BAHIR DAR UNIVERSITY

BAHIR DAR INSTITUTE OF TECHNOLOGY

FACULITY OF ELECTRICAL AND COMPUTER ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

Introduction to Robotics (EEng 5182)

Individual Assignment I

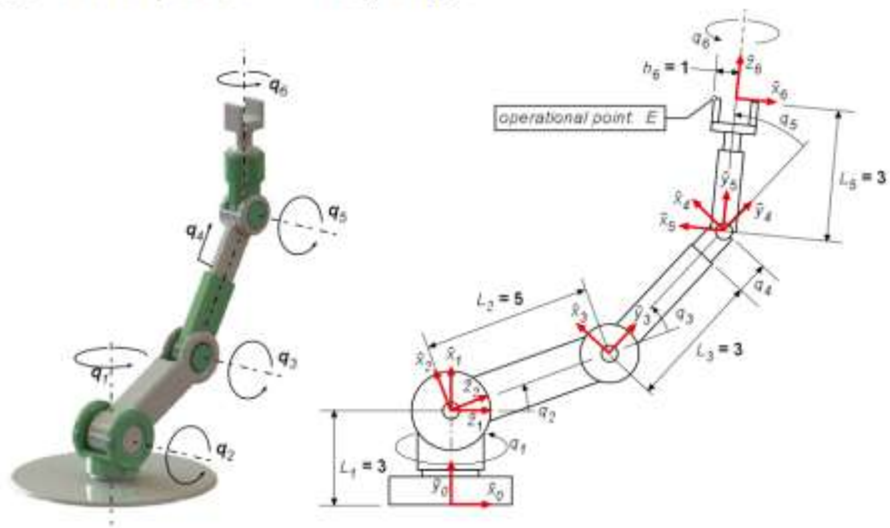
Name Sefinew Yeshaneh

Id 1307579

Submitted to mr Mehari M

Submission date 29/07/2017 EC

(6) For the manipulator shown in the adjacent figure



► Write Matlab code to calculate the homogeneous transforms derived above as a function of the joint variables q_1, q_2, q_3, q_4, and q_5.

```
% Define symbolic variables
syms q1 q2 q3 q4 q5 q6 L1 L2 L3 L5 real
% Given link lengths (example values, adjust as needed)
L1 = 3;
L2 = 5;
L3 = 3;
L5 = 3;
% Homogeneous transformation matrices
T01 = [0    sin(q1)  cos(q1)  0;
       1    0        0        L1;
       0    cos(q1) -sin(q1)  0;
       0    0        0        1];
T12 = [cos(q2)  0  sin(q2)  0;
       0        1  0        0;
      -sin(q2)  0  cos(q2)  0;
       0        0  0        1];
T23 = [cos(q3)  sin(q3)  0    0;
       0        0       -1    0;
      -sin(q3)  cos(q3)  0    L2;
       0        0        0    1];
T34 = [1  0  0       0;
       0  1  0       L3+q4;
```

```
     0  0  1       0;
     0  0  0       1];
T45 = [cos(q5)  sin(q5)  0  0;
     -sin(q5)  cos(q5)  0  0;
     0       0       1  0;
     0       0       0  1];
T56 = [-cos(q6)  sin(q6)  0  0;
     0        0       1  L5;
     sin(q6)   cos(q6)  0  0;
     0        0       0  1];
% Function to evaluate transforms at specific joint angles
function evalTransforms(q_vals)
   % Substitute joint angle values
   T01_val = double(subs(T01, [q1 q2 q3 q4 q5 q6], q_vals));
   T12_val = double(subs(T12, [q1 q2 q3 q4 q5 q6], q_vals));
   T23_val = double(subs(T23, [q1 q2 q3 q4 q5 q6], q_vals));
   T34_val = double(subs(T34, [q1 q2 q3 q4 q5 q6], q_vals));
   T45_val = double(subs(T45, [q1 q2 q3 q4 q5 q6], q_vals));
   T56_val = double(subs(T56, [q1 q2 q3 q4 q5 q6], q_vals));
   % Display results
   disp('T01:'); disp(T01_val);
   disp('T12:'); disp(T12_val);
   disp('T23:'); disp(T23_val);
   disp('T34:'); disp(T34_val);
   disp('T45:'); disp(T45_val);
   disp('T56:'); disp(T56_val);
end
% Example usage:
% q_vals = [pi/4, pi/6, pi/3, 0.5, pi/4, pi/6]; % Replace with desired joint angles
% evalTransforms(q_vals);
```

