Answer Sheet

Index No: 210204R

1. What is the geometrical structure of the Puma 560 robot?

   RRRRRR (6 dof serial manupilator)

2. Fill the DH parameter table for Puma 560 robot.

j	θ	d	a	α
1	q_1	0	0	1.5708
2	q_2	0	0.4318	0
3	q_3	0.15005	0.0203	−1.5708
4	q_4	0.4318	0	1.5708
5	q_5	0	0	−1.5708
6	q_6	0	0	0

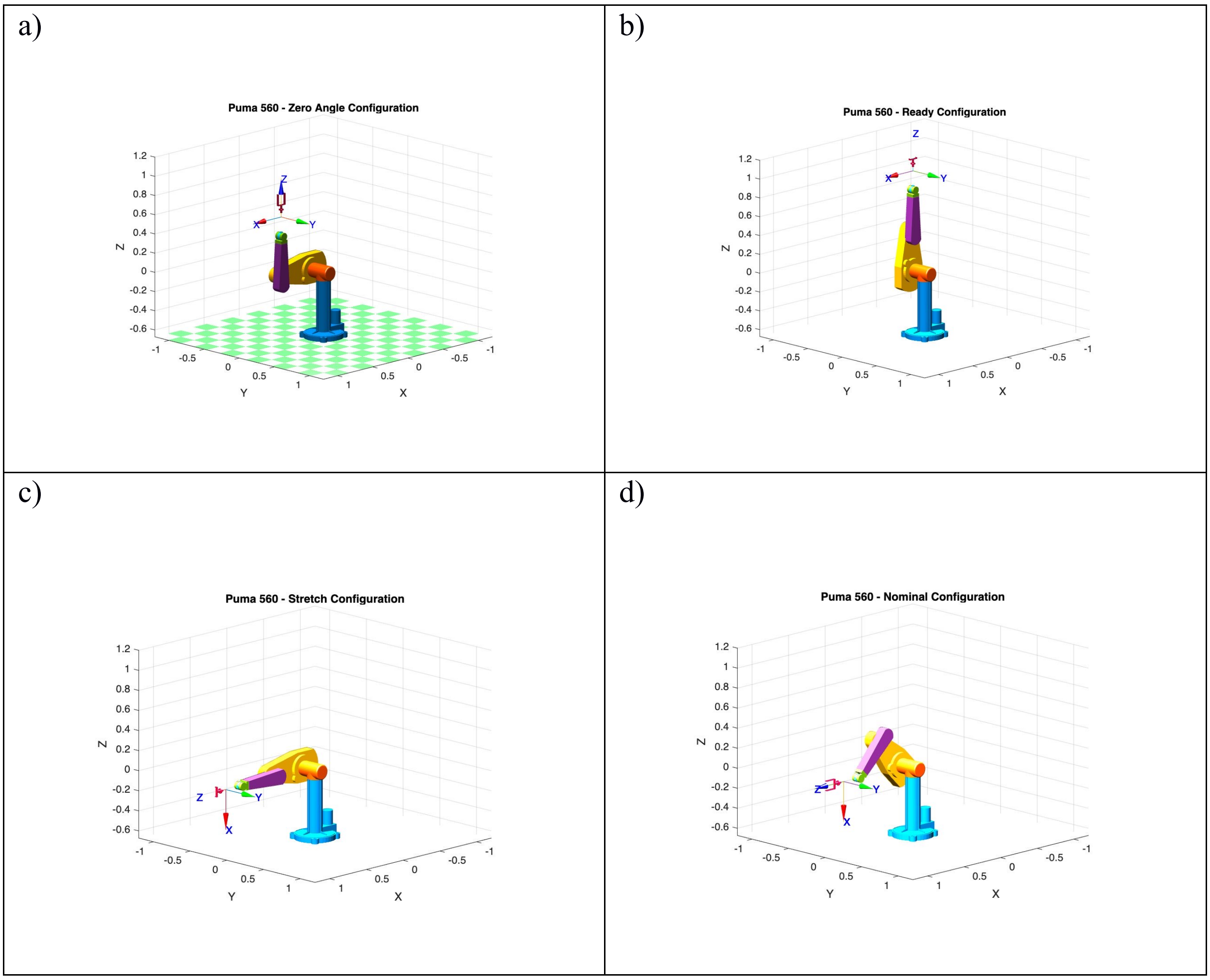
3. Joint coordinate vectors of Puma 560 robot for the following canonical configurations:
   a) zero angle: [0 0 0 0 0 0]
   b) ready : [0 1.5708 −1.5708 0 0 0]
   c) stretch : [0 0 −1.5708 0 0 0]
   d) nominal : [0 0.7854 3.1416 0 0.7854 0]

