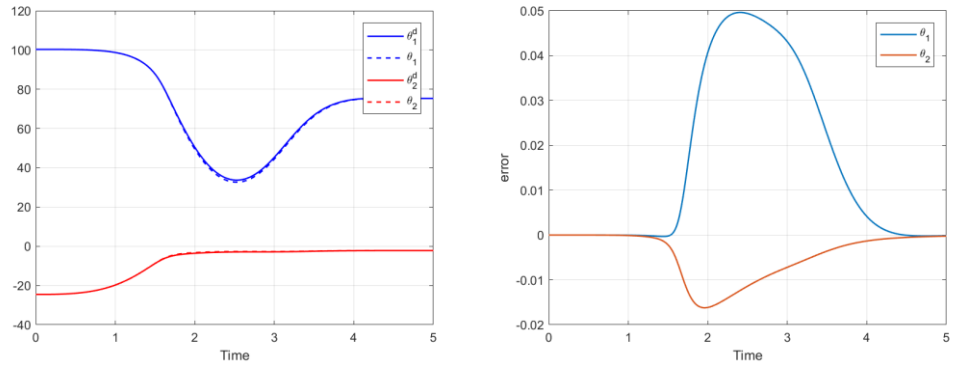


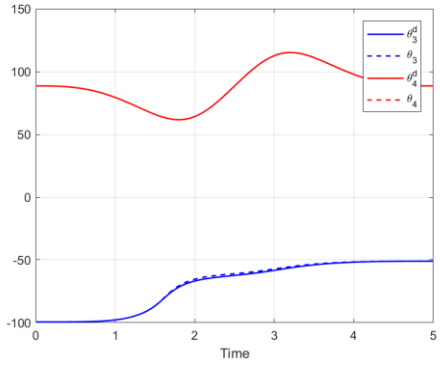


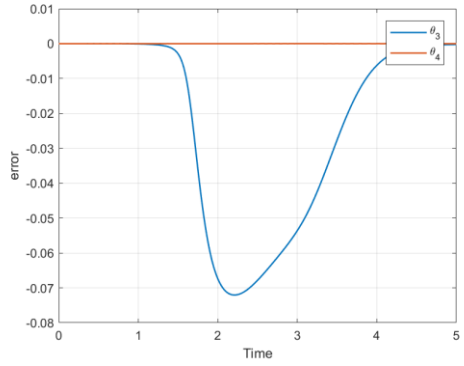
Case 2:

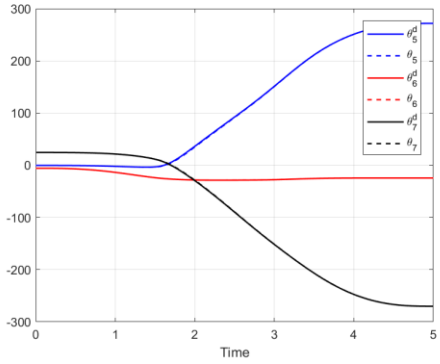
$$\epsilon = 0.3 \;\; \lambda = 1.5 \;\; \eta = 4$$

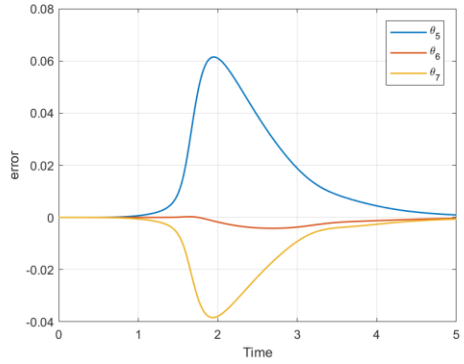
The following figures show the actual response compared with the desired response. Also, the error for each response is presented.

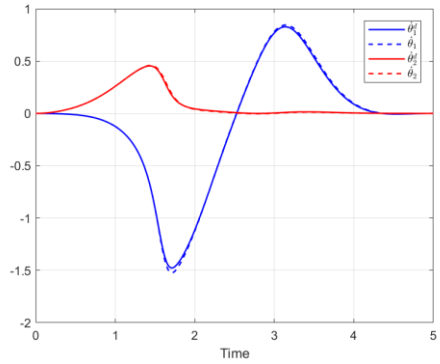


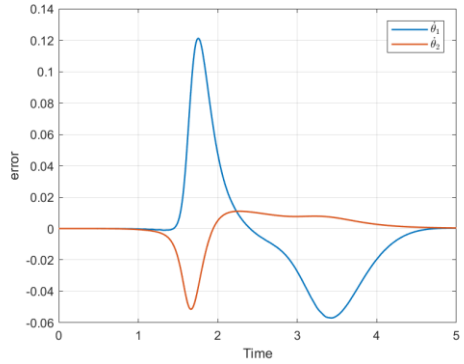

Time

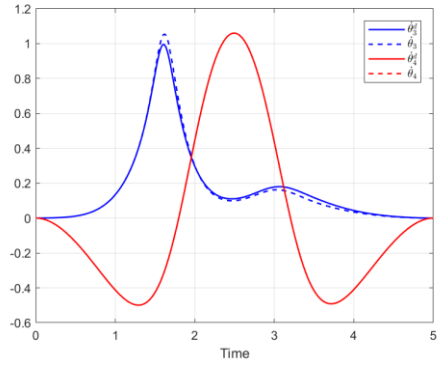

error
Time

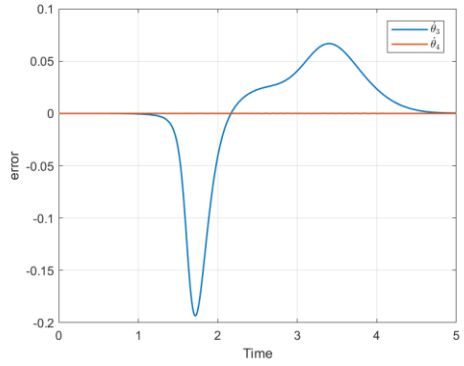

Time

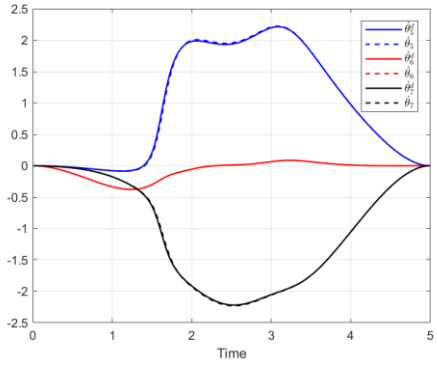

error
Time

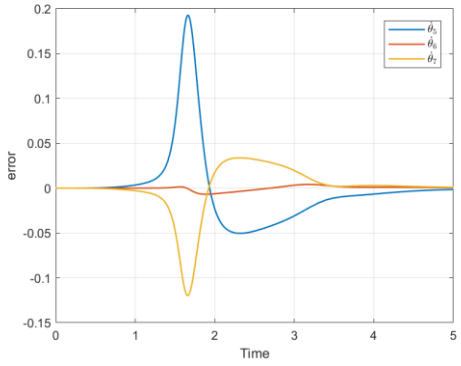

Time


error
Time

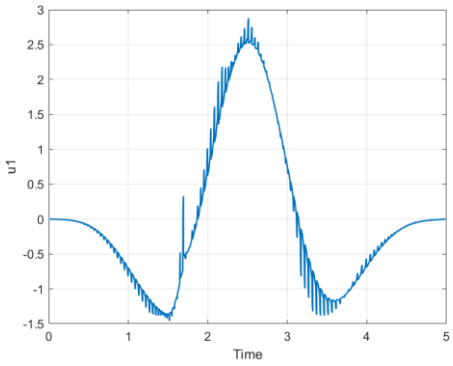

Time

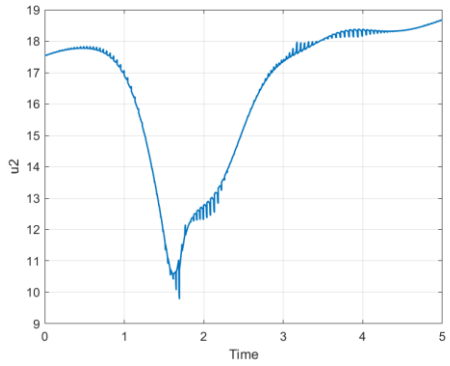

error
Time

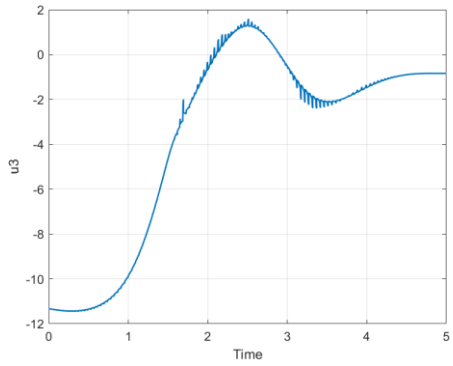


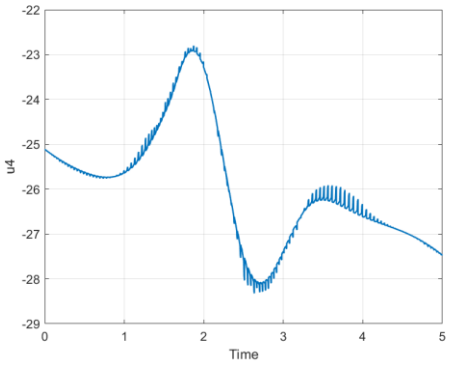


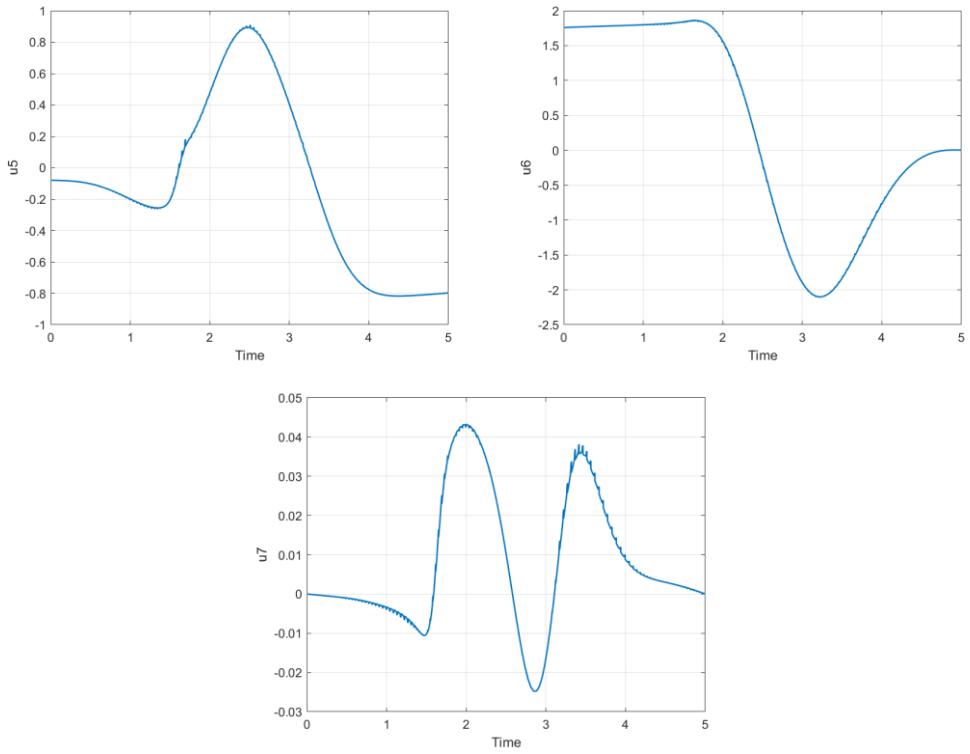
The input of the sliding controller is presented the figures below. The chattering still exist but it is much lower compared to case 1.



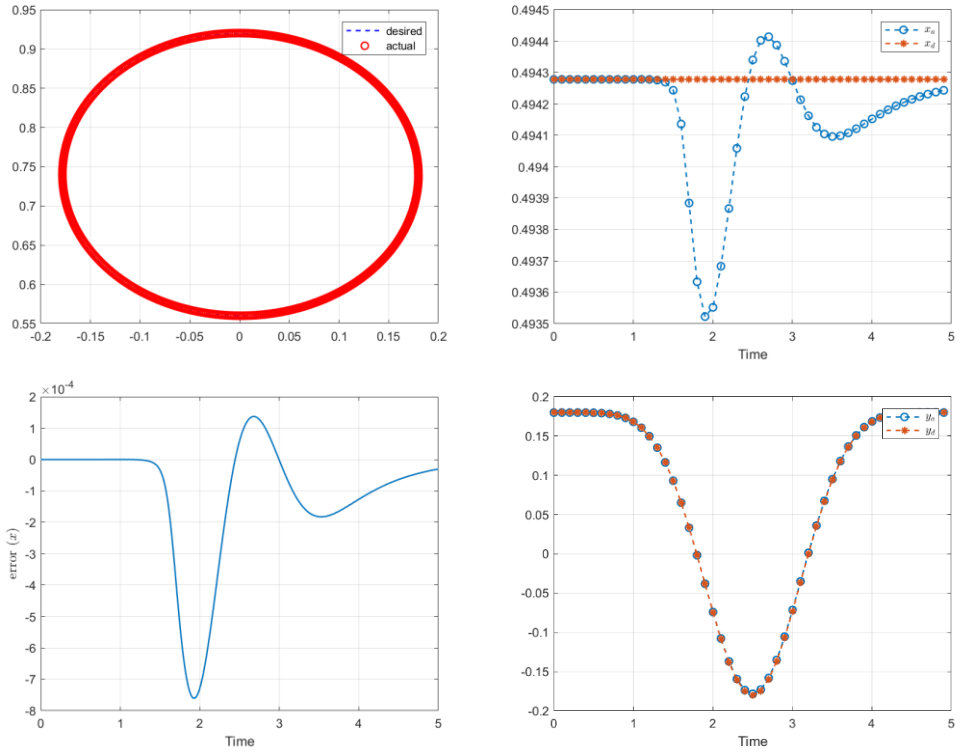


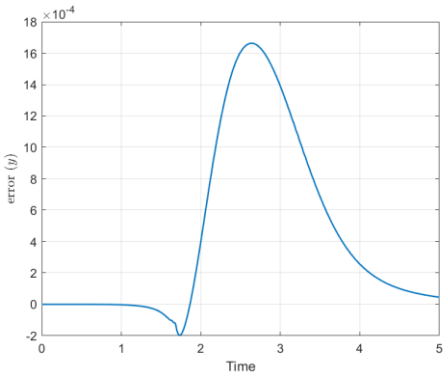


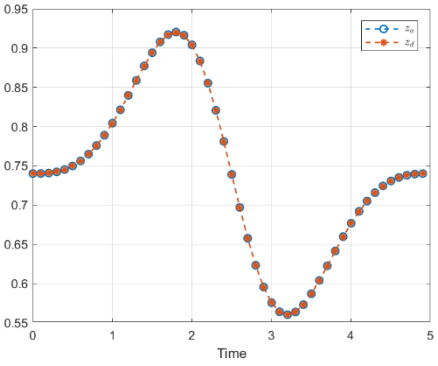


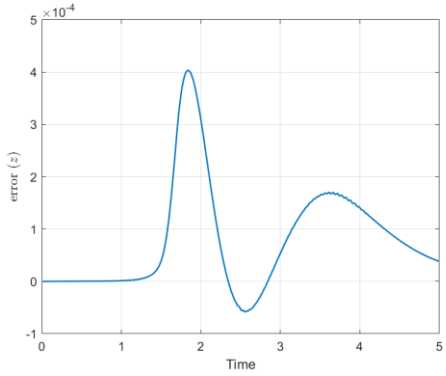


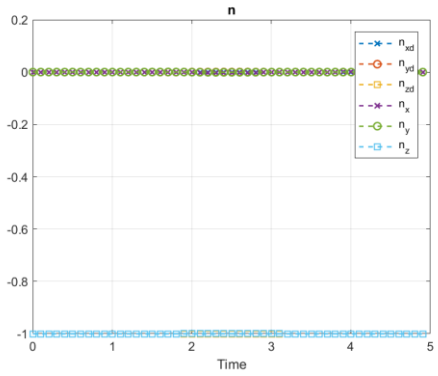
Finally, a comparison between the actual and the desired position and orientation of the end effector are shown below:

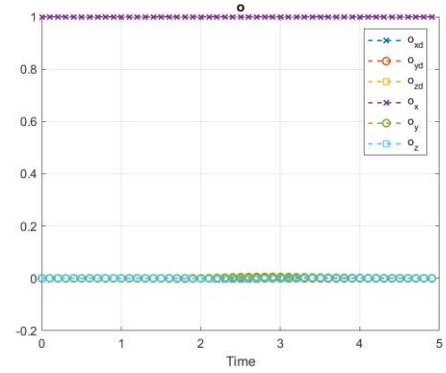


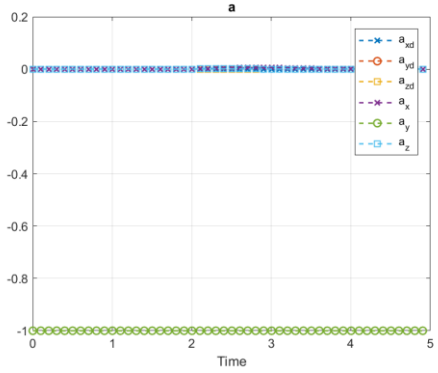






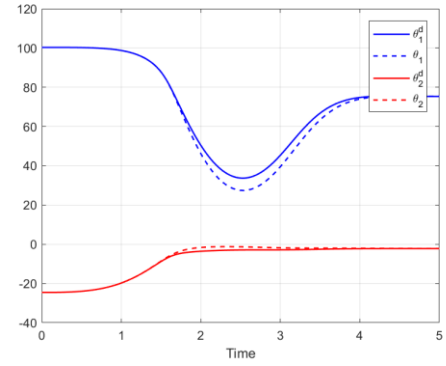


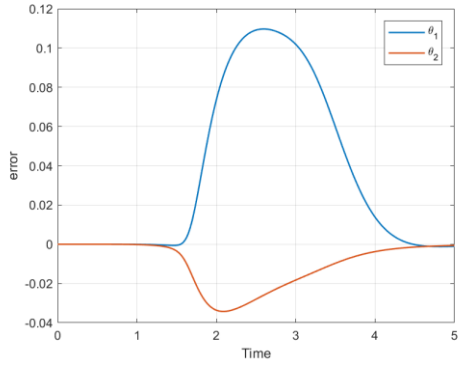


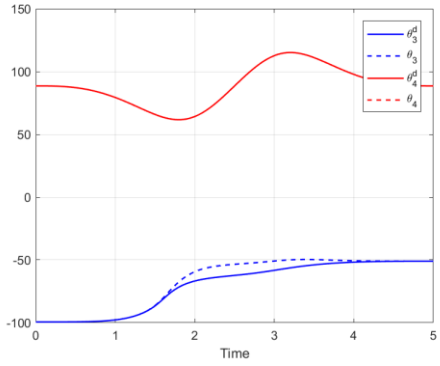


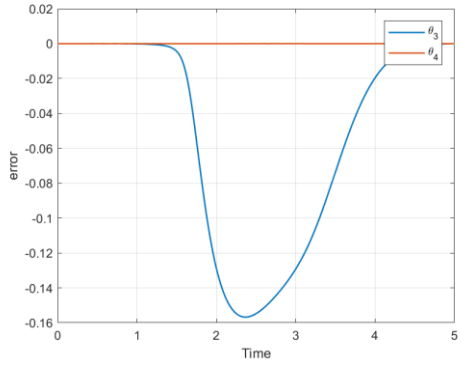
Case 3:

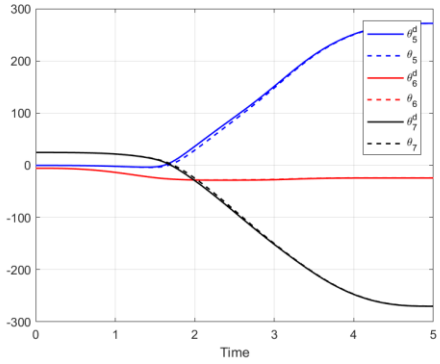
$$\epsilon = 0.7 \;\; \lambda = 1.5 \;\; \eta = 4$$

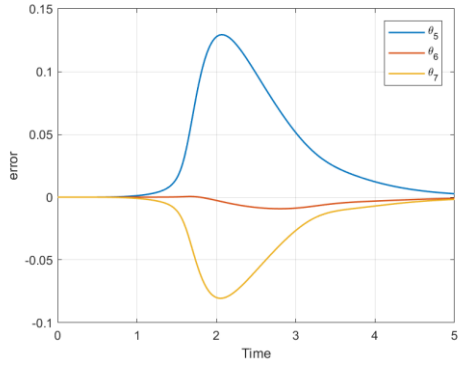


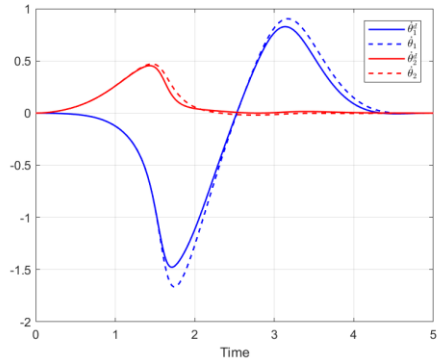


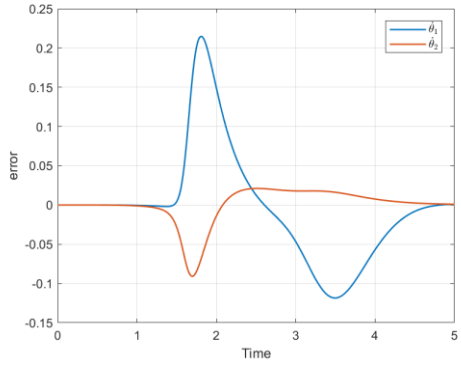

Time

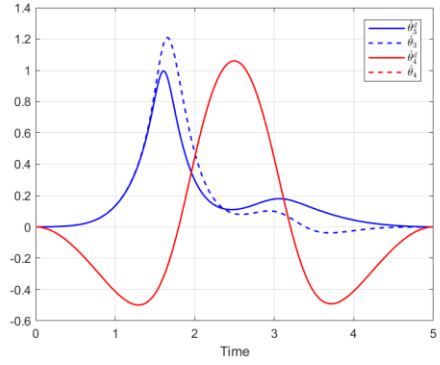

error
Time

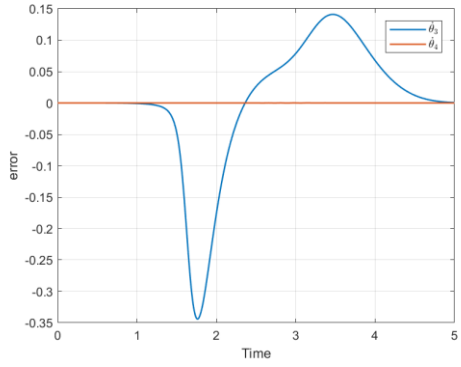

Time

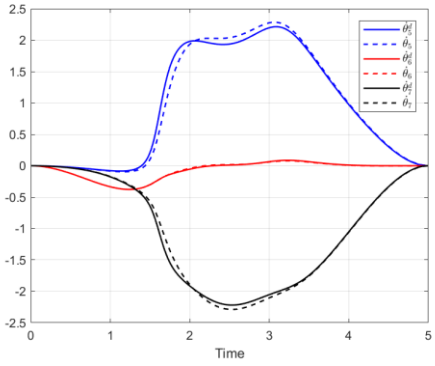

error
Time

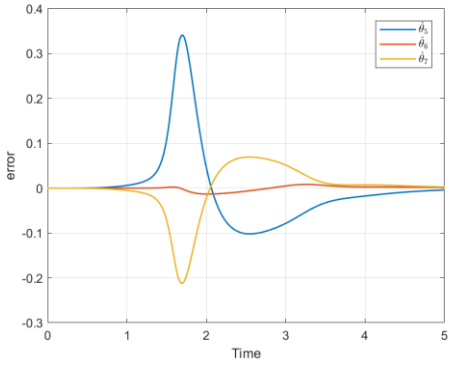

Time


error
Time

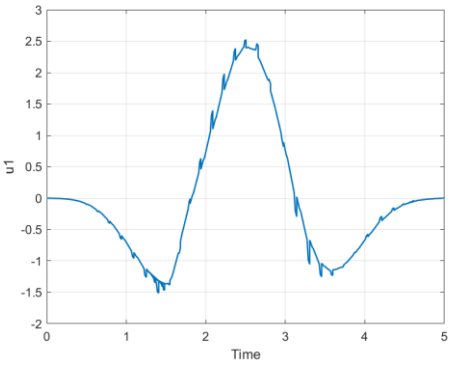

Time

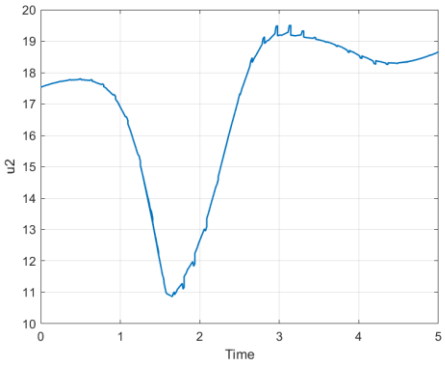

error
Time

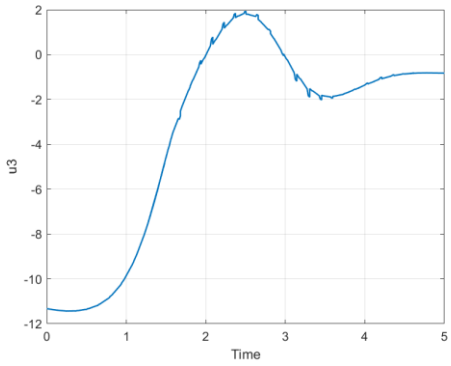


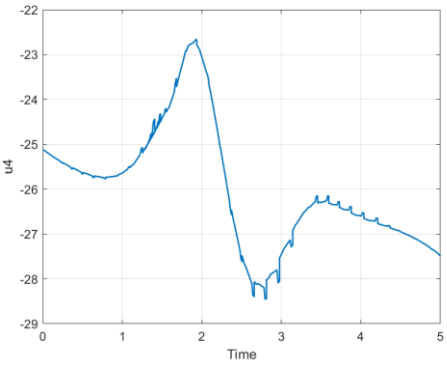


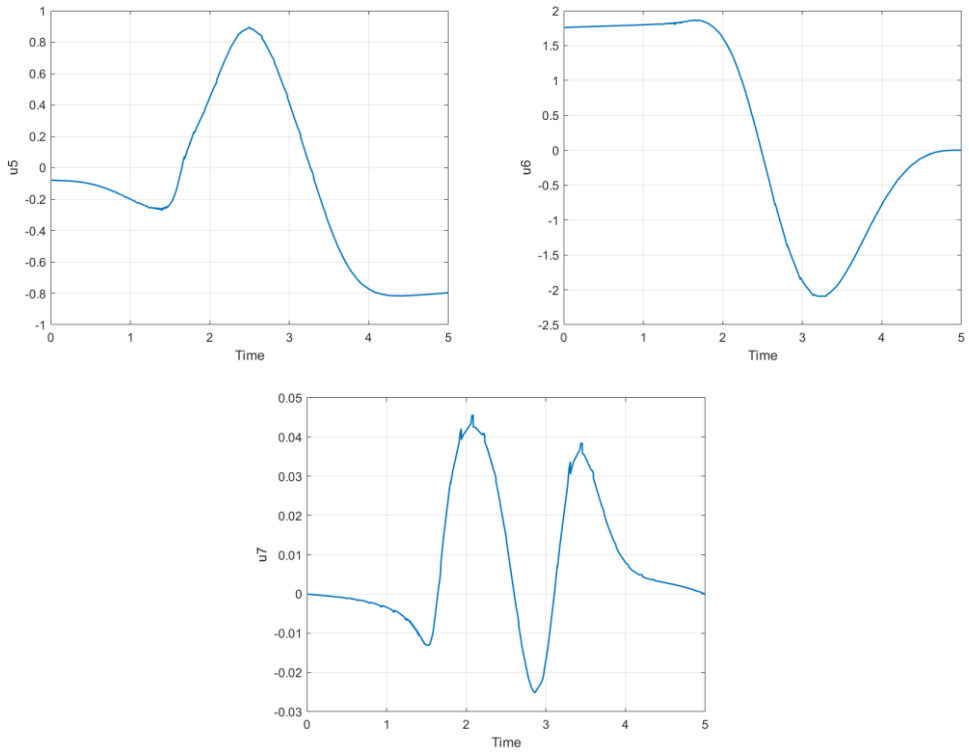
The input of the sliding controller is presented the figures below. The chattering still exist but it is much lower compared to case 2.