- Write Matlab code to calculate the homogeneous transformation matrices that describe the frame displacements relative to the ground frame $\{0\}$ (i.e. numerically evaluate $T_1^0, T_2^0, T_3^0, T_4^0, T_5^0, T_6^0$).

```
% Define symbolic variables
syms q1 q2 q3 q4 q5 q6 L1 L2 L3 L5 real
% Given link lengths (use actual values from the problem)
L1 = 3; L2 = 5; L3 = 3; L5 = 3;
% Homogeneous transforms between successive frames (from Problem 6.1)
T01 = [0     sin(q1)  cos(q1)  0;
      1    0       0       L1;
      0    cos(q1) -sin(q1)  0;
      0    0       0       1];
T12 = [cos(q2)  0  sin(q2)  0;
```

```
      0       1  0       0;
     -sin(q2)  0  cos(q2)  0;
      0       0  0       1];
T23 = [cos(q3)  sin(q3)  0    0;
      0       0      -1    0;
     -sin(q3)  cos(q3)  0    L2;
      0       0       0    1];
T34 = [1  0  0       0;
      0  1  0       L3+q4;
      0  0  1       0;
      0  0  0       1];
T45 = [cos(q5)  sin(q5)  0  0;
     -sin(q5)  cos(q5)  0  0;
      0       0       1  0;
      0       0       0  1];
T56 = [-cos(q6)  sin(q6)  0  0;
      0        0       1  L5;
      sin(q6)   cos(q6)  0  0;
      0        0       0  1];
% Compute transforms relative to ground frame {0}
T02 = T01 * T12;
T03 = T02 * T23;
T04 = T03 * T34;
T05 = T04 * T45;
T06 = T05 * T56;

% Function to evaluate all ground-relative transforms at given joint angles
function evalGroundTransforms(q_vals)
   % Substitute joint values and convert to numeric
   T01_val = double(subs(T01, [q1 q2 q3 q4 q5 q6], q_vals));
   T02_val = double(subs(T02, [q1 q2 q3 q4 q5 q6], q_vals));
   T03_val = double(subs(T03, [q1 q2 q3 q4 q5 q6], q_vals));
   T04_val = double(subs(T04, [q1 q2 q3 q4 q5 q6], q_vals));
   T05_val = double(subs(T05, [q1 q2 q3 q4 q5 q6], q_vals));
   T06_val = double(subs(T06, [q1 q2 q3 q4 q5 q6], q_vals));

   % Display results
   disp('T01 (Frame {1} relative to {0}):'); disp(T01_val);
   disp('T02 (Frame {2} relative to {0}):'); disp(T02_val);
   disp('T03 (Frame {3} relative to {0}):'); disp(T03_val);
   disp('T04 (Frame {4} relative to {0}):'); disp(T04_val);
   disp('T05 (Frame {5} relative to {0}):'); disp(T05_val);
   disp('T06 (Frame {6} relative to {0}):'); disp(T06_val);
end
```