4. Forward kinematics for tool center point (TCP) in Procedure 3.5 for the canonical configurations.

Configuration	**Position**	**Orientation (Rotation Matrix)**
Zero angle	$\begin{bmatrix} 0.4521 \\ -0.15 \\ 0.6318 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$
Ready	$\begin{bmatrix} 0.0203 \\ -0.15 \\ 1.064 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$
Stretch	$\begin{bmatrix} 1.064 \\ -0.1501 \\ -0.0203 \end{bmatrix}$	$\begin{bmatrix} 6.123e-17 & 6.123e-17 & 1 \\ -6.123e-17 & & 1-6.123e-17 \\ & -1-6.123e-17 & 6.123e-17 \end{bmatrix}$
Nominal	$\begin{bmatrix} 0.7963 \\ -0.1501 \\ -0.01435 \end{bmatrix}$	$\begin{bmatrix} -2.776e-16 & 6.123e-17 & 1 \\ -6.123e-17 & & 1-6.123e-17 \\ & -1-6.123e-17 & -2.776e-16 \end{bmatrix}$

5. Visualization of Puma 560 robot for Procedure 3.6.

a) b) c) d)



6. Inverse kinematics joint vector for Procedure 3.8. What is your observation?

[2.6486 -3.9270 0.0940 2.5326 0.9743 0.3734]

The joint angles appear to be within reasonable limits for the robot, with angles for joints 1 to 6 being within standard ranges for a Puma 560.

7. Correct arm configuration for Procedure 3.9.

```
disp('Right hand, elbow up:');
```

Right hand, elbow up:

```
disp(q_right_elbow_up);
```

-0.0000   0.7854   3.1416   -0.0000   0.7854   0.0000

Right Hand, Elbow up.

8. What can be observed for Procedure 3.10?

```
%3.10
T_unreachable = SE3(10, 10, 10);
q_unreachable = p560.ikine6s(T_unreachable);  % Inverse kinematics for unreachable point
```

Warning: point not reachable

```
disp('Inverse kinematics result for unreachable point:');
```

Inverse kinematics result for unreachable point:

```
disp(q_unreachable);
```

NaN NaN NaN NaN NaN NaN

The output joint angles are all NaN, which is the default result returned when the inverse kinematic solver can not find a valid solution.