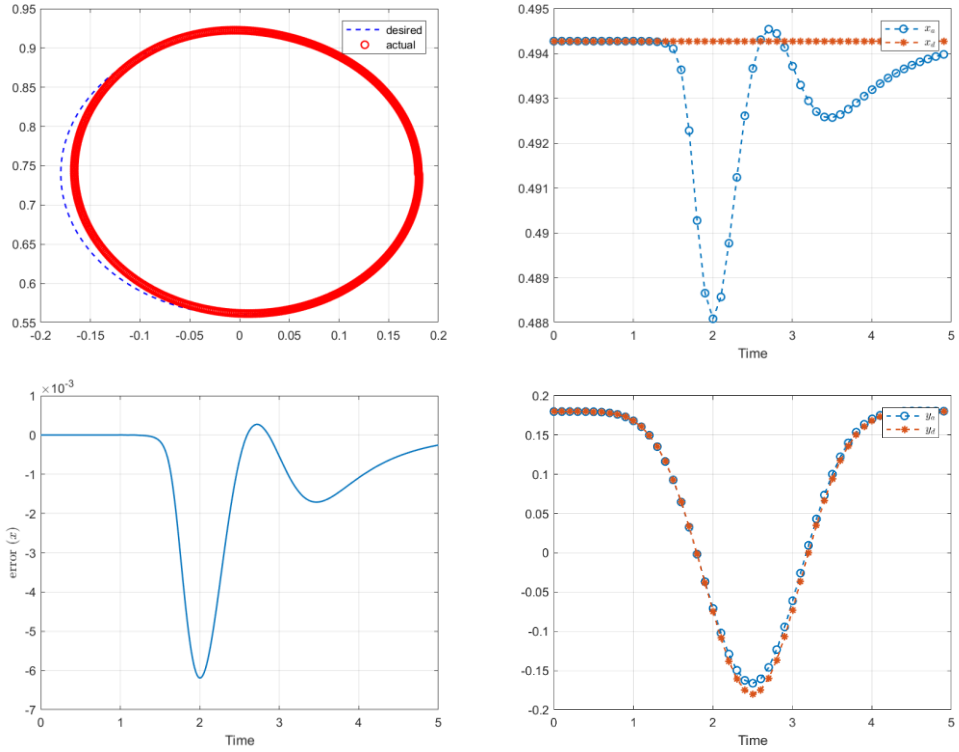


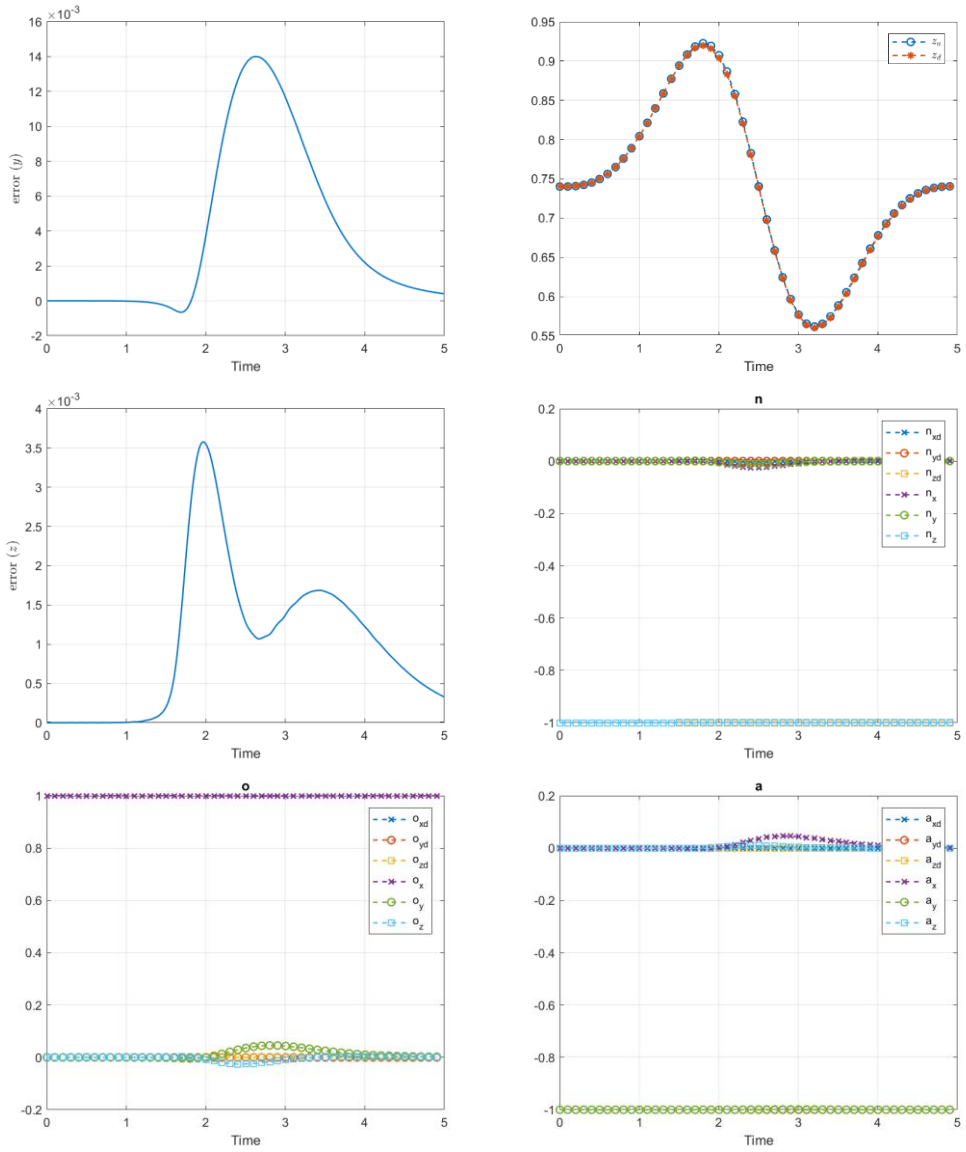






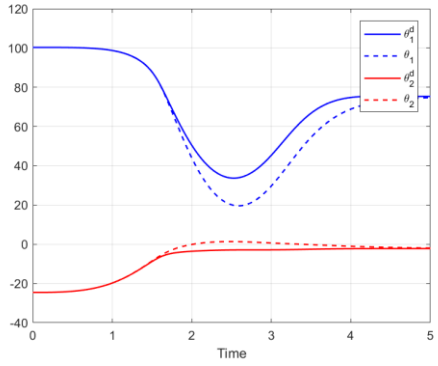
Finally, a comparison between the actual and the desired position and orientation of the end effector are shown below:

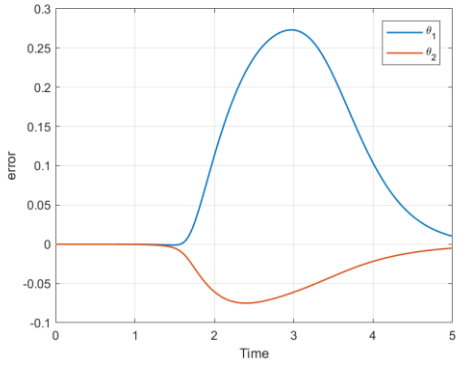


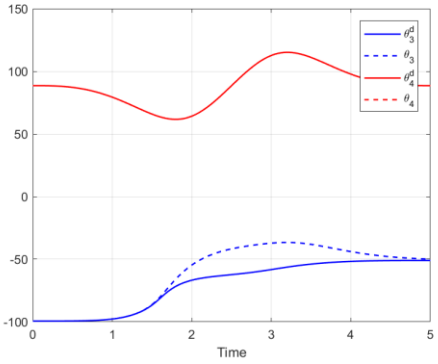


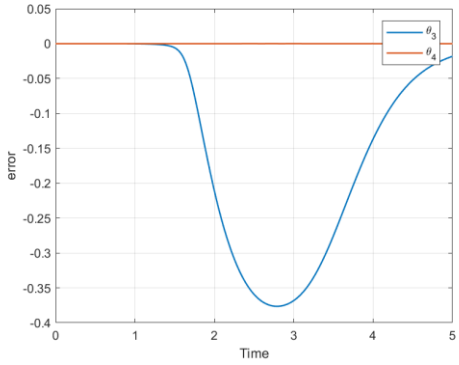
Case 4:

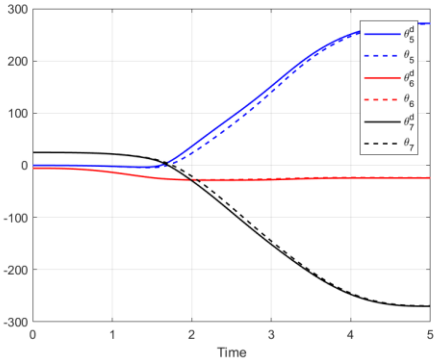
$$\epsilon = 5.0 \;\; \lambda = 1.5 \;\; \eta = 10$$