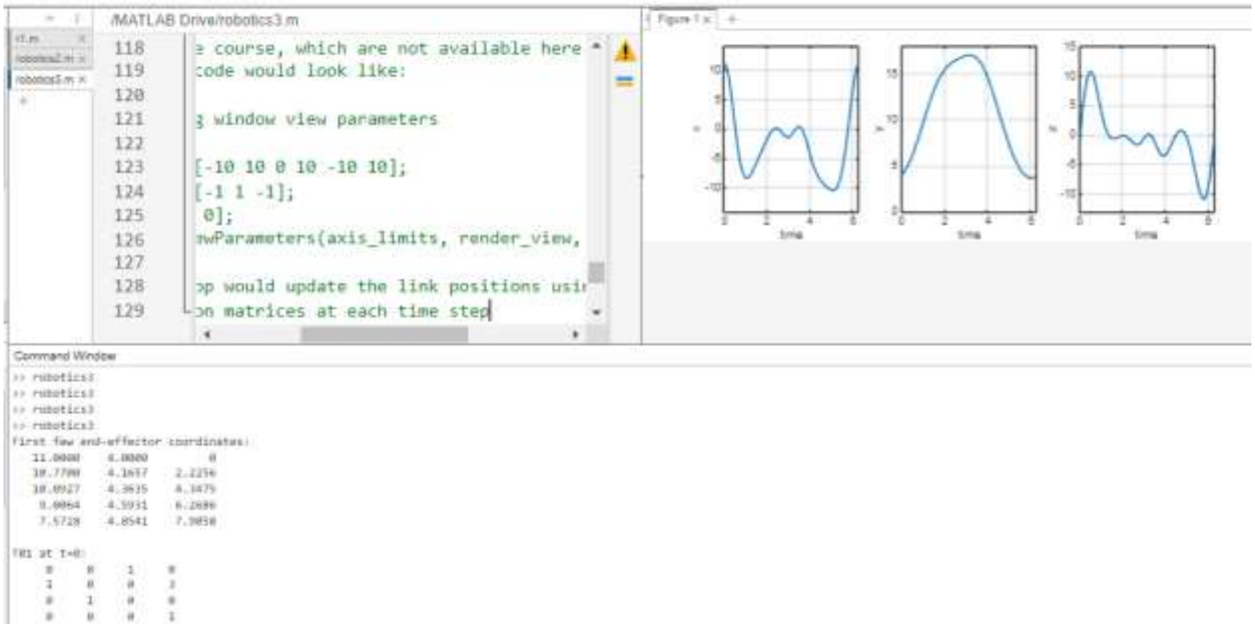
```
% Example usage:
% q_vals = [pi/4, pi/6, pi/3, 0.5, pi/4, pi/6]; % Replace with desired angles
% evalGroundTransforms(q_vals);
```

part 3

```
% This script plots the x, y, z coordinates of the end-effector E
% as a function of time and animates the manipulator
% Define symbolic variables
syms q1 q2 q3 q4 q5 q6 L1 L2 L3 L5 t real
% Define constants
L1 = 3;
L2 = 5;
L3 = 3;
L5 = 3;
h6 = 1;
% Define joint variables as functions of time
q1 = -pi*sin(t);
q2 = pi/4*(1-cos(t));
q3 = pi/4*sin(t);
q4 = 1/2*L3*(1-cos(t));
q5 = -pi/4*sin(t);
q6 = pi/4*sin(t);
% Define the transformation matrices T01 to T56
T01 = [0, sin(q1), cos(q1), 0;
    1, 0, 0, L1;
    0, cos(q1), -sin(q1), 0;
    0, 0, 0, 1];
T12 = [cos(q2), 0, sin(q2), 0;
    0, 1, 0, 0;
    -sin(q2), 0, cos(q2), 0;
    0, 0, 0, 1];
T23 = [cos(q3), sin(q3), 0, 0;
    0, 0, -1, 0;
    -sin(q3), cos(q3), 0, L2;
    0, 0, 0, 1];
T34 = [1, 0, 0, 0;
    0, 1, 0, L3+q4;
    0, 0, 1, 0;
    0, 0, 0, 1];
T45 = [cos(q5), sin(q5), 0, 0;
    -sin(q5), cos(q5), 0, 0;
    0, 0, 1, 0;
```

```
    0, 0, 0, 1];
T56 = [-cos(q6), sin(q6), 0, 0;
    0, 0, 1, L5;
    sin(q6), cos(q6), 0, 0;
    0, 0, 0, 1];
% Calculate the complete transformation matrix T06
T06 = T01*T12*T23*T34*T45*T56;
% Create a function handle to evaluate T06 at specific time instances
endPos = @(t0) double(subs(T06, t, t0));
% Set up time scale and initialize coordinates matrix
timeScale = linspace(0, 2*pi, 100);
coords = zeros(length(timeScale), 3);
% Generate the coordinates of the end effector
for i = 1:length(timeScale)
  p = endPos(timeScale(i)) * [-h6; 0; 0; 1];
  coords(i,:) = p(1:3)';
end
% Plot the results
figure('Position', [100, 100, 900, 300]);
% X coordinate plot
subplot(1, 3, 1);
plot(timeScale, coords(:,1), 'LineWidth', 1.5);
xlabel('time');
ylabel('x');
grid on;
xlim([0, 2*pi]);
ylim([-14, 14]);
% Y coordinate plot
subplot(1, 3, 2);
plot(timeScale, coords(:,2), 'LineWidth', 1.5);
xlabel('time');
ylabel('y');
grid on;
xlim([0, 2*pi]);
ylim([0, 18]);
% Z coordinate plot
subplot(1, 3, 3);
plot(timeScale, coords(:,3), 'LineWidth', 1.5);
xlabel('time');
ylabel('z');
grid on;
xlim([0, 2*pi]);
ylim([-13, 15]);
% Display the first few coordinates to verify
```

```
disp('First few end-effector coordinates:');
disp(coords(1:5,:));
% Calculate and display transformation matrices at t=0 for verification
t0 = 0;
disp('T01 at t=0:');
disp(double(subs(T01, t, t0)));
disp('T02 at t=0:');
disp(double(subs(T01*T12, t, t0)));
disp('T03 at t=0:');
disp(double(subs(T01*T12*T23, t, t0)));
disp('T06 at t=0:');
disp(double(subs(T06, t, t0)));
```



