Answer Sheet

Index No: 220343B

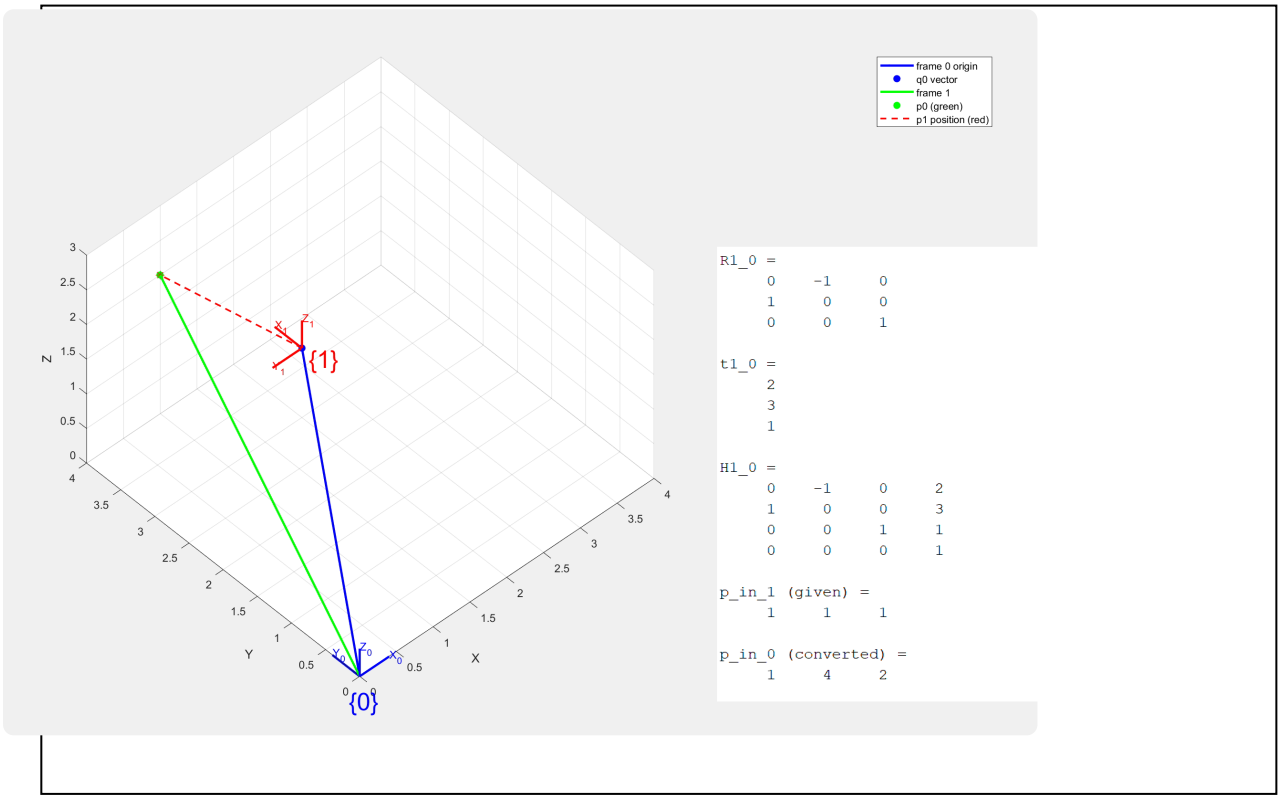
1. Homogeneous transformation matrix H_1^0 for 3.4.

```
H1_0 =
     0    -1     0     2
     1     0     0     3
     0     0     1     1
     0     0     0     1
```