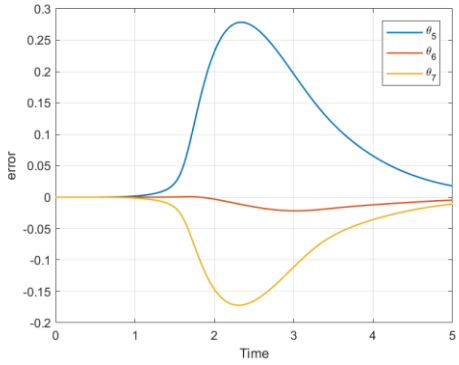

Time

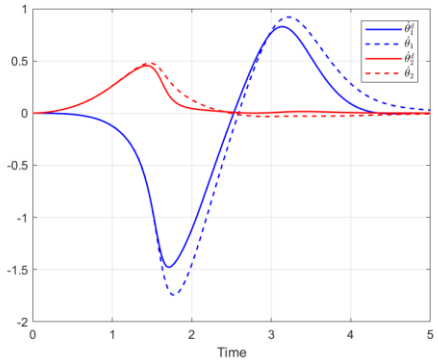

error
Time

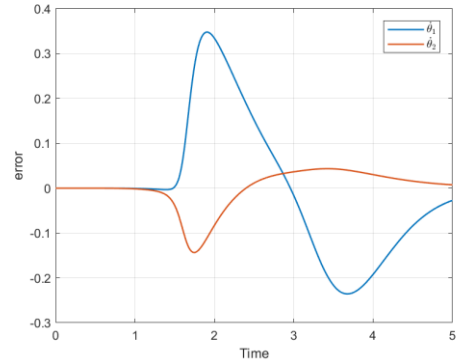

Time

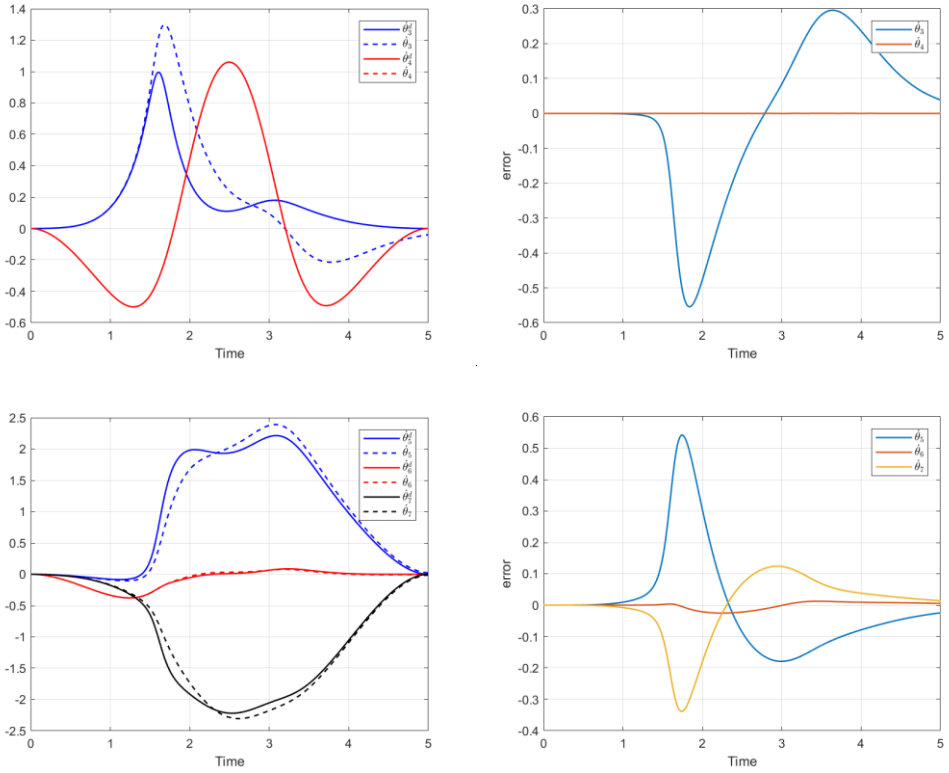

error
Time


Time

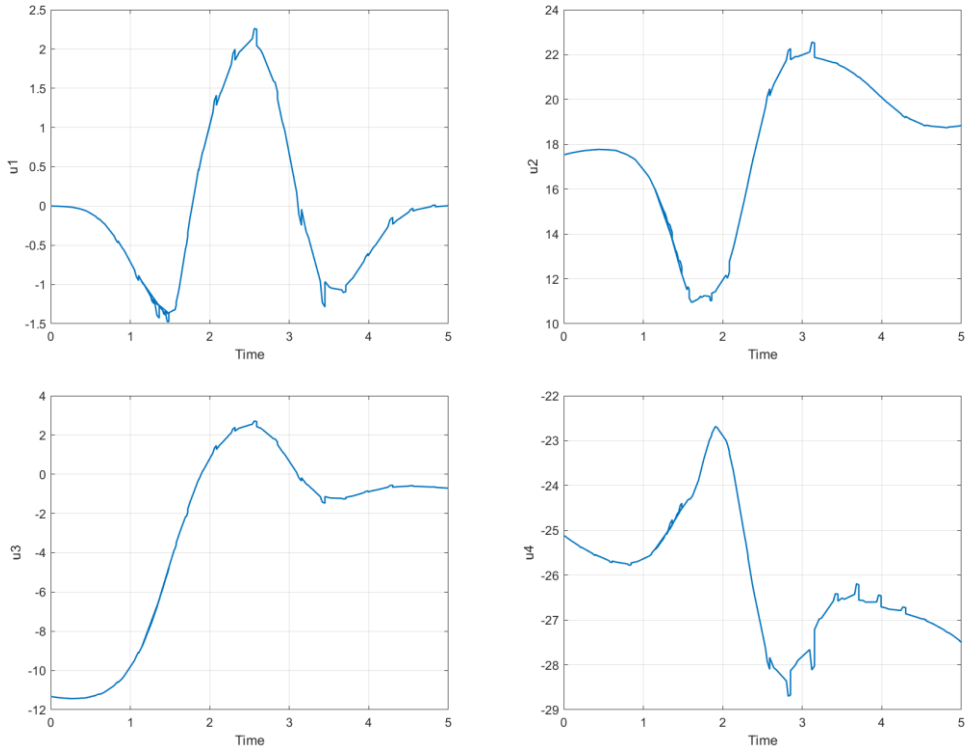

error
Time

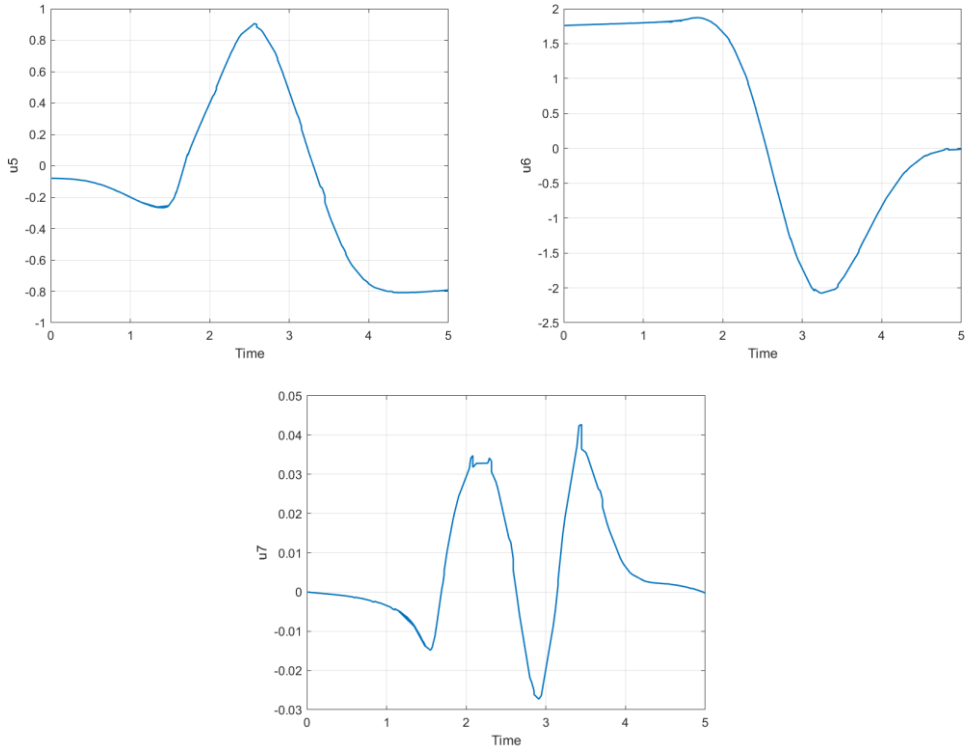

Time


error
Time

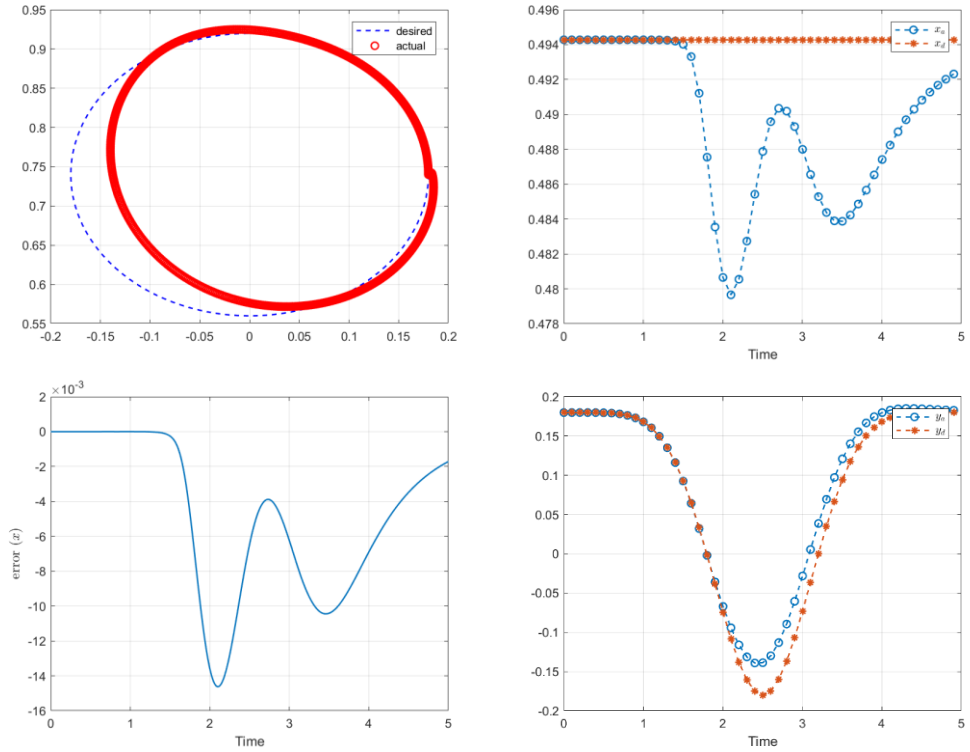


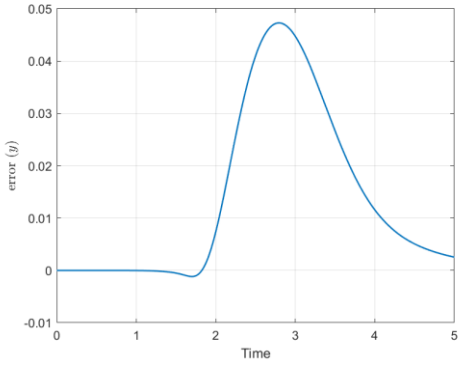
The input of the sliding controller is presented the figures below. The chattering still exist but it is much lower compared to case 3.

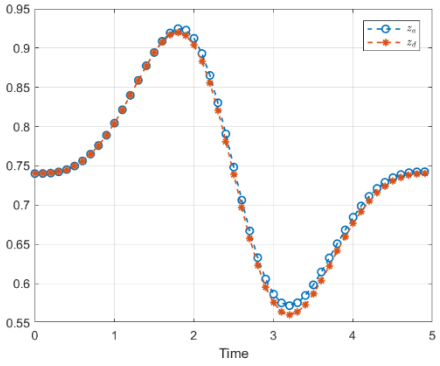


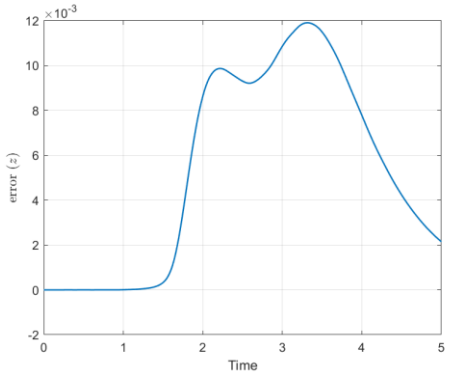


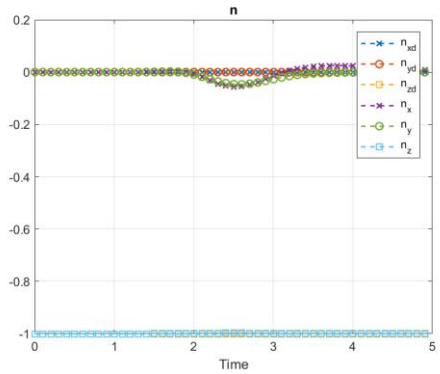
Finally, a comparison between the actual and the desired position and orientation of the end effector are shown below:

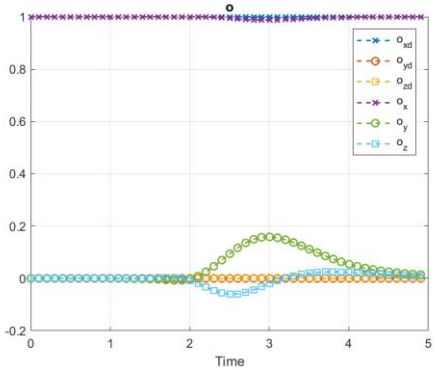


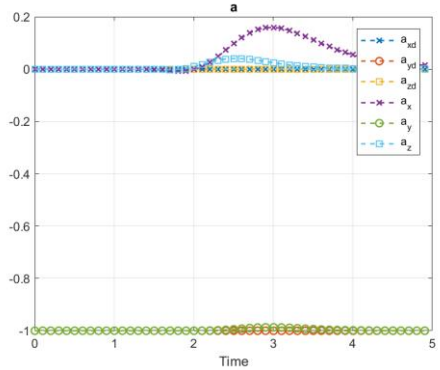

error (y)
Time


Time


×10-3
error (z)
Time


n
Time


o
Time


a
Time

Conclusion

In this project, iiwa 7 R800 robotic arm (7 degree of freedom manipulator) was selected for modeling and control. The kinematic modeling was derived for any set of joint variable. The dynamical model was derived symbolically. Then a proposed trajectory was tested for singularity. A circular trajectory was then selected to be tracked by the end effector with a fixed orientation. Using the desired trajectory, an inverse solution was found for each joint variable along with the velocity and the acceleration. Finally, a sliding control was used to achieve the desired values. Error of each variable was compared. Results indicated a good tracking using the sliding control, however chattering problem was observed. Increasing ϵ helped in eliminating the chattering problem to an acceptable extend but the error between the actual and desired was increased. Further investigation should consider: uncertainty, friction of joints, off diagonal mass moment of inertia.