9. MATLAB code for the entire procedure.

```matlab
%3.1
mdl_puma560;

%3.2
p560;

% 3.3
disp('Zero angle configuration:');
qz = p560.qz;
disp(p560.qz);
disp('Ready configuration:');
qr = p560.qr;
disp(p560.qr);
disp('Stretch configuration:');
qs = p560.qs;
disp(p560.qs);
disp('Nominal configuration:');
qn = p560.qn;
disp(p560.qn);

% 3.4
p560.tool = SE3(0, 0, 0.2);

%3.5
T_zero = p560.fkine(qz);  % Forward kinematics for zero angle configuration
T_ready = p560.fkine(qr);  % Forward kinematics for ready configuration
T_stretch = p560.fkine(qs);  % Forward kinematics for stretch configuration
T_nominal = p560.fkine(qn);  % Forward kinematics for nominal configuration

disp('Position and Orientation (TCP) for Zero Angle:');
disp(T_zero);
disp('Position and Orientation (TCP) for Ready:');
disp(T_ready);
disp('Position and Orientation (TCP) for Stretch:');
disp(T_stretch);
disp('Position and Orientation (TCP) for Nominal:');
disp(T_nominal);

%3.6
p560.plot3d(qz);  % Visualization for zero angle configuration
hold on;
title('Puma 560 - Zero Angle Configuration');

p560.plot3d(qr);  % Visualization for ready configuration
hold on;
title('Puma 560 - Ready Configuration');

p560.plot3d(qs);  % Visualization for stretch configuration
hold on;
title('Puma 560 - Stretch Configuration');

p560.plot3d(qn);  % Visualization for nominal configuration
hold on
title('Puma 560 - Nominal Configuration');

%3.7
p560.tool = SE3();  % Reset tool to no extension

%3.8
T_nominal = p560.fkine(qn);  % Forward kinematics for nominal configuration
q_inv = p560.ikine6s(T_nominal);  % Inverse kinematics

disp('Inverse kinematics joint vector for nominal configuration:');
disp(q_inv);

%3.9
q_left_elbow_up = p560.ikine6s(T_nominal, 'l', 'u');  % Left hand, elbow up
q_left_elbow_down = p560.ikine6s(T_nominal, 'l', 'd');  % Left hand, elbow down
q_right_elbow_up = p560.ikine6s(T_nominal, 'r', 'u');  % Right hand, elbow up
q_right_elbow_down = p560.ikine6s(T_nominal, 'r', 'd');  % Right hand, elbow down

% Display the results
disp('Left hand, elbow up:');
disp(q_left_elbow_up);
disp('Left hand, elbow down:');
disp(q_left_elbow_down);
disp('Right hand, elbow up:');
disp(q_right_elbow_up);
disp('Right hand, elbow down:');
disp(q_right_elbow_down);

%3.10
T_unreachable = SE3(10, 10, 10);
q_unreachable = p560.ikine6s(T_unreachable);

% Display the result
disp('Inverse kinematics result for unreachable point:');
disp(q_unreachable);

%3.11
close all;
clear all;

mdl_puma560;  % Load Puma 560 robot model

T1 = SE3(0.8, 0, 0) * SE3.Ry(pi/2);  % First transformation
T2 = SE3(-0.8, 0, 0) * SE3.Rx(pi);  % Second transformation

q1 = p560.ikine6s(T1);  % Inverse kinematics for T1
q2 = p560.ikine6s(T2);  % Inverse kinematics for T2

t = [0:0.05:2]';  % Time vector for trajectory
q = jtraj(q1, q2, t);  % Joint space trajectory from q1 to q2

p560.plot3d(q);  % Plot the robot motion
```

10. Explain in point form what the MATLAB code in 3.11 does.

- Initialize Environment: Clears the workspace and closes all figures for a clean start.
- Load Puma 560 Model: Loads the Puma 560 robot model into MATLAB for further analysis.
- Define End-Effector Poses:

T1: Sets an end-effector pose at [0.8,0,0] with a 90deg rotation around the Y-axis.

T2: Sets an end-effector pose at [−0.8,0,0] with a 180deg rotation around the X-axis.

- Compute Joint Configurations: Calculates joint angles (q1 and q2) using inverse kinematics to achieve poses T1 and T2.
- Generate Trajectory: Computes a smooth joint-space trajectory from q1 to q2 over a time vector (t), ensuring smooth motion.
- Visualize Motion: Plots the robot's movement along the computed trajectory using 3D visualization.

Observation: The robot moves smoothly from the initial pose T1 to the target pose T2. This illustrates how forward kinematics, inverse kinematics, and trajectory generation work together for effective robot control and visualization.