2. MATLAB code for 3.1 ~ 3.6.

```
% Lab02 - tasks 3.1 to 3.6
clc; clear; close all;
% ---
q0_in_0 = [2; 3; 1];         % q0 expressed in frame {0}
angle_deg = 90;              % rotation about Z to get frame {-9}
p_in_1 = [1; 1; 1];   % point p expressed in frame {1}

% 3.1 Visualize coordinate system {0}
figure('Name','Lab02 3.1-3.6');
trplot(eye(4), 'frame', '0', 'color', 'b', 'length', 0.4, 'thick', 2);
hold on; grid on; axis equal;
axis([0 4 0 4 0 3]); xlabel('X'); ylabel('Y'); zlabel('Z');

% 3.2 Obtain rotation R1_0 and translation t1_0
theta = deg2rad(angle_deg);
R1_0 = rotz(theta);         % rotation matrix 3x3 (frame1 relative to frame0)
t1_0 = q0_in_0;             % translation vector (origin of {1} in {0})

% Display R and t
disp('R1_0 ='); disp(R1_0);
disp('t1_0 ='); disp(t1_0);

% 3.3 Visualize q0 in the figure using blue color (arrow from origin)
% plot a blue arrow from origin to q0_in_0
plot3([0 q0_in_0(1)], [0 q0_in_0(2)], [0 q0_in_0(3)], 'b-', 'LineWidth', 2);
plot3(q0_in_0(1), q0_in_0(2), q0_in_0(3), 'bo', 'MarkerSize', 6, 'MarkerFaceColor','b');

% 3.4 Obtain H1_0 and visualize coordinate frame {1} using red color
H1_0 = rt2tr(R1_0, t1_0);   % homogeneous transform (Robotics Toolbox)
% (alternatively: H1_0 = [R1_0 t1_0; 0 0 0 1];)
trplot(H1_0, 'frame', '1', 'color', 'r', 'length', 0.4, 'thick', 2);

% show H1_0 numeric
disp('H1_0 ='); disp(H1_0);

% 3.5 Find p0 and visualize it in the same figure using green color
% Convert p1 (in {1}) to homogeneous, then to {0}
P1_h = [p_in_1; 1];           % homogeneous coordinates in frame {1}
P0_h = H1_0 * P1_h;           % p expressed in frame {0}
p_in_0 = P0_h(1:3);

% plot p0 as green arrow from origin
plot3([0 p_in_0(1)], [0 p_in_0(2)], [0 p_in_0(3)], 'g-', 'LineWidth', 2);
plot3(p_in_0(1), p_in_0(2), p_in_0(3), 'go', 'MarkerFaceColor','g');

disp('p_in_1 (given) ='); disp(p_in_1');
disp('p_in_0 (converted) ='); disp(p_in_0');

% 3.6 Visualize p1 in the same figure using red color
% To plot p1 in the global figure we show its location in {0} coordinates (already computed)
plot3([q0_in_0(1) p_in_0(1)], [q0_in_0(2) p_in_0(2)], [q0_in_0(3) p_in_0(3)], 'r--', 'LineWidth', 1.5);
plot3(p_in_0(1), p_in_0(2), p_in_0(3), 'r*', 'MarkerSize', 8);

legend({'frame 0 origin','q0 vector','frame 1','p0 (green)','p1 position (red)'}, 'Location', 'northeastoutside');
hold off;
```

3. Final output MATLAB figure for the operations in 3.1 ~ 3.6.




```
R1_0 =
     0    -1     0
     1     0     0
     0     0     1

t1_0 =
     2
     3
     1

H1_0 =
     0    -1     0     2
     1     0     0     3
     0     0     1     1
     0     0     0     1

p_in_1 (given) =
     1     1     1

p_in_0 (converted) =
     1     4     2
```

4. Homogeneous transformation matrix H_0^1 for 3.8.

```
H0_1 (inverse transformation) =
     0     1     0    -3
    -1     0     0     2
     0     0     1    -1
     0     0     0     1
```