Appendix

MATLAB Codes:

```
%% Solving the Inverse Kinematic problem for iiwa 7 arm manipulator
% Turki Haj Mohamad
% Villanova University
% 5/8/2018

%% cleaning
clear
clc
close all
tic
%% Desired path
plot_flag_cartesian = 1; %plot desired trajectory flag
plot_flag_joint = 1; %plot joint variables
traj_case =1; % selecting the desired trajectory, two options
folder_name = 'Figures_traj1'
fname1 = 'C:\Users\Admin\Dropbox (VCADS)\1SCH\Spring 2018\Robatics\Project term\iiwa 7\Matlab code\Desired_joints_inverse\';
fname = strcat(fname1,folder_name);
ifig = 0;
%% Desired Trajectory
Desired_creator;
%% Inverse Kinematics using Fsolve
Inverse_position

%% Inverse solution --> Forward solution
Forward_solution;

%% Inverse velocity and acceleration
Inverse_velocity;
```

```
%% Trajectory planning
% Turki Haj Mohamad
% Villanova University
% 5/8/2018
%% Initiating
global r xc yc zc a
DH_table
%% Trajectory specs
num_points = 1000;
t0 = 0; % start time
tf = 5; % final time
%% Trajectory options
if traj_case ==1
    r = (L3+L4)*sin(20*pi/180); % radius
    xc = (L3+L4)*cos(20*pi/180); % strat point X
    yc = 0;
    zc = L1+L2;
else
    r = 0.2;
    xc = (L3+L4); % strat point X
    yc = -r;
    zc = L1+L2;
end
%% Boundary Condition problem
% this equations to describe a circle, describing x and y using alpha
alpha_0 = 0; % initial angle
alpha_f = 2*pi; %final angle

alpha_dot0 = 0; % initial velocity
alpha_dotf = 0; %final velocity

alpha_ddot0 = 0; % initial acc.
alpha_ddotf = 0; %final acc.

a=coeff_poly5(t0,tf,alpha_0,alpha_f,alpha_dot0,alpha_dotf,alpha_ddot0,alpha_ddotf);
```

```
%% Desired end effector x,y,z and orientation
time = linspace(t0,tf,num_points);
disp_v = [1:20:length(time)];
% position
X = xdes(time);
Xdot = xdes_dot(time);
Xddot = xdes_ddot(time);

Y =  ydes(a,time);
Ydot = ydes_dot(a,time);
Yddot = ydes_ddot(a,time);

Z =  zdes(a,time);
Zdot = zdes_dot(a,time);
Zddot = zdes_ddot(a,time);


% orientation
nx = 0*ones(1,length(time));
ny = 0*ones(1,length(time));
nz = -1*ones(1,length(time));

ox = 1*ones(1,length(time));
oy = 0*ones(1,length(time));
oz = 0*ones(1,length(time));

ax = 0*ones(1,length(time));
ay = -1*ones(1,length(time));
az = 0*ones(1,length(time));

n_t = [nx;ny;nz];
o_t = [ox;oy;oz];
a_t = [ax;ay;az];

%% plotting and visulization
if plot_flag_cartesian == 1;
plot_desired
end
```

```
figure
plot3(X,Y,Z,'-k','linewidth',1.2)
xlabel('x_d'),ylabel('y_d'),zlabel('z_d')
grid on
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');


figure
plot(Y,Z,'-k','linewidth',1.2)
xlabel('y_d'),ylabel('z_d')
grid on
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');


figure
plot(time(disp_v),X(disp_v),'-','linewidth',1.2)
hold on
plot(time(disp_v),Xdot(disp_v),'--o','linewidth',1.2)
hold on
plot(time(disp_v),Xddot(disp_v),'--*','linewidth',1.2)
legend('position','velocity','acceleration')
xlabel('Time')
title('x_d')
grid on
ifig = ifig+1;
```

```
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');

figure
plot(time(disp_v),Y(disp_v),'linewidth',1.2)
hold on
plot(time(disp_v),Ydot(disp_v),'--o','linewidth',1.2)
hold on
plot(time(disp_v),Yddot(disp_v),'--*','linewidth',1.2)
legend('position','velocity','acceleration')
xlabel('Time')
title('y_d')
grid on
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');


figure
plot(time(disp_v),Z(disp_v),'linewidth',1.2)
hold on
plot(time(disp_v),Zdot(disp_v),'--o','linewidth',1.2)
hold on
plot(time(disp_v),Zddot(disp_v),'--*','linewidth',1.2)
legend('position','velocity','acceleration')
xlabel('Time')
title('z_{d}')
grid on
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');
```

```
% Initial guess assigning depending on the trajectory option
if traj_case ==1
theta0 = zeros(1,7);
theta0(1) = 20*pi/180;
theta0(4)=pi/2;
else
theta0 = zeros(1,7);
theta0(4)=pi/2;
end

Theta = zeros(7,length(time));

for i=1:length(time)
    cost_value(:,i) = fcost4(theta0,X(i),Y(i),Z(i),n_t(:,i),o_t(:,i),a_t(:,i));
    option  = optimset('Algorithm','levenberg-marquardt');
    fun = @(theta) fcost4(theta,X(i),Y(i),Z(i),n_t(:,i),o_t(:,i),a_t(:,i));
    theta_new = fsolve(fun,theta0,option);
    theta0 = theta_new;
    Theta(:,i)=theta0;
end


toc
if plot_flag_joint ==1
plot_desired_joint
end
```

```
figure
plot(time,Theta(1,:)*180/pi,'b','linewidth',1.2)
hold on
plot(time,Theta(2,:)*180/pi,'r','linewidth',1.2)
grid on
legend('\theta_1','\theta_2')
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
```

```
saveas(gca, fullfile(fname, filename), 'png');


figure
plot(time,Theta(3,:)*180/pi,'b','linewidth',1.2)
hold on
plot(time,Theta(4,:)*180/pi,'r','linewidth',1.2)
grid on
legend('\theta_3','\theta_4')
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');


figure
plot(time,Theta(5,:)*180/pi,'b','linewidth',1.2)
hold on
plot(time,Theta(6,:)*180/pi,'r','linewidth',1.2)
hold on
plot(time,Theta(7,:)*180/pi,'k','linewidth',1.2)
grid on
legend('\theta_5','\theta_6','\theta_7')
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');
```

```
pxd = zeros(1,length(time));
pyd = pxd;
pzd = pxd;

n_r = zeros(3,length(time));

o_r = n_r;

a_r = n_r;

for i=1:length(time)
    v = Theta(:,i);
   [pxd(i),pyd(i),pzd(i),n_r(:,i),o_r(:,i),a_r(:,i)] = ff_fun(v);
end

figure
plot(Y,Z,'b--','linewidth',1.2)
grid on
hold on
plot(pyd,pzd,'ro','linewidth',1.2)
legend('desired','actual')
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');

figure
plot(time(disp_v),pxd(disp_v),'--o','linewidth',1.2)
hold on
plot(time(disp_v),X(disp_v),'--*','linewidth',1.2)
grid on
hl = legend('$x_a$','$x_d$');
set(hl, 'Interpreter', 'latex');
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');

figure
plot(time(disp_v),pyd(disp_v),'--o','linewidth',1.2)
hold on
plot(time(disp_v),Y(disp_v),'--*','linewidth',1.2)
hl = legend('$y_a$','$y_d$');
set(hl, 'Interpreter', 'latex');
grid on
```

```
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');

figure
plot(time(disp_v),pzd(disp_v),'--o','linewidth',1.2)
hold on
plot(time(disp_v),Z(disp_v),'--*','linewidth',1.2)
grid on
hl = legend('$z_a$','$z_d$');
set(hl, 'Interpreter', 'latex');
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');



figure
plot(time(disp_v),n_t(1,disp_v)','--x','linewidth',1.2)
hold on
plot(time(disp_v),n_t(2,disp_v)','--o','linewidth',1.2)
hold on
plot(time(disp_v),n_t(3,disp_v)','--s','linewidth',1.2)
hold on
plot(time(disp_v),n_r(1,disp_v)','--x','linewidth',1.2)
hold on
plot(time(disp_v),n_r(2,disp_v)','--o','linewidth',1.2)
hold on
plot(time(disp_v),n_r(3,disp_v)','--s','linewidth',1.2)
legend('n_{xd}','n_{yd}','n_{zd}','n_x','n_y','n_z')
title('n')
grid on
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');

figure
plot(time(disp_v),o_t(1,disp_v)','--x','linewidth',1.2)
hold on
plot(time(disp_v),o_t(2,disp_v)','--o','linewidth',1.2)
hold on
plot(time(disp_v),o_t(3,disp_v)','--s','linewidth',1.2)
hold on
plot(time(disp_v),o_r(1,disp_v)','--x','linewidth',1.2)
hold on
plot(time(disp_v),o_r(2,disp_v)','--o','linewidth',1.2)
hold on
plot(time(disp_v),o_r(3,disp_v)','--s','linewidth',1.2)
legend('o_{xd}','o_{yd}','o_{zd}','o_x','o_y','o_z')
title('o')
grid on
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');

figure
plot(time(disp_v),a_t(1,disp_v)','--x','linewidth',1.2)
hold on
plot(time(disp_v),a_t(2,disp_v)','--o','linewidth',1.2)
hold on
plot(time(disp_v),a_t(3,disp_v)','--s','linewidth',1.2)
hold on
plot(time(disp_v),a_r(1,disp_v)','--x','linewidth',1.2)
hold on
plot(time(disp_v),a_r(2,disp_v)','--o','linewidth',1.2)
hold on
plot(time(disp_v),a_r(3,disp_v)','--s','linewidth',1.2)
legend('a_{xd}','a_{yd}','a_{zd}','a_x','a_y','a_z')
title('a')
grid on
```

```
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');
```

```
Theta_dot = zeros(7,length(time));
Theta_ddot = zeros(7,length(time));
actual_vel = zeros(6,length(time));
actual_acc = zeros(6,length(time));



for i=1:length(time)
    % Velocities
    J = iwwa_J(Theta(:,i));
    desired_velocity = [Xdot(i);Ydot(i);Zdot(i);0;0;0];
%     Theta_dot(:,i) = pinv(J)*desired_velocity;
    pinvmat = transpose(J)*inv(J*transpose(J));
    Theta_dot(:,i) = pinvmat*desired_velocity;
    v_det(i) = det(J*transpose(J));
    % Acceleration
    desired_acc = [Xddot(i);Yddot(i);Zddot(i);0;0;0];
    Jdot = iwwa_Jdot(Theta(:,i),Theta_dot(:,i));
%     Theta_ddot(:,i) = pinv(J)*(desired_acc-Jdot*Theta_dot(:,i));
    Theta_ddot(:,i) = pinvmat*(desired_acc-Jdot*Theta_dot(:,i));
    actual_vel(:,i) = J*Theta_dot(:,i);
    actual_acc(:,i) = J*Theta_ddot(:,i) + Jdot *Theta_dot(:,i);
end


% using only Jv
% for i=1:length(time)
%     % Velocities
% % %     J = iwwa_Jac(Theta(:,i));
% % %     Jv = J(1:3,:);
%     Jv = iwwa_Jv(Theta(:,i));
%     desired_velocity = [Xdot(i);Ydot(i);Zdot(i)];
% % %     Theta_dot(:,i) = pinv(Jv)*desired_velocity;
%     pinvmat = transpose(Jv)*inv(Jv*transpose(Jv));
%     Theta_dot(:,i) = pinvmat*desired_velocity;
%
%     % Acceleration
%     desired_acc = [Xddot(i);Yddot(i);Zddot(i)];
%     Jvdot = iwwa_Jvdot(Theta(:,i),Theta_dot(:,i));
% % %     Theta_ddot(:,i) = pinv(Jv)*(desired_acc-Jvdot*Theta_dot(:,i));
%     Theta_ddot(:,i) = pinvmat*(desired_acc-Jvdot*Theta_dot(:,i));
%
%     actual_vel(:,i) = Jv*Theta_dot(:,i);
%     actual_acc(:,i) = Jv*Theta_ddot(:,i) + Jvdot *Theta_dot(:,i);
% end

plot_desired_veolocity
```

```
figure
plot(time,Theta_dot(1,:),'.b','linewidth',1.2)
hold on
plot(time,Theta_dot(2,:),'.r','linewidth',1.2)
grid on
hl = legend('$\dot{\theta}_{1}$','$\dot{\theta}_{2}$');
set(hl, 'Interpreter', 'latex');
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');


figure
plot(time,Theta_dot(3,:),'.b','linewidth',1.2)
```

```
hold on
plot(time,Theta_dot(4,:),'.r','linewidth',1.2)
grid on
hl = legend('$\dot{\theta}_{3}$','$\dot{\theta}_{4}$');
set(hl, 'Interpreter', 'latex');
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');

figure
plot(time,Theta_dot(5,:),'.b','linewidth',1.2)
hold on
plot(time,Theta_dot(6,:),'.r','linewidth',1.2)
hold on
plot(time,Theta_dot(7,:),'.k','linewidth',1.2)
grid on
hl = legend('$\dot{\theta}_{5}$','$\dot{\theta}_{6}$','$\dot{\theta}_{7}$');
set(hl, 'Interpreter', 'latex');
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');


figure
plot(time,Theta_ddot(1,:),'.b','linewidth',1.2)
hold on
plot(time,Theta_ddot(2,:),'.r','linewidth',1.2)
grid on
hl = legend('$\ddot{\theta}_{1}$','$\ddot{\theta}_{2}$');
set(hl, 'Interpreter', 'latex');
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');

figure
plot(time,Theta_ddot(3,:),'.b','linewidth',1.2)
hold on
plot(time,Theta_ddot(4,:),'.r','linewidth',1.2)
grid on
hl = legend('$\ddot{\theta}_{3}$','$\ddot{\theta}_{4}$');
set(hl, 'Interpreter', 'latex');
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');

figure
plot(time,Theta_ddot(5,:),'.b','linewidth',1.2)
hold on
plot(time,Theta_ddot(6,:),'.r','linewidth',1.2)
hold on
plot(time,Theta_ddot(7,:),'.k','linewidth',1.2)
grid on
hl = legend('$\ddot{\theta}_{5}$','$\ddot{\theta}_{6}$','$\ddot{\theta}_{7}$');
set(hl, 'Interpreter', 'latex');
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');


figure
plot(time(disp_v),Xdot(disp_v),'--o','linewidth',1.2)
hold on
plot(time(disp_v),actual_vel(1,disp_v),'--*','linewidth',1.2)
grid on
hl = legend('$\dot{x}_{d}$','$\dot{x}_{a}$');
set(hl, 'Interpreter', 'latex');
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');
```

```
figure
plot(time(disp_v),Ydot(disp_v),'--o','linewidth',1.2)
hold on
plot(time(disp_v),actual_vel(2,disp_v),'--*','linewidth',1.2)
grid on
hl = legend('$\dot{y}_{d}$','$\dot{y}_{a}$');
set(hl, 'Interpreter', 'latex');
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');

figure
plot(time(disp_v),Zdot(disp_v),'--o','linewidth',1.2)
hold on
plot(time(disp_v),actual_vel(3,disp_v),'--*','linewidth',1.2)
grid on
hl = legend('$\dot{z}_{d}$','$\dot{z}_{a}$');
set(hl, 'Interpreter', 'latex');
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');

figure
plot(time(disp_v),Xddot(disp_v),'--o','linewidth',1.2)
hold on
plot(time(disp_v),actual_acc(1,disp_v),'--*','linewidth',1.2)
grid on
hl = legend('$\ddot{x}_{d}$','$\ddot{x}_{a}$');
set(hl, 'Interpreter', 'latex');
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');

figure
plot(time(disp_v),Yddot(disp_v),'--o','linewidth',1.2)
hold on
plot(time(disp_v),actual_acc(2,disp_v),'--*','linewidth',1.2)
grid on
hl = legend('$\ddot{y}_{d}$','$\ddot{y}_{a}$');
set(hl, 'Interpreter', 'latex');
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');

figure
plot(time(disp_v),Zddot(disp_v),'--o','linewidth',1.2)
hold on
plot(time(disp_v),actual_acc(3,disp_v),'--*','linewidth',1.2)
grid on
hl = legend('$\ddot{z}_{d}$','$\ddot{z}_{a}$');
set(hl, 'Interpreter', 'latex');
ifig = ifig+1;
filename = strcat('fig',num2str(ifig));
saveas(gca, fullfile(fname, filename), 'png');
```

```
function J = iwwa_J(v)

th1 = v(1);
th2 = v(2);
th3 = v(3);
th4 = v(4);
th5 = v(5);
th6 = v(6);
th7 = v(7);
 L1 = 0.34;
L2 = 0.4;
L3 = 0.4;
L4 = 0.126;
```

```
J = [   L2*sin(th1)*sin(th2) - L3*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)) -
L4*(cos(th6)*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)) -
sin(th6)*(cos(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th1)*cos(th3) -
cos(th2)*sin(th1)*sin(th3)))), - L4*(cos(th1)*sin(th6)*(sin(th2)*sin(th3)*sin(th5) + cos(th2)*cos(th5)*sin(th4) -
cos(th3)*cos(th4)*cos(th5)*sin(th2)) + cos(th1)*cos(th6)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4))) - L2*cos(th1)*cos(th2) -
L3*cos(th1)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4)), - L4*(sin(th6)*(sin(th5)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) -
cos(th4)*cos(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) + cos(th6)*sin(th4)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) -
L3*sin(th4)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3)), - L4*(cos(th6)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) -
cos(th1)*sin(th2)*sin(th4)) + cos(th5)*sin(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2))) -
L3*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)), -
L4*sin(th6)*(sin(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) - cos(th5)*(cos(th3)*sin(th1) +
cos(th1)*cos(th2)*sin(th3))),  L4*(sin(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2)) +
cos(th6)*(cos(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th3)*sin(th1) +
cos(th1)*cos(th2)*sin(th3)))),
0;
 - L4*(cos(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2)) -
sin(th6)*(cos(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th3)*sin(th1) +
cos(th1)*cos(th2)*sin(th3)))) - L3*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2)) -
L2*cos(th1)*sin(th2), - L4*(sin(th1)*sin(th6)*(sin(th2)*sin(th3)*sin(th5) + cos(th2)*cos(th5)*sin(th4) - cos(th3)*cos(th4)*cos(th5)*sin(th2))
+ cos(th6)*sin(th1)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4))) - L3*sin(th1)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4)) -
