Kumarage R.V

Answer Sheet

Index No: 220343B

ENTC

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

   Serial-chain manipulator with 6 revolute joints

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

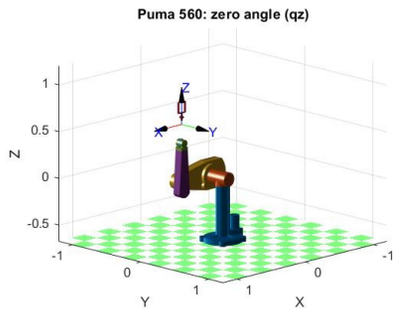
3. Joint coordinate vectors of Puma 560 robot for the following canonical configurations:
   a) zero angle: [0 0 0 0 0 0]
   b) ready : [0 1.5708 -1.5708 0 0 0]
   c) stretch : [0 0 -1.5708 0 0 0]
   d) nominal : [0 0.7854 3.1416 0 0.7854 0]

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

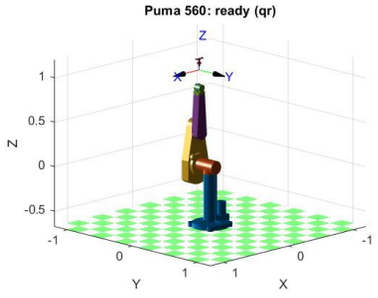
Configuration	**Position**	**Orientation (Rotation Matrix)**
Zero angle	$\begin{bmatrix} 0.4521 \\ -0.1500 \\ 0.6318 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Ready	$\begin{bmatrix} 0.0203 \\ -0.1500 \\ 1.0636 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Stretch	$\begin{bmatrix} 1.0636 \\ -0.1501 \\ -0.0203 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$
Nominal	$\begin{bmatrix} 0.7963 \\ -0.1501 \\ -0.0144 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$

5. Visualization of Puma 560 robot for Procedure 3.6.

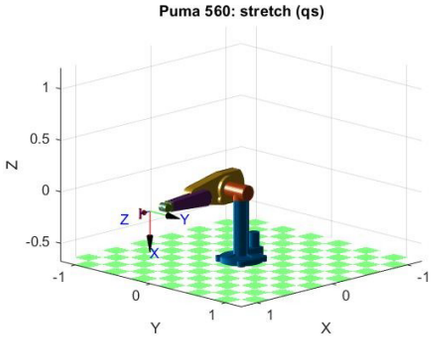
a)



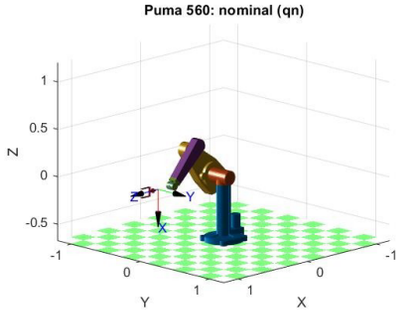
b)



c)



d)

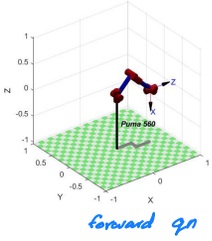


6. Inverse kinematics joint vector for Procedure 3.8. What is your observation?

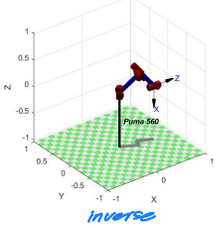
The resulting q_inv matches the original qn almost exactly (numerical difference < 1×10^{-4} rad), confirming that the FK→IK mapping is consistent & the solver is working correctly for reachable configurations.

7. Correct arm configuration for Procedure 3.9.

c) Right hand, elbow up



forward qn



inverse

8. What can be observed for Procedure 3.10?

```
Warning: point not reachable
> In SerialLink/ikine6s (line 573)
  In roboticslab3 (line 73)
    NaN    NaN    NaN    NaN    NaN    NaN

Unreachable: ikine6s returned NaNs (no solution).
```