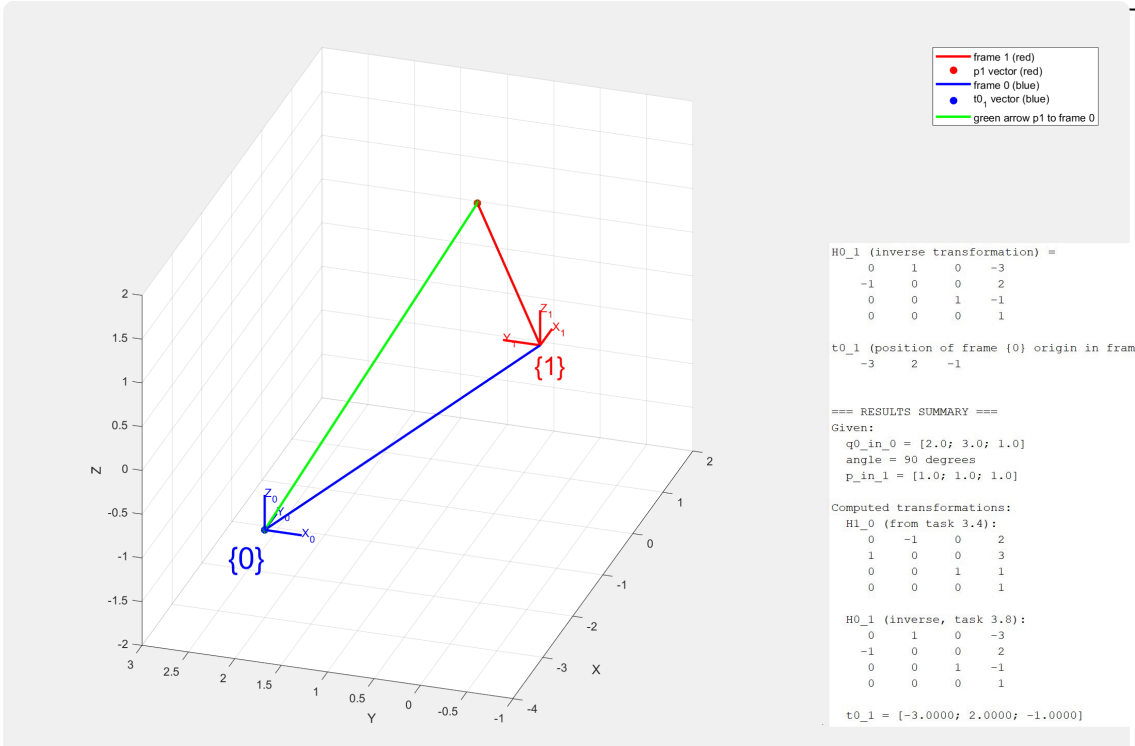
5. t_0^1 for 3.10.

```
t0_1 (position of frame {0} origin in frame {1})
    -3     2    -1
```

6. MATLAB code for 3.7 ~ 3.11.

```
% Lab02 - tasks 3.7 to 3.11
% Continuation from tasks 3.1-3.6

clc; clear; close all;

% ---
% Given values from previous tasks
q0_in_0 = [2; 3; 1];          % q0 expressed in frame {0}
angle_deg = 90;               % rotation about Z to get frame {1}
p_in_1 = [1; 1; 1];           % point p expressed in frame {1}
% ---

% First, we need to compute the transformations from previous tasks
theta = deg2rad(angle_deg);
R1_0 = rotz(theta);           % rotation matrix 3x3 (frame1 relative to frame0)
t1_0 = q0_in_0;               % translation vector (origin of {1} in {0})
H1_0 = rt2tr(R1_0, t1_0);     % homogeneous transform from {0} to {1}

% 3.7 Use a new MATLAB figure to visualize coordinate system {1} and p1 using red color
figure('Name','Lab02 3.7-3.11');
trplot(eye(4), 'frame', '1', 'color', 'r', 'length', 0.4, 'thick', 2);
hold on; grid on; axis equal;
axis([-4 2 -1 3 -2 2]); xlabel('X'); ylabel('Y'); zlabel('Z');

% Visualize p1 in frame {1} using red color
plot3([0 p_in_1(1)], [0 p_in_1(2)], [0 p_in_1(3)], 'r-', 'LineWidth', 2);
plot3(p_in_1(1), p_in_1(2), p_in_1(3), 'ro', 'MarkerSize', 6, 'MarkerFaceColor','r');

% 3.8 Obtain the homogeneous transformation matrix H0_1 (inverse of H1_0)
H0_1 = inv(H1_0);
% Alternative method using the inverse formula: H0_1 = [R1_0' -R1_0'*t1_0; 0 0 0 1];

disp('H0_1 (inverse transformation) =');
disp(H0_1);

% 3.9 Visualize frame {0} with blue color
trplot(H0_1, 'frame', '0', 'color', 'b', 'length', 0.4, 'thick', 2);

% 3.10 Find t0_1 and visualize it on the figure with blue color
% t0_1 is the translation part of H0_1, which represents the position of frame {0} origin in frame {1}
t0_1 = H0_1(1:3, 4);

% Visualize t0_1 as blue arrow from origin of frame {1}
plot3([0 t0_1(1)], [0 t0_1(2)], [0 t0_1(3)], 'b-', 'LineWidth', 2);
plot3(t0_1(1), t0_1(2), t0_1(3), 'bo', 'MarkerSize', 6, 'MarkerFaceColor','b');

disp('t0_1 (position of frame {0} origin in frame {1}) =');
disp(t0_1');

% 3.11 Visualize a green arrow from the tip of p1 to the origin of frame {0}
% The origin of frame {0} in frame {1} is at t0_1
% The tip of p1 is at p_in_1
% So we draw an arrow from p_in_1 to t0_1
plot3([p_in_1(1) t0_1(1)], [p_in_1(2) t0_1(2)], [p_in_1(3) t0_1(3)], 'g-', 'LineWidth', 2);
plot3([p_in_1(1) t0_1(1)], [p_in_1(2) t0_1(2)], [p_in_1(3) t0_1(3)], 'g*', 'MarkerSize', 6);

% Add legend
legend({'frame 1 (red)', 'p1 vector (red)', 'frame 0 (blue)', 't0_1 vector (blue)', 'green arrow p1 to frame 0'}, 'location', 'northeastoutside');

% Display results summary
fprintf('\n=== RESULTS SUMMARY ===\n');
fprintf('Given:\n');
fprintf('  q0_in_0 = [%.1f; %.1f; %.1f]\n', q0_in_0);
fprintf('  angle = %.0f degrees\n', angle_deg);
fprintf('  p_in_1 = [%.1f; %.1f; %.1f]\n', p_in_1);

fprintf('\nComputed transformations:\n');
fprintf('  H1_0 (from task 3.4):\n');
disp(H1_0);
fprintf('  H0_1 (inverse, task 3.8):\n');
disp(H0_1);
fprintf('  t0_1 = [%.4f; %.4f; %.4f]\n', t0_1);

hold off;
```