L2*cos(th2)*sin(th1),   L4*(sin(th6)*(sin(th5)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - cos(th4)*cos(th5)*(cos(th1)*cos(th3) -
cos(th2)*sin(th1)*sin(th3))) + cos(th6)*sin(th4)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) + L3*sin(th4)*(cos(th1)*cos(th3) -
cos(th2)*sin(th1)*sin(th3)),   L4*(cos(th6)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) +
cos(th5)*sin(th6)*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2))) + L3*(cos(th4)*(cos(th1)*sin(th3)
+ cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)),  L4*sin(th6)*(sin(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1))
+ sin(th1)*sin(th2)*sin(th4)) - cos(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))), -L4*(sin(th6)*(sin(th4)*(cos(th1)*sin(th3) +
cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)) + cos(th6)*(cos(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) +
sin(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3)))),
0;

0,                                  - L4*(sin(th6)*(cos(th5)*(sin(th2)*sin(th4) + cos(th2)*cos(th3)*cos(th4)) - cos(th2)*sin(th3)*sin(th5)) +
cos(th6)*(cos(th4)*sin(th2) - cos(th2)*cos(th3)*sin(th4))) - L3*(cos(th4)*sin(th2) - cos(th2)*cos(th3)*sin(th4)) - L2*sin(th2),
L4*(sin(th6)*(cos(th3)*sin(th2)*sin(th5) + cos(th4)*cos(th5)*sin(th2)*sin(th3)) - cos(th6)*sin(th2)*sin(th3)*sin(th4)) -
L3*sin(th2)*sin(th3)*sin(th4),                                                                                                  - L3*(cos(th2)*sin(th4) -
cos(th3)*cos(th4)*sin(th2)) - L4*(cos(th6)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) - cos(th5)*sin(th6)*(cos(th2)*cos(th4) +
cos(th3)*sin(th2)*sin(th4))),                                    -L4*sin(th6)*(sin(th5)*(cos(th2)*sin(th4) -
cos(th3)*cos(th4)*sin(th2)) - cos(th5)*sin(th2)*sin(th3)),
L4*(cos(th6)*(cos(th5)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) + sin(th2)*sin(th3)*sin(th5)) - sin(th6)*(cos(th2)*cos(th4) +
cos(th3)*sin(th2)*sin(th4))),
0;

0,
sin(th1),
-cos(th1)*sin(th2),
- cos(th3)*sin(th1) - cos(th1)*cos(th2)*sin(th3),                                                                  - sin(th4)*(sin(th1)*sin(th3) -
cos(th1)*cos(th2)*cos(th3)) - cos(th1)*cos(th4)*sin(th2),
cos(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3)) - sin(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) -
cos(th1)*sin(th2)*sin(th4)), sin(th6)*(cos(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) +
sin(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) - cos(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) +
cos(th1)*cos(th4)*sin(th2));

0,
-cos(th1),
-sin(th1)*sin(th2),
cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3),                                                                   sin(th4)*(cos(th1)*sin(th3) +
cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2),
sin(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) - cos(th5)*(cos(th1)*cos(th3) -
cos(th2)*sin(th1)*sin(th3)), cos(th6)*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)) -
sin(th6)*(cos(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th1)*cos(th3) -
cos(th2)*sin(th1)*sin(th3)));

1,
0,
cos(th2),
-sin(th2)*sin(th3),                                                                                   cos(th2)*cos(th4) +
cos(th3)*sin(th2)*sin(th4),
cos(th5)*sin(th2)*sin(th3) - sin(th5)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)),
sin(th6)*(cos(th5)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) + sin(th2)*sin(th3)*sin(th5)) + cos(th6)*(cos(th2)*cos(th4) +
cos(th3)*sin(th2)*sin(th4))];
```

```
function Jdot = iwwa_Jdot(v,vdot)

th1 = v(1);
th2 = v(2);
th3 = v(3);
th4 = v(4);
th5 = v(5);
th6 = v(6);
th7 = v(7);

th1dot = vdot(1);
th2dot = vdot(2);
th3dot = vdot(3);
th4dot = vdot(4);
th5dot = vdot(5);
th6dot = vdot(6);
th7dot = vdot(7);

 L1 = 0.34;
L2 = 0.4;
L3 = 0.4;
L4 = 0.126;


Jdot = [ th2dot*(L4*(sin(th6)*(cos(th5)*(cos(th2)*sin(th1)*sin(th4) - cos(th3)*cos(th4)*sin(th1)*sin(th2)) +
sin(th1)*sin(th2)*sin(th3)*sin(th5)) + cos(th6)*(cos(th2)*cos(th4)*sin(th1) + cos(th3)*sin(th1)*sin(th2)*sin(th4))) +
L3*(cos(th2)*cos(th4)*sin(th1) + cos(th3)*sin(th1)*sin(th2)*sin(th4)) + L2*cos(th2)*sin(th1)) +
th1dot*(L4*(cos(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2)) -
sin(th6)*(cos(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th3)*sin(th1) +
cos(th1)*cos(th2)*sin(th3)))) + L3*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2)) +
L2*cos(th1)*sin(th2)) - th4dot*(L4*(cos(th6)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) +
cos(th5)*sin(th6)*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2))) + L3*(cos(th4)*(cos(th1)*sin(th3)
+ cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4))) - th3dot*(L4*(sin(th6)*(sin(th5)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1))
- cos(th4)*cos(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) + cos(th6)*sin(th4)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) +
L3*sin(th4)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) + L4*th6dot*(sin(th6)*(sin(th4)*(cos(th1)*sin(th3) +
cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)) + cos(th6)*(cos(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) +
sin(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3)))) -
L4*th5dot*sin(th6)*(sin(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) -
cos(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))), th2dot*(L4*(cos(th1)*sin(th6)*(cos(th5)*sin(th2)*sin(th4) -
cos(th2)*sin(th3)*sin(th5) + cos(th2)*cos(th3)*cos(th4)*cos(th5)) + cos(th1)*cos(th6)*(cos(th4)*sin(th2) - cos(th2)*cos(th3)*sin(th4))) +
L2*cos(th1)*sin(th2) + L3*cos(th1)*(cos(th4)*sin(th2) - cos(th2)*cos(th3)*sin(th4))) -
th3dot*(L4*(cos(th1)*sin(th6)*(cos(th3)*sin(th2)*sin(th5) + cos(th4)*cos(th5)*sin(th2)*sin(th3)) -
cos(th1)*cos(th6)*sin(th2)*sin(th3)*sin(th4)) - L3*cos(th1)*sin(th2)*sin(th3)*sin(th4)) +
th1dot*(L4*(sin(th1)*sin(th6)*(sin(th2)*sin(th3)*sin(th5) + cos(th2)*cos(th5)*sin(th4) - cos(th3)*cos(th4)*cos(th5)*sin(th2)) +
cos(th6)*sin(th1)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4))) + L3*sin(th1)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4)) +
L2*cos(th2)*sin(th1)) - th4dot*(L4*(cos(th1)*sin(th6)*(cos(th2)*cos(th4)*cos(th5) + cos(th3)*cos(th5)*sin(th2)*sin(th4)) -
cos(th1)*cos(th6)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2))) - L3*cos(th1)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2))) -
L4*th6dot*(cos(th1)*cos(th6)*(sin(th2)*sin(th3)*sin(th5) + cos(th2)*cos(th5)*sin(th4) - cos(th3)*cos(th4)*cos(th5)*sin(th2)) -
cos(th1)*sin(th6)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4))) - L4*th5dot*cos(th1)*sin(th6)*(cos(th5)*sin(th2)*sin(th3) -
cos(th2)*sin(th4)*sin(th5) + cos(th3)*cos(th4)*sin(th2)*sin(th5)), - th4dot*(L4*(cos(th4)*cos(th6)*(cos(th3)*sin(th1) +
cos(th1)*cos(th2)*sin(th3)) + cos(th5)*sin(th4)*sin(th6)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) + L3*cos(th4)*(cos(th3)*sin(th1)
+ cos(th1)*cos(th2)*sin(th3))) - th2dot*(L4*(sin(th6)*(cos(th1)*cos(th3)*sin(th2)*sin(th5) + cos(th1)*cos(th4)*cos(th5)*sin(th2)*sin(th3)) -
cos(th1)*cos(th6)*sin(th2)*sin(th3)*sin(th4)) - L3*cos(th1)*sin(th2)*sin(th3)*sin(th4)) - th1dot*(L4*(sin(th6)*(sin(th5)*(cos(th1)*sin(th3) +
cos(th2)*cos(th3)*sin(th1)) - cos(th4)*cos(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) + cos(th6)*sin(th4)*(cos(th1)*cos(th3) -
cos(th2)*sin(th1)*sin(th3))) + L3*sin(th4)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) -
th3dot*(L4*(sin(th6)*(sin(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3)) + cos(th4)*cos(th5)*(sin(th1)*sin(th3) -
cos(th1)*cos(th2)*cos(th3))) - cos(th6)*sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3))) - L3*sin(th4)*(sin(th1)*sin(th3) -
cos(th1)*cos(th2)*cos(th3))) - L4*th6dot*(cos(th6)*(sin(th5)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) -
cos(th4)*cos(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) - sin(th4)*sin(th6)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) -
L4*th5dot*sin(th6)*(cos(th5)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th4)*sin(th5)*(cos(th3)*sin(th1) +
cos(th1)*cos(th2)*sin(th3))), th4dot*(L4*(cos(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2)) -
cos(th5)*sin(th6)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4))) + L3*(sin(th4)*(sin(th1)*sin(th3) -
cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2))) - th3dot*(L4*(cos(th4)*cos(th6)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))
+ cos(th5)*sin(th4)*sin(th6)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) + L3*cos(th4)*(cos(th3)*sin(th1) +
cos(th1)*cos(th2)*sin(th3))) - th1dot*(L4*(cos(th6)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4))
+ cos(th5)*sin(th6)*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2))) +
L3*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4))) +
th2dot*(L4*(cos(th6)*(cos(th1)*cos(th2)*sin(th4) - cos(th1)*cos(th3)*cos(th4)*sin(th2)) - cos(th5)*sin(th6)*(cos(th1)*cos(th2)*cos(th4) +
cos(th1)*cos(th3)*sin(th2)*sin(th4))) + L3*(cos(th1)*cos(th2)*sin(th4) - cos(th1)*cos(th3)*cos(th4)*sin(th2))) +
```

```
L4*th6dot*(sin(th6)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) -
cos(th5)*cos(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2))) +
L4*th5dot*sin(th5)*sin(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2)),
L4*th2dot*sin(th6)*(sin(th5)*(cos(th1)*cos(th2)*sin(th4) - cos(th1)*cos(th3)*cos(th4)*sin(th2)) - cos(th1)*cos(th5)*sin(th2)*sin(th3)) -
L4*th5dot*sin(th6)*(cos(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) +
sin(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) - L4*th1dot*sin(th6)*(sin(th5)*(cos(th4)*(cos(th1)*sin(th3) +
cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) - cos(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) -
L4*th3dot*sin(th6)*(cos(th5)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th4)*sin(th5)*(cos(th3)*sin(th1) +
cos(th1)*cos(th2)*sin(th3))) - L4*th6dot*cos(th6)*(sin(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) -
cos(th1)*sin(th2)*sin(th4)) - cos(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) +
L4*th4dot*sin(th5)*sin(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2)),
L4*th6dot*(cos(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2)) -
sin(th6)*(cos(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th3)*sin(th1) +
cos(th1)*cos(th2)*sin(th3)))) + L4*th1dot*(sin(th6)*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)) +
cos(th6)*(cos(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th1)*cos(th3) -
cos(th2)*sin(th1)*sin(th3)))) + L4*th2dot*(sin(th6)*(cos(th1)*cos(th2)*cos(th4) + cos(th1)*cos(th3)*sin(th2)*sin(th4)) -
cos(th6)*(cos(th5)*(cos(th1)*cos(th2)*sin(th4) - cos(th1)*cos(th3)*cos(th4)*sin(th2)) + cos(th1)*sin(th2)*sin(th3)*sin(th5))) -
L4*th3dot*(cos(th6)*(sin(th5)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th4)*cos(th5)*(cos(th3)*sin(th1) +
cos(th1)*cos(th2)*sin(th3))) - sin(th4)*sin(th6)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) +
L4*th4dot*(sin(th6)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) -
cos(th5)*cos(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2))) -
L4*th5dot*cos(th6)*(sin(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) -
cos(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))),
0;
 L4*th6dot*(sin(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2)) +
cos(th6)*(cos(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th3)*sin(th1) +
cos(th1)*cos(th2)*sin(th3)))) - th1dot*(L4*(cos(th6)*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)) -
sin(th6)*(cos(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th1)*cos(th3) -
cos(th2)*sin(th1)*sin(th3)))) + L3*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)) -
L2*sin(th1)*sin(th2)) - th2dot*(L4*(cos(th6)*(cos(th1)*cos(th2)*cos(th4) + cos(th1)*cos(th3)*sin(th2)*sin(th4)) +
sin(th6)*(cos(th5)*(cos(th1)*cos(th2)*sin(th4) - cos(th1)*cos(th3)*cos(th4)*sin(th2)) + cos(th1)*sin(th2)*sin(th3)*sin(th5))) +
L3*(cos(th1)*cos(th2)*cos(th4) + cos(th1)*cos(th3)*sin(th2)*sin(th4)) + L2*cos(th1)*cos(th2)) -
th3dot*(L4*(sin(th6)*(sin(th5)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th4)*cos(th5)*(cos(th3)*sin(th1) +
cos(th1)*cos(th2)*sin(th3))) + cos(th6)*sin(th4)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) + L3*sin(th4)*(cos(th3)*sin(th1) +
cos(th1)*cos(th2)*sin(th3))) - th4dot*(L4*(cos(th6)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) +
cos(th5)*sin(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2))) + L3*(cos(th4)*(sin(th1)*sin(th3)
- cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4))) - L4*th5dot*sin(th6)*(sin(th5)*(cos(th4)*(sin(th1)*sin(th3) -
cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) - cos(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))),
th2dot*(L4*(sin(th1)*sin(th6)*(cos(th5)*sin(th2)*sin(th4) - cos(th2)*sin(th3)*sin(th5) + cos(th2)*cos(th3)*cos(th4)*cos(th5)) +
cos(th6)*sin(th1)*(cos(th4)*sin(th2) - cos(th2)*cos(th3)*sin(th4))) + L3*sin(th1)*(cos(th4)*sin(th2) - cos(th2)*cos(th3)*sin(th4)) +
L2*sin(th1)*sin(th2)) - th1dot*(L4*(cos(th1)*sin(th6)*(sin(th2)*sin(th3)*sin(th5) + cos(th2)*cos(th5)*sin(th4) -
cos(th3)*cos(th4)*cos(th5)*sin(th2)) + cos(th1)*cos(th6)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4))) + L2*cos(th1)*cos(th2) +
L3*cos(th1)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4))) - th4dot*(L4*(sin(th1)*sin(th6)*(cos(th2)*cos(th4)*cos(th5) +
cos(th3)*cos(th5)*sin(th2)*sin(th4)) - cos(th6)*sin(th1)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2))) - L3*sin(th1)*(cos(th2)*sin(th4) -
cos(th3)*cos(th4)*sin(th2))) - th3dot*(L4*(sin(th1)*sin(th6)*(cos(th3)*sin(th2)*sin(th5) + cos(th4)*cos(th5)*sin(th2)*sin(th3)) -
cos(th6)*sin(th1)*sin(th2)*sin(th3)*sin(th4)) - L3*sin(th1)*sin(th2)*sin(th3)*sin(th4)) -
L4*th6dot*(cos(th6)*sin(th1)*(sin(th2)*sin(th3)*sin(th5) + cos(th2)*cos(th5)*sin(th4) - cos(th3)*cos(th4)*cos(th5)*sin(th2)) -
sin(th1)*sin(th6)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4))) - L4*th5dot*sin(th1)*sin(th6)*(cos(th5)*sin(th2)*sin(th3) -
cos(th2)*sin(th4)*sin(th5) + cos(th3)*cos(th4)*sin(th2)*sin(th5)),   th4dot*(L4*(cos(th4)*cos(th6)*(cos(th1)*cos(th3) -
cos(th2)*sin(th1)*sin(th3)) + cos(th5)*sin(th4)*sin(th6)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) + L3*cos(th4)*(cos(th1)*cos(th3) -
cos(th2)*sin(th1)*sin(th3))) - th2dot*(L4*(sin(th6)*(cos(th3)*sin(th1)*sin(th2)*sin(th5) + cos(th4)*cos(th5)*sin(th1)*sin(th2)*sin(th3)) -
cos(th6)*sin(th1)*sin(th2)*sin(th3)*sin(th4)) - L3*sin(th1)*sin(th2)*sin(th3)*sin(th4)) - th1dot*(L4*(sin(th6)*(sin(th5)*(sin(th1)*sin(th3) -
cos(th1)*cos(th2)*cos(th3)) - cos(th4)*cos(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) + cos(th6)*sin(th4)*(cos(th3)*sin(th1) +
cos(th1)*cos(th2)*sin(th3))) + L3*sin(th4)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) +