9. MATLAB code for the entire procedure.

```
%% Tasks for Lab 03 (Puma 560)
%% 3.1 Instantiate Puma 560 robot model
mdl_puma560   % built-in model from the Robotics Toolbox
fprintf('\n Puma 560 object created \n');

%% 3.2 - Robot geometry and DH parameter table
p560; %Puma 560 [Unimation]:: 6 axis, RRRRRR, stdDH, slowRNE - viscous friction; params of 8/95;

%% 3.3 - Joint coordinate vectors for canonical configurations
% Canonical Configurations (Joint Coordinate Vectors)
qz, qr, qs,qn

%% 3.4 - Define tool transform: extension of 200 mm in z of T6 frame
p560.tool = SE3(0,0,0.2);  % extend 200mm along z-axis of T6

%% 3.5 Forward kinematics for canonical configurations
configs = {'zero angle (qz)', 'ready (qr)', 'stretch (qs)', 'nominal (qn)'};
q_vecs = {qz,qr,qs,qn};

for i = 1:4
    q = q_vecs{i};
    T = p560.fkine(q);
    fprintf('\nConfiguration: %s\n', configs{i});
    fprintf('Position (m): '); disp(transl(T));
    disp('Rotation matrix:'); disp(t2r(T));
end

%% 3.6 Graphically display realistic plots for canonical configurations
for i = 1:4
    figure('Name',sprintf('Puma560 - %s',configs{i}));
    % plot3d provides more realistic model (shaded links) if supported
    p560.plot3d(q_vecs{i})
    title(sprintf('Puma 560: %s', configs{i}));
end

%% 3.7  reset the tool transform to zero extension
p560.tool = SE3(0,0,0);  % extend 200mm along z-axis of T6

%% 3.8 perform inverse kinematics
T_nominal = p560.fkine(qn);
disp('Target T_nominal:'); disp(T_nominal.T);
%p560.plot(qn)

% Analytical inverse kinematics (for 6-axis spherical wrist)
q_inv_nominal = p560.ikine6s(T_nominal);
disp('Recovered joint vector (ikine6s):'); disp(q_inv_nominal);
%p560.plot(q_inv_nominal)

fprintf('Difference from qn (radians):\n');
disp(q_inv_nominal - qn);

T_check = p560.fkine(q_inv_nominal);
disp('FK result from IK solution:'); disp(T_check.T);
fprintf('Pose difference (should be near zero):\n');
disp(T_check.T - T_nominal.T);

%% 3.9 Inverse Kinematics for a New Pose
configs = {'lu','ld','ru','rd'};     % 3.9 options (wrist default = 'n')
sol = struct([]);

for i = 1:numel(configs)
       q_try = p560.ikine6s(T_nominal, configs{i});
      sol(i).cfg = configs{i};
      sol(i).q   = q_try;
      sse = sum((q_try - qn).^2);      % how close to the provided nominal vector
      sol(i).sse = sse;
      fprintf('%s  SSE=%.6f\n', configs{i}, sse);
      figure; p560.plot3d(q_try); title(['IK: ', configs{i}]);
end

%% 3.10 Give an unreachable point
 T_bad = SE3(2.0, 0, 0);             % way beyond reach
 q_bad = p560.ikine6s(T_bad);        % spherical-wrist IK
 disp(q_bad)
 if any(isnan(q_bad))
     disp('Unreachable: ikine6s returned NaNs (no solution).');
 end

%% observe Puma 560 motion
close all; clear all;
mdl_puma560
T1 = SE3(0.8, 0, 0) * SE3.Ry(pi/2);
T2 = SE3(-0.8,0, 0) * SE3.Rx(pi);

% IK (spherical wrist)
q1 = p560.ikine6s(T1);
q2 = p560.ikine6s(T2);

% Quintic joint trajectory
t = (0:0.05:2)';                    % time vector
q = jtraj(q1, q2, t);               % joint waypoints

% Animate realistic 3D model
p560.plot3d(q);
```

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

- mdl_puma560 – loads the predefined 6-DoF PUMA 560 model.
- T1, T2 → create 2 target tool poses.
  - T_1 : 0.8m along +X, oriented 90° about Y.
  - T_2 : 0.8m along -X, oriented 180° about X.
- ikine6s → Computes inverse-kinematics joint angles for each target pose.
- jtraj(q1, q2, t) → generate a smooth joint-space trajectory between q1 & q2 using a quintic polynomial with zero initial/final velocity & acceleration
- plot3d(q) → animates a realistic 3-D motion of the PUMA 560 executing that trajectory.

The robot moves smoothly from the first pose (reading right) to the second pose (reading left) through physically valid intermediate configurations - demonstrating forward + inverse kinematics and trajectory generation working together.