7. Final output MATLAB figure for the operations in 3.7 ~ 3.11.




```
H0_1 (inverse transformation) =
     0     1     0    -3
    -1     0     0     2
     0     0     1    -1
     0     0     0     1

t0_1 (position of frame {0} origin in frame {1}) =
    -3     2    -1


=== RESULTS SUMMARY ===
Given:
  q0_in_0 = [2.0; 3.0; 1.0]
  angle = 90 degrees
  p_in_1 = [1.0; 1.0; 1.0]

Computed transformations:
  H1_0 (from task 3.4):
     0    -1     0     2
     1     0     0     3
     0     0     1     1
     0     0     0     1

  H0_1 (inverse, task 3.8):
     0     1     0    -3
    -1     0     0     2
     0     0     1    -1
     0     0     0     1

  t0_1 = [-3.0000; 2.0000; -1.0000]
```

8. Homogeneous transformation table.

Requirement	MATLAB script to satisfy the requirement	Homogeneous transformation matrix result
$o_0x_0y_0z_0$ to $o_1x_1y_1z_1$	Script for H1_0: t1_0 = [0.0; 1.0; 1.0]; R1_0 = eye(3); H1_0 = rt2tr(R1_0, t1_0);	H1_0 (Frame {0} to Frame {1} - Table surface): 1 0 0 0 0 1 0 1 0 0 1 1 0 0 0 1
$o_0x_0y_0z_0$ to $o_2x_2y_2z_2$	Script for H2_0: t2_0 = [-0.5; 1.5; 1.0]; R2_0 = roty(deg2rad(45)); H2_0 = rt2tr(R2_0, t2_0);	H2_0 (Frame {0} to Frame {2} - Box location): 1.0000 0 0 -0.5000 0 1.0000 0 1.5000 0 0 1.0000 1.0000 0 0 0 1.0000
$o_0x_0y_0z_0$ to $o_3x_3y_3z_3$	Script for H3_0: t3_0 = [-0.5; 1.5; 3.0]; R3_0 = rotx(deg2rad(180)) * rotz(deg2rad(90)); H3_0 = rt2tr(R3_0, t3_0);	H3_0 (Frame {0} to Frame {3} - Camera location): 0 1.0000 0 -0.5000 1.0000 0 0 1.5000 0 0 -1.0000 3.0000 0 0 0 1.0000