th3dot*(L4*(sin(th6)*(sin(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3)) + cos(th4)*cos(th5)*(cos(th1)*sin(th3) +
cos(th2)*cos(th3)*sin(th1))) - cos(th6)*sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1))) - L3*sin(th4)*(cos(th1)*sin(th3) +
cos(th2)*cos(th3)*sin(th1))) + L4*th6dot*(cos(th6)*(sin(th5)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) -
cos(th4)*cos(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) - sin(th4)*sin(th6)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) +
L4*th5dot*sin(th6)*(cos(th5)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + cos(th4)*sin(th5)*(cos(th1)*cos(th3) -
cos(th2)*sin(th1)*sin(th3))), th2dot*(L4*(cos(th6)*(cos(th2)*sin(th1)*sin(th4) - cos(th3)*cos(th4)*sin(th1)*sin(th2)) -
cos(th5)*sin(th6)*(cos(th2)*cos(th4)*sin(th1) + cos(th3)*sin(th1)*sin(th2)*sin(th4))) + L3*(cos(th2)*sin(th1)*sin(th4) -
cos(th3)*cos(th4)*sin(th1)*sin(th2))) + th3dot*(L4*(cos(th4)*cos(th6)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3)) +
cos(th5)*sin(th4)*sin(th6)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) + L3*cos(th4)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) -
th1dot*(L4*(cos(th6)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) +
cos(th5)*sin(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2))) + L3*(cos(th4)*(sin(th1)*sin(th3)
- cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4))) - th4dot*(L4*(cos(th6)*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1))
- cos(th4)*sin(th1)*sin(th2)) - cos(th5)*sin(th6)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4))) +
L3*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2))) -
L4*th6dot*(sin(th6)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) -
cos(th5)*cos(th6)*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2))) -
```

```
L4*th5dot*sin(th5)*sin(th6)*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)),
L4*th6dot*cos(th6)*(sin(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) -
cos(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) - L4*th1dot*sin(th6)*(sin(th5)*(cos(th4)*(sin(th1)*sin(th3) -
cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) - cos(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) +
L4*th5dot*sin(th6)*(cos(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) +
sin(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) + L4*th3dot*sin(th6)*(cos(th5)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) +
cos(th4)*sin(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) + L4*th2dot*sin(th6)*(sin(th5)*(cos(th2)*sin(th1)*sin(th4) -
cos(th3)*cos(th4)*sin(th1)*sin(th2)) - cos(th5)*sin(th1)*sin(th2)*sin(th3)) - L4*th4dot*sin(th5)*sin(th6)*(sin(th4)*(cos(th1)*sin(th3) +
cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)), L4*th1dot*(sin(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) +
cos(th1)*cos(th4)*sin(th2)) + cos(th6)*(cos(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) +
sin(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3)))) - L4*th2dot*(cos(th6)*(cos(th5)*(cos(th2)*sin(th1)*sin(th4) -
cos(th3)*cos(th4)*sin(th1)*sin(th2)) + sin(th1)*sin(th2)*sin(th3)*sin(th5)) - sin(th6)*(cos(th2)*cos(th4)*sin(th1) +
cos(th3)*sin(th1)*sin(th2)*sin(th4))) - L4*th4dot*(sin(th6)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) +
sin(th1)*sin(th2)*sin(th4)) - cos(th5)*cos(th6)*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2))) -
L4*th6dot*(cos(th6)*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)) -
sin(th6)*(cos(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th1)*cos(th3) -
cos(th2)*sin(th1)*sin(th3)))) + L4*th3dot*(cos(th6)*(sin(th5)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) -
cos(th4)*cos(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) - sin(th4)*sin(th6)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) +
L4*th5dot*cos(th6)*(sin(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) -
cos(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))),
0;

0,
th4dot*(L3*(sin(th2)*sin(th4) + cos(th2)*cos(th3)*cos(th4)) + L4*(cos(th6)*(sin(th2)*sin(th4) + cos(th2)*cos(th3)*cos(th4)) -
cos(th5)*sin(th6)*(cos(th4)*sin(th2) - cos(th2)*cos(th3)*sin(th4)))) + th3dot*(L4*(sin(th6)*(cos(th2)*cos(th3)*sin(th5) +
cos(th2)*cos(th4)*cos(th5)*sin(th3)) - cos(th2)*cos(th6)*sin(th3)*sin(th4)) - L3*cos(th2)*sin(th3)*sin(th4)) - th2dot*(L3*(cos(th2)*cos(th4)
+ cos(th3)*sin(th2)*sin(th4)) + L4*(sin(th6)*(cos(th5)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) + sin(th2)*sin(th3)*sin(th5)) +
cos(th6)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4))) + L2*cos(th2)) - L4*th6dot*(cos(th6)*(cos(th5)*(sin(th2)*sin(th4) +
cos(th2)*cos(th3)*cos(th4)) - cos(th2)*sin(th3)*sin(th5)) - sin(th6)*(cos(th4)*sin(th2) - cos(th2)*cos(th3)*sin(th4))) +
L4*th5dot*sin(th6)*(sin(th5)*(sin(th2)*sin(th4) + cos(th2)*cos(th3)*cos(th4)) + cos(th2)*cos(th5)*sin(th3)),
th2dot*(L4*(sin(th6)*(cos(th2)*cos(th3)*sin(th5) + cos(th2)*cos(th4)*cos(th5)*sin(th3)) - cos(th2)*cos(th6)*sin(th3)*sin(th4)) -
L3*cos(th2)*sin(th3)*sin(th4)) - th4dot*(L4*(cos(th4)*cos(th6)*sin(th2)*sin(th3) + cos(th5)*sin(th2)*sin(th3)*sin(th4)*sin(th6)) +
L3*cos(th4)*sin(th2)*sin(th3)) - th3dot*(L4*(sin(th6)*(sin(th2)*sin(th3)*sin(th5) - cos(th3)*cos(th4)*cos(th5)*sin(th2)) +
cos(th3)*cos(th6)*sin(th2)*sin(th4)) + L3*cos(th3)*sin(th2)*sin(th4)) + L4*th6dot*(cos(th6)*(cos(th3)*sin(th2)*sin(th5) +
cos(th4)*cos(th5)*sin(th2)*sin(th3)) + sin(th2)*sin(th3)*sin(th4)*sin(th6)) + L4*th5dot*sin(th6)*(cos(th3)*cos(th5)*sin(th2) -
cos(th4)*sin(th2)*sin(th3)*sin(th5)),
th2dot*(L3*(sin(th2)*sin(th4) + cos(th2)*cos(th3)*cos(th4)) + L4*(cos(th6)*(sin(th2)*sin(th4) + cos(th2)*cos(th3)*cos(th4)) -
cos(th5)*sin(th6)*(cos(th4)*sin(th2) - cos(th2)*cos(th3)*sin(th4)))) - th3dot*(L4*(cos(th4)*cos(th6)*sin(th2)*sin(th3) +
cos(th5)*sin(th2)*sin(th3)*sin(th4)*sin(th6)) + L3*cos(th4)*sin(th2)*sin(th3)) - th4dot*(L3*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4))
+ L4*(cos(th6)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4)) + cos(th5)*sin(th6)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)))) +
L4*th6dot*(sin(th6)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) + cos(th5)*cos(th6)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4))) -
L4*th5dot*sin(th5)*sin(th6)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4)),
L4*th2dot*sin(th6)*(sin(th5)*(sin(th2)*sin(th4) + cos(th2)*cos(th3)*cos(th4)) + cos(th2)*cos(th5)*sin(th3)) -
L4*th6dot*cos(th6)*(sin(th5)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) - cos(th5)*sin(th2)*sin(th3)) -
L4*th5dot*sin(th6)*(cos(th5)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) + sin(th2)*sin(th3)*sin(th5)) +
L4*th3dot*sin(th6)*(cos(th3)*cos(th5)*sin(th2) - cos(th4)*sin(th2)*sin(th3)*sin(th5)) - L4*th4dot*sin(th5)*sin(th6)*(cos(th2)*cos(th4) +
cos(th3)*sin(th2)*sin(th4)),
L4*th4dot*(sin(th6)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) + cos(th5)*cos(th6)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4))) -
L4*th6dot*(sin(th6)*(cos(th5)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) + sin(th2)*sin(th3)*sin(th5)) + cos(th6)*(cos(th2)*cos(th4) +
cos(th3)*sin(th2)*sin(th4))) - L4*th2dot*(cos(th6)*(cos(th5)*(sin(th2)*sin(th4) + cos(th2)*cos(th3)*cos(th4)) - cos(th2)*sin(th3)*sin(th5)) -
sin(th6)*(cos(th4)*sin(th2) - cos(th2)*cos(th3)*sin(th4))) + L4*th3dot*(cos(th6)*(cos(th3)*sin(th2)*sin(th5) +
cos(th4)*cos(th5)*sin(th2)*sin(th3)) + sin(th2)*sin(th3)*sin(th4)*sin(th6)) - L4*th5dot*cos(th6)*(sin(th5)*(cos(th2)*sin(th4) -
cos(th3)*cos(th4)*sin(th2)) - cos(th5)*sin(th2)*sin(th3)),
0;

0,
th1dot*cos(th1),
th1dot*sin(th1)*sin(th2) - th2dot*cos(th1)*cos(th2),
th3dot*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - th1dot*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3)) +
th2dot*cos(th1)*sin(th2)*sin(th3),
- th4dot*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) - th1dot*(sin(th4)*(cos(th1)*sin(th3) +
cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)) - th2dot*(cos(th1)*cos(th2)*cos(th4) + cos(th1)*cos(th3)*sin(th2)*sin(th4)) -
th3dot*sin(th4)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3)),
th2dot*(sin(th5)*(cos(th1)*cos(th2)*sin(th4) - cos(th1)*cos(th3)*cos(th4)*sin(th2)) - cos(th1)*cos(th5)*sin(th2)*sin(th3)) -
th1dot*(sin(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) - cos(th5)*(cos(th1)*cos(th3) -
cos(th2)*sin(th1)*sin(th3))) - th3dot*(cos(th5)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th4)*sin(th5)*(cos(th3)*sin(th1) +
cos(th1)*cos(th2)*sin(th3))) - th5dot*(cos(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) +
sin(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) + th4dot*sin(th5)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) +
cos(th1)*cos(th4)*sin(th2)), th6dot*(sin(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2)) +
cos(th6)*(cos(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th3)*sin(th1) +
```

```
cos(th1)*cos(th2)*sin(th3)))) - th1dot*(cos(th6)*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)) -
sin(th6)*(cos(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th1)*cos(th3) -
cos(th2)*sin(th1)*sin(th3)))) - th2dot*(cos(th6)*(cos(th1)*cos(th2)*cos(th4) + cos(th1)*cos(th3)*sin(th2)*sin(th4)) +
sin(th6)*(cos(th5)*(cos(th1)*cos(th2)*sin(th4) - cos(th1)*cos(th3)*cos(th4)*sin(th2)) + cos(th1)*sin(th2)*sin(th3)*sin(th5))) -
th3dot*(sin(th6)*(sin(th5)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th4)*cos(th5)*(cos(th3)*sin(th1) +
cos(th1)*cos(th2)*sin(th3))) + cos(th6)*sin(th4)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) -
th4dot*(cos(th6)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) +
cos(th5)*sin(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2))) -
th5dot*sin(th6)*(sin(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) -
cos(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3)));

0,
th1dot*sin(th1),
- th1dot*cos(th1)*sin(th2) - th2dot*cos(th2)*sin(th1),
th2dot*sin(th1)*sin(th2)*sin(th3) - th3dot*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - th1dot*(cos(th3)*sin(th1) +
cos(th1)*cos(th2)*sin(th3)),
th4dot*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) - th1dot*(sin(th4)*(sin(th1)*sin(th3) -
cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2)) - th2dot*(cos(th2)*cos(th4)*sin(th1) + cos(th3)*sin(th1)*sin(th2)*sin(th4)) +
th3dot*sin(th4)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3)),
th5dot*(cos(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th1)*cos(th3) -
cos(th2)*sin(th1)*sin(th3))) - th1dot*(sin(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) -
cos(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) + th3dot*(cos(th5)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) +
cos(th4)*sin(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) + th2dot*(sin(th5)*(cos(th2)*sin(th1)*sin(th4) -
cos(th3)*cos(th4)*sin(th1)*sin(th2)) - cos(th5)*sin(th1)*sin(th2)*sin(th3)) - th4dot*sin(th5)*(sin(th4)*(cos(th1)*sin(th3) +
cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)), th3dot*(sin(th6)*(sin(th5)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) -
cos(th4)*cos(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) + cos(th6)*sin(th4)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) -
th6dot*(sin(th6)*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)) +
cos(th6)*(cos(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th1)*cos(th3) -
cos(th2)*sin(th1)*sin(th3)))) - th1dot*(cos(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2)) -
sin(th6)*(cos(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th3)*sin(th1) +
cos(th1)*cos(th2)*sin(th3)))) + th4dot*(cos(th6)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) +
cos(th5)*sin(th6)*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2))) -
th2dot*(sin(th6)*(cos(th5)*(cos(th2)*sin(th1)*sin(th4) - cos(th3)*cos(th4)*sin(th1)*sin(th2)) + sin(th1)*sin(th2)*sin(th3)*sin(th5)) +
cos(th6)*(cos(th2)*cos(th4)*sin(th1) + cos(th3)*sin(th1)*sin(th2)*sin(th4))) + th5dot*sin(th6)*(sin(th5)*(cos(th4)*(cos(th1)*sin(th3) +
cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) - cos(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3)));

0,
0,
-th2dot*sin(th2),
- th2dot*cos(th2)*sin(th3) - th3dot*cos(th3)*sin(th2),
- th2dot*(cos(th4)*sin(th2) - cos(th2)*cos(th3)*sin(th4)) - th4dot*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) -
th3dot*sin(th2)*sin(th3)*sin(th4),
th3dot*(cos(th3)*cos(th5)*sin(th2) - cos(th4)*sin(th2)*sin(th3)*sin(th5)) + th2dot*(sin(th5)*(sin(th2)*sin(th4) + cos(th2)*cos(th3)*cos(th4))
+ cos(th2)*cos(th5)*sin(th3)) - th5dot*(cos(th5)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) + sin(th2)*sin(th3)*sin(th5)) -
th4dot*sin(th5)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4)),
th6dot*(cos(th6)*(cos(th5)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) + sin(th2)*sin(th3)*sin(th5)) - sin(th6)*(cos(th2)*cos(th4) +
cos(th3)*sin(th2)*sin(th4))) - th2dot*(sin(th6)*(cos(th5)*(sin(th2)*sin(th4) + cos(th2)*cos(th3)*cos(th4)) - cos(th2)*sin(th3)*sin(th5)) +
cos(th6)*(cos(th4)*sin(th2) - cos(th2)*cos(th3)*sin(th4))) - th4dot*(cos(th6)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) -
cos(th5)*sin(th6)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4))) + th3dot*(sin(th6)*(cos(th3)*sin(th2)*sin(th5) +
cos(th4)*cos(th5)*sin(th2)*sin(th3)) - cos(th6)*sin(th2)*sin(th3)*sin(th4)) - th5dot*sin(th6)*(sin(th5)*(cos(th2)*sin(th4) -
cos(th3)*cos(th4)*sin(th2)) - cos(th5)*sin(th2)*sin(th3))];
```

```
function [px,py,pz,n_r,o_r,a_r] = ff_fun(v)

th1 = v(1);
th2 = v(2);
th3 = v(3);
th4 = v(4);
th5 = v(5);
th6 = v(6);
th7 = v(7);

L1 = 0.34;
L2 = 0.4;
L3 = 0.4;
L4 = 0.126;
```

```
T=[ sin(th7)*(sin(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) - cos(th5)*(cos(th3)*sin(th1) +
cos(th1)*cos(th2)*sin(th3))) - cos(th7)*(sin(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2)) +
cos(th6)*(cos(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th3)*sin(th1) +
cos(th1)*cos(th2)*sin(th3)))), sin(th6)*(cos(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) +
sin(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) - cos(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) +
cos(th1)*cos(th4)*sin(th2)), - cos(th7)*(sin(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) -
cos(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) - sin(th7)*(sin(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) +
cos(th1)*cos(th4)*sin(th2)) + cos(th6)*(cos(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) +
sin(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3)))), - L4*(cos(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) +
cos(th1)*cos(th4)*sin(th2)) - sin(th6)*(cos(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) +
sin(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3)))) - L3*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) +
cos(th1)*cos(th4)*sin(th2)) - L2*cos(th1)*sin(th2);