Part 4 anmation

```
% MATLAB code to animate the manipulator  to
function animate_manipulator()
   % Initialize rendering (UW toolbox)
   f_handle = 1;
   axis_limits = [-10 10 0 10 -10 10];
   render_view = [-1 1 -1];
   view_up = [0 1 0];
   SetRenderingViewParameters(axis_limits, render_view, view_up, f_handle);
   % Link lengths
   L1 = 3; L2 = 5; L3 = 3; L5 = 3;
   % Time parameters
   t_max = 2*pi;  % 2π seconds animation
```

```
  dt = 0.05;    % Time step
  % Create links (simplified visualization)
  link_colors = {[1 0 0], [0 1 0], [0 0 1], [1 1 0], [1 0 1], [0 1 1]};
  for i = 1:6
    links(i) = CreateLinkRendering(1, 0.5, 6, 1, -1, link_colors{i}, 0, f_handle);
  end
  % Animation loop
  for t = 0:dt:t_max
    % Joint angles (time-dependent)
    q1 = -pi*sin(t);
    q2 = pi/4*(1-cos(t));
    q3 = pi/4*sin(t);
    q4 = 0.5*L3*(1-cos(t));
    q5 = -pi/4*sin(t);
    q6 = pi/4*sin(t);
    % Compute transforms
    T01 = [0 sin(q1) cos(q1) 0; 1 0 0 L1; 0 cos(q1) -sin(q1) 0; 0 0 0 1];
    T12 = [cos(q2) 0 sin(q2) 0; 0 1 0 0; -sin(q2) 0 cos(q2) 0; 0 0 0 1];
    T23 = [cos(q3) sin(q3) 0 0; 0 0 -1 0; -sin(q3) cos(q3) 0 L2; 0 0 0 1];
    T34 = [1 0 0 0; 0 1 0 L3+q4; 0 0 1 0; 0 0 0 1];
    T45 = [cos(q5) sin(q5) 0 0; -sin(q5) cos(q5) 0 0; 0 0 1 0; 0 0 0 1];
    T56 = [-cos(q6) sin(q6) 0 0; 0 0 1 L5; sin(q6) cos(q6) 0 0; 0 0 0 1];
    % Chain transforms
    T02 = T01*T12; T03 = T02*T23; T04 = T03*T34; T05 = T04*T45; T06 = T05*T56;
    % Update link positions
    UpdateLink(links(1), T01);
    UpdateLink(links(2), T02);
    UpdateLink(links(3), T03);
    UpdateLink(links(4), T04);
    UpdateLink(links(5), T05);
    UpdateLink(links(6), T06);

    title(sprintf('Time: %.2f sec', t));
    drawnow;
  end
end
```