 cos(th7)*(sin(th6)*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)) +
cos(th6)*(cos(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th1)*cos(th3) -
cos(th2)*sin(th1)*sin(th3)))) - sin(th7)*(sin(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) -
cos(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))), cos(th6)*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) -
cos(th4)*sin(th1)*sin(th2)) - sin(th6)*(cos(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) +
sin(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))),   cos(th7)*(sin(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) +
sin(th1)*sin(th2)*sin(th4)) - cos(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) + sin(th7)*(sin(th6)*(sin(th4)*(cos(th1)*sin(th3) +
cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)) + cos(th6)*(cos(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) +
sin(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3)))),   L4*(cos(th6)*(sin(th4)*(cos(th1)*sin(th3) +
cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)) - sin(th6)*(cos(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) +
sin(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3)))) + L3*(sin(th4)*(cos(th1)*sin(th3) +
cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)) - L2*sin(th1)*sin(th2);

sin(th7)*(sin(th5)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) - cos(th5)*sin(th2)*sin(th3)) -
cos(th7)*(cos(th6)*(cos(th5)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) + sin(th2)*sin(th3)*sin(th5)) - sin(th6)*(cos(th2)*cos(th4) +
cos(th3)*sin(th2)*sin(th4))),                                                                                    sin(th6)*(cos(th5)*(cos(th2)*sin(th4) -
cos(th3)*cos(th4)*sin(th2)) + sin(th2)*sin(th3)*sin(th5)) + cos(th6)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4)),
- cos(th7)*(sin(th5)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) - cos(th5)*sin(th2)*sin(th3)) -
sin(th7)*(cos(th6)*(cos(th5)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) + sin(th2)*sin(th3)*sin(th5)) - sin(th6)*(cos(th2)*cos(th4) +
cos(th3)*sin(th2)*sin(th4))),                                                                                                                  L1 +
L3*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4)) + L4*(sin(th6)*(cos(th5)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) +
sin(th2)*sin(th3)*sin(th5)) + cos(th6)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4))) + L2*cos(th2)];
nx = T(1,1);
ny = T(2,1);
nz = T(3,1);

ox = T(1,2);
oy = T(2,2);
oz = T(3,2);

ax = T(1,3);
ay = T(2,3);
az = T(3,3);

px = T(1,4);
py = T(2,4);
pz = T(3,4);

n_r = [nx,ny,nz];
o_r = [ox;oy;oz];
a_r = [ax;ay;az];
```

```
function y = fcost4(v,px,py,pz,n_t,o_t,a_t)

th1 = v(1);
th2 = v(2);
th3 = v(3);
th4 = v(4);
th5 = v(5);
th6 = v(6);
th7 = v(7);

L1 = 0.34;
L2 = 0.4;
```

```
L3 = 0.4;
L4 = 0.126;


T_07 = [sin(th7)*(sin(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) -
cos(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) - cos(th7)*(sin(th6)*(sin(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) +
cos(th1)*cos(th4)*sin(th2)) + cos(th6)*(cos(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) - cos(th1)*sin(th2)*sin(th4)) +
sin(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3)))), sin(th6)*(cos(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) -
cos(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) - cos(th6)*(sin(th4)*(sin(th1)*sin(th3) -
cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2)), - cos(th7)*(sin(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) -
cos(th1)*sin(th2)*sin(th4)) - cos(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3))) - sin(th7)*(sin(th6)*(sin(th4)*(sin(th1)*sin(th3) -
cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2)) + cos(th6)*(cos(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) -
cos(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3)))), - L4*(cos(th6)*(sin(th4)*(sin(th1)*sin(th3) -
cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2)) - sin(th6)*(cos(th5)*(cos(th4)*(sin(th1)*sin(th3) - cos(th1)*cos(th2)*cos(th3)) -
cos(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th3)*sin(th1) + cos(th1)*cos(th2)*sin(th3)))) - L3*(sin(th4)*(sin(th1)*sin(th3) -
cos(th1)*cos(th2)*cos(th3)) + cos(th1)*cos(th4)*sin(th2)) - L2*cos(th1)*sin(th2);
 cos(th7)*(sin(th6)*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)) +
cos(th6)*(cos(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th1)*cos(th3) -
cos(th2)*sin(th1)*sin(th3)))) - sin(th7)*(sin(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) -
cos(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))), cos(th6)*(sin(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) -
cos(th4)*sin(th1)*sin(th2)) - sin(th6)*(cos(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) + sin(th1)*sin(th2)*sin(th4)) +
sin(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))),   cos(th7)*(sin(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) +
sin(th1)*sin(th2)*sin(th4)) - cos(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3))) + sin(th7)*(sin(th6)*(sin(th4)*(cos(th1)*sin(th3) +
cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)) + cos(th6)*(cos(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) +
sin(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3)))),   L4*(cos(th6)*(sin(th4)*(cos(th1)*sin(th3) +
cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)) - sin(th6)*(cos(th5)*(cos(th4)*(cos(th1)*sin(th3) + cos(th2)*cos(th3)*sin(th1)) +
sin(th1)*sin(th2)*sin(th4)) + sin(th5)*(cos(th1)*cos(th3) - cos(th2)*sin(th1)*sin(th3)))) + L3*(sin(th4)*(cos(th1)*sin(th3) +
cos(th2)*cos(th3)*sin(th1)) - cos(th4)*sin(th1)*sin(th2)) - L2*sin(th1)*sin(th2);

sin(th7)*(sin(th5)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) - cos(th5)*sin(th2)*sin(th3)) -
cos(th7)*(cos(th6)*(cos(th5)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) + sin(th2)*sin(th3)*sin(th5)) - sin(th6)*(cos(th2)*cos(th4) +
cos(th3)*sin(th2)*sin(th4))),                                                                                    sin(th6)*(cos(th5)*(cos(th2)*sin(th4) -
cos(th3)*cos(th4)*sin(th2)) + sin(th2)*sin(th3)*sin(th5)) + cos(th6)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4)),
- cos(th7)*(sin(th5)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) - cos(th5)*sin(th2)*sin(th3)) -
sin(th7)*(cos(th6)*(cos(th5)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) + sin(th2)*sin(th3)*sin(th5)) - sin(th6)*(cos(th2)*cos(th4) +
cos(th3)*sin(th2)*sin(th4))),                                                                                                                                  L1 +
L3*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4)) + L4*(sin(th6)*(cos(th5)*(cos(th2)*sin(th4) - cos(th3)*cos(th4)*sin(th2)) +
sin(th2)*sin(th3)*sin(th5)) + cos(th6)*(cos(th2)*cos(th4) + cos(th3)*sin(th2)*sin(th4))) + L2*cos(th2);

0,
0,
0,
1];
 nx = n_t(1);
 ny = n_t(2);
 nz = n_t(3);

  ox = o_t(1);
 oy = o_t(2);
 oz = o_t(3);

  ax = a_t(1);
 ay = a_t(2);
 az = a_t(3);

% desired position
y(1) = T_07(1,4)-px;
y(2) = T_07(2,4)-py;
y(3) = T_07(3,4)-pz;

% desired orientation
y(4) = T_07(1,1)-nx;
y(5) = T_07(2,2)-oy;
y(6) = T_07(3,3)-az;
y(7) = T_07(2,1)-ny;
y(8) = T_07(3,1)-nz;
y(9) = T_07(1,2)-ox;
y(10) = T_07(3,2)-oz;
y(11) = T_07(1,3)-ax;
y(12) = T_07(2,3)-ay;
```

```
%% Kinematic Simulation of iiwa 7 R800

% load('Theta_star.mat')
% load('feed_results_net2.mat')
%% Loading DH table
DH_table
%% Defining Links
for ilink =1:7
   L(ilink) = Link(DH_iiwa7(ilink,:));
end

Rob =  SerialLink(L); % creating the robot

Rob.name = 'iiwa 7 R800'; % to add name

%% Animation

k=0;
axis tight manual
ax = gca;
ax.NextPlot = 'replaceChildren';


v = VideoWriter('sim_robot.avi');
open(v);
for j=1:10:length(time)
       k=k+1;
   joints_angles = Theta(:,j);
   Rob.plot(joints_angles')
   hold on
   axp = plot3(pxd(j),pyd(j),pzd(j),'xr')
   axis([-1 1 -1 1 -1 1])
   pause(0.25)
   videoM(k) = getframe(gcf);
   writeVideo(v,videoM(k));
end
```

```
%% symbolic program for equations of motion
clear; clc; close all

syms g m1 m2 m3 m4 m5 m6 m7
syms Ixx1 Iyy1 Izz1 Ixy1 Iyz1 Izx1
syms Ixx2 Iyy2 Izz2 Ixy2 Iyz2 Izx2
syms Ixx3 Iyy3 Izz3 Ixy3 Iyz3 Izx3
syms Ixx4 Iyy4 Izz4 Ixy4 Iyz4 Izx4
syms Ixx5 Iyy5 Izz5 Ixy5 Iyz5 Izx5
syms Ixx6 Iyy6 Izz6 Ixy6 Iyz6 Izx6
syms Ixx7 Iyy7 Izz7 Ixy7 Iyz7 Izx7



syms th1 th2 th3 th4 th5 th6 th7
syms th1dot th2dot th3dot th4dot th5dot th6dot th7dot
syms L1 L2 L3 L4
%% DH matrices
Forward_solution
% Inertia_parameters
% T_2c1=[eye(3),[rx(1);ry(1);rz(1)];[0,0,0,1]];
% Tnc(:,:,1)=T_02*T_2c1;
center_mass_tranformation
Tnc(:,:,1) =T_01*T_11c;
Tnc(:,:,2) = T_02*T_22c;
Tnc(:,:,3) = T_03*T_33c;
Tnc(:,:,4) = T_04*T_44c;
Tnc(:,:,5) = T_05*T_55c;
Tnc(:,:,6)= T_06*T_66c;
```

```
Tnc(:,:,7)= T_07*T_77c;

%% change this always
g = [0;0;g]; %the gravitational vector in the ditrection of z0 in this example.
q = [th1; th2;th3;th4;th5;th6;th7];
ru = ones(1,7);
qdot = [th1dot;th2dot;th3dot;th4dot;th5dot;th6dot;th7dot];
dof = 7; % Number of DOF
mass = [m1;m2;m3;m4;m5;m6;m7];

% for iI = 1:dof
% I(1,1,iI) = Ixx(iI);
% I(2,2,iI) = Iyy(iI);
% I(3,4,iI) = Izz(iI);
%
% I(1,2,iI) = Ixy(iI);
% I(2,1,iI) = Ixy(iI);
%
% I(2,3,iI) = Iyz(iI);
% I(3,2,iI) = Iyz(iI);
%
% I(1,3,iI) = Izx(iI);
% I(3,1,iI) = Izx(iI);
% end
 D = [Ixx1 Iyy1 Izz1 Ixy1 Iyz1 Izx1;
 Ixx2 Iyy2 Izz2 Ixy2 Iyz2 Izx2;
Ixx3 Iyy3 Izz3 Ixy3 Iyz3 Izx3;
 Ixx4 Iyy4 Izz4 Ixy4 Iyz4 Izx4;
 Ixx5 Iyy5 Izz5 Ixy5 Iyz5 Izx5;
 Ixx6 Iyy6 Izz6 Ixy6 Iyz6 Izx6;
 Ixx7 Iyy7 Izz7 Ixy7 Iyz7 Izx7];

 Ixx = D(:,1);
 Iyy = D(:,2);
 Izz = D(:,3);
 Ixy = D(:,4);
 Iyz = D(:,5);
 Izx = D(:,6);
for iI = 1:dof
I(1,1,iI) = Ixx(iI);
I(2,2,iI) = Iyy(iI);
I(3,3,iI) = Izz(iI);

I(1,2,iI) = Ixy(iI);
I(2,1,iI) = Ixy(iI);

I(2,3,iI) = Iyz(iI);
I(3,2,iI) = Iyz(iI);

I(1,3,iI) = Izx(iI);
I(3,1,iI) = Izx(iI);
end


%% angular velocity vectors
z0 = [0; 0; 1];
ru0 = ru(1);
w01 = ru0*z0*qdot(1:1);
wii = transpose(Tn(1:3,1:3,1))*w01;
ruz =[ru0*z0];

for i=2:dof
z = Tnc(1:3,3,i);
rui = ru(i);
ruz = [ruz,rui*z];
w0i = ruz*qdot(1:i);
wii(:,i) = transpose(Tn(1:3,1:3,i))*w0i;
end
```

```
%% linear velocity vectors
rdot = sym(zeros(3,dof));
for idof=1:dof
   for i =1:idof
      rdot(:,idof) = rdot(:,idof) + diff(Tnc(1:3,4,idof),q(i))*qdot(i);
   end
end

%% potential energy due to gravity
P=sym(0);
for idof =1:dof
   P = P+ mass(idof)*transpose(g)*Tnc(1:3,4,idof);
end

%% kinetic energy

K=sym(0);
for idof =1:dof
   K = K +1/2* mass(idof)*transpose(rdot(:,idof))*rdot(:,idof) + 1/2*(transpose(wii(:,idof))*I(:,:,idof)*wii(:,idof));
end
%% Equations of motion
 phi = sym(zeros(dof,1));
 c = sym(zeros(dof,1));
 M = sym(zeros(dof));
for i=1:dof
   phi(i,1) = diff(P,q(i));
   c(i,1) = -diff(K,q(i));
   dK_dqdot = diff(K,qdot(i));
   for j=1:dof
      M(i,j) = diff(dK_dqdot,qdot(j));
      c(i,1) = c(i,1) + diff(dK_dqdot,q(j))*qdot(j);
   end
end
% M = simplify(M);
% c = simplify(c);
% phi = simplify(phi);
M2 = char(M);
fileID = fopen('M_mat.txt','w');
fprintf(fileID,'%s',M2);
fclose(fileID);


c2 = char(c);
fileID = fopen('c_mat.txt','w');
fprintf(fileID,'%s',c2);
fclose(fileID);

phi2 = char(phi);
fileID = fopen('phi_mat.txt','w');
fprintf(fileID,'%s',phi2);
fclose(fileID);
```

```
%%% HW7 PD control%%%
clear; clc; close all
%Initial conditions
tic
%
global fname ifig save_flag
folder_name = 'Figures_point6'
fname1 = 'C:\Users\Turki\Dropbox (VCADS)\1SCH\Spring 2018\Robatics\Project term\iiwa 7\Matlab code\Sliding Control\iiwa real
trajectory with poly coef - Copy\';
fname = strcat(fname1,folder_name);
ifig = 0;
save_flag = 1;

% integration
clear global u ode_indx eta eps lambda ts tt
global u ode_indx eta eps lambda  tt iper tf
iper =1;
```

```
ode_indx = 0;
eta = 10;
eps = 2;
lambda = 1.5 ;



% t0 = 0;
% tfin = 5; % total simulation time
% tspan = [t0 tfin]; %time span

dof = 7;
load('desired_trajectory_joints3.mat')
tf = time(end);
x0(1:dof) = Theta(:,1);
x0(dof+1:2*dof) = Theta_dot(:,1);


[t, x] = ode45('fun_ode2',time,x0);


toc
plot_results
plot_input
Forward_solution
```

```
% HW6 derivative equations
% x = [theta1; d2; theta1_dot; d2_dot];
% xder =x_dot =[theta1_dot; d2_dot; theta1_ddot; d2_ddot];
function xder = fun_ode2(t,x)
global u ode_indx tt iper tf
ode_indx = ode_indx +1;
tt(ode_indx) = t;
dof  = 7;
ns = dof*2;



th1 = x(1);
th2 = x(2);
th3 = x(3);
th4 = x(4);
th5 = x(5);
th6 = x(6);
th7 = x(7);

th1dot = x(1+7);
th2dot = x(2+7);
th3dot = x(3+7);
th4dot = x(4+7);
th5dot = x(5+7);
th6dot = x(6+7);
th7dot = x(7+7);

%define M c phi
iiwa_mat;
% iiwa_mat2;

%desired_trajectory
% desired_case = 2;
% desired_pathes;
% load('desired_trajectory_joints.mat');
% load('poly_coef3.mat')
% for i=1:7
%     q_des(i,1) =polyval(Q_theta(i,:),t) ;
%     qdot_des(i,1) =polyval(Q_theta_dot(i,:),t) ;
%     qddot_des(i,1) =polyval(Q_theta_ddot(i,:),t) ;
% end
load('desired_trajectory_joints3.mat')
q_des(1,1) = interp1(time,Theta(1,:),t);
```

```
q_des(2,1) = interp1(time,Theta(2,:),t);
q_des(3,1) = interp1(time,Theta(3,:),t);
q_des(4,1) = interp1(time,Theta(4,:),t);
q_des(5,1) = interp1(time,Theta(5,:),t);
q_des(6,1) = interp1(time,Theta(6,:),t);
q_des(7,1) = interp1(time,Theta(7,:),t);

qdot_des(1,1) = interp1(time,Theta_dot(1,:),t);
qdot_des(2,1) = interp1(time,Theta_dot(2,:),t);
qdot_des(3,1) = interp1(time,Theta_dot(3,:),t);
qdot_des(4,1) = interp1(time,Theta_dot(4,:),t);
qdot_des(5,1) = interp1(time,Theta_dot(5,:),t);
qdot_des(6,1) = interp1(time,Theta_dot(6,:),t);
qdot_des(7,1) = interp1(time,Theta_dot(7,:),t);

qddot_des(1,1) = interp1(time,Theta_ddot(1,:),t);
qddot_des(2,1) = interp1(time,Theta_ddot(2,:),t);
qddot_des(3,1) = interp1(time,Theta_ddot(3,:),t);
qddot_des(4,1) = interp1(time,Theta_ddot(4,:),t);
qddot_des(5,1) = interp1(time,Theta_ddot(5,:),t);
qddot_des(6,1) = interp1(time,Theta_ddot(6,:),t);
qddot_des(7,1) = interp1(time,Theta_ddot(7,:),t);


% define controller
% sliding_control;
sliding_control2;


% Actuator forces/torques
xdd = M\(-c-phi-B.*x(dof+1:ns)+u(:,ode_indx)); %accelerations
xder = [x(dof+1:ns); xdd]; %state time derivatives

perc = t/tf*100;
perc_d = [10:10:100];
if round(perc,0) ==perc_d(iper)
    perc_Str = strcat(num2str(perc_d(iper)),'%');
    display(perc_Str)
    if iper<length(perc_d)
    iper = iper+1;
    else
        iper = length(perc_d);
    end
end
clear M c phi
%sliding mode control
%assume the only uncertainity is froction coefficient 0<=B<=2*B
global eta eps lambda

F = 2*B.*abs(x(dof+1:ns));
k = F + eta; %robsustness gain
s = x(dof+1:ns) - qdot_des + lambda*(x(1:dof)-q_des); %Surfaces
sr = qddot_des - lambda*(x(dof+1:ns)-qdot_des); %See notes
sw = k.*sign(s);

for i=1:dof
   if abs(s(i)) <= eps
     %sw(i) = k(i)*tanh(s(i)/eps); %hyper-tangent
      sw(i) = k(i)*s(i)/eps; %saturation
   end
end

global u
u(:,ode_indx) = c + phi + B.*x(dof+1:ns) + M*(sr - sw);
```