TTK4130 Assignment 3

Problem 1

a) C

b) A is false, $(a_1 \times a_2) \times (a_2 \times a_3) = a_3 \times a_1 = a_2$

c)

d) D is correct,

$$R_x(30^\circ) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(30) & -\sin(30) \\ 0 & \sin(30) & \cos(30 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.866 & -0.5 \\ 0 & 0.5 & 0.860 \end{bmatrix}$$

e) B, euler angles

Problem 2

a) Recognize that point B is the origin of frame 1 and point C is the origin of frame 2. The rotations and translation between each frame is then:

$$R_1^0 = R_{z_0}(q_1) = \begin{bmatrix} \cos q_1 & -\sin q_1 & 0 \\ \sin q_1 & \cos q_1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$B^0 = o_1^0 = \begin{bmatrix} L_1 \cdot \cos(q_1) \\ L_1 \cdot \sin(q_1) \\ 0 \end{bmatrix}$$

$$R_2^1 = R_{y_1}(q_2) = \begin{bmatrix} \cos q_2 & 0 & \sin q_2 \\ 0 & 1 & 0 \\ -\sin q_2 & 0 & \cos q_2 \end{bmatrix}$$

$$C^1 = O_2^1 = \begin{bmatrix} L_2 \cdot \cos(q_2) \\ 0 \\ -L_2 \cdot \sin(q_2) \end{bmatrix}$$

I find $C^0 = O_2^0$ by using homogenous transformations:

$$H_2^0 = \begin{bmatrix} R_2^0 & O_2^0 \\ O_{1\times3} & 1 \end{bmatrix} = H_1^0 H_2^1 = \begin{bmatrix} R_1^0 & O_1^0 \\ O_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} R_2^1 & O_2^1 \\ O_{1\times3} & 1 \end{bmatrix}$$

Using Matlab as shown, I extract $O_2^0 = C^0$:

$$C^0 = \begin{bmatrix} (L_1 + L_2 \cos q_2) \cdot \cos q_1 \\ (L_1 + L_2 \cos q_2) \cdot \sin q_1 \\ -L_2 \sin q_1 \end{bmatrix}$$

b) $(\omega_{01}^0)^\times = \dot{R}_1^0 (R_1^0)^T \quad , \quad (\omega_{02}^0)^\times = \dot{R}_2^0 (R_2^0)^T$

Differentiate R_1^0 and R_2^0 and do the matrix multiplication using Matlab. Find the following angular velocities:

$$\omega_{01}^0 = \begin{bmatrix} 0 \\ 0 \\ \dot{q}_1 \end{bmatrix} \qquad \omega_{02}^0 = \begin{bmatrix} -\sin(q_1)\,\dot{q}_2 \\ \cos(q_1)\,\dot{q}_2 \\ \dot{q}_1 \end{bmatrix}$$

c) $$v_B = \omega_{01}^0 \times B^0 = \begin{bmatrix} -L_1 \sin(q_1)\dot{q}_1 \\ L_1 \cos(q_1)\dot{q}_1 \\ 0 \end{bmatrix}$$

$$v_c = \omega_{02}^0 \times C^0 = \begin{bmatrix} -(L_1 \sin(q_1) + L_2 (\cos(q_2)\sin(q_1)))\dot{q}_1 - \cos(q_1)\sin(q_2)\dot{q}_2 \\ (L_1 \cos(q_1) + L_2 (\cos(q_1)\cos(q_2)))\dot{q}_1 - \sin(q_1)\sin(q_2)\dot{q}_2 \\ -(L_1 + L_2 \cos(q_2)\dot{q}_2 \end{bmatrix}$$

d) $$J(q) = \begin{bmatrix} -L_1 \sin(q_1) + L_2(\sin(q_1)\cos(q_2)) & -\cos(q_1)\sin(q_2) \\ L_1\cos(q_1) + L_2(\cos(q_1)\cos(q_2)) & -\sin(q_1)\sin(q_2) \\ 0 & -L_1 + L_2\cos(q_2) \end{bmatrix}$$

Problem 2

Code used to solve problem 2 of TTK4130 Assignment 3

```
syms q1 q2 L1 L2 real
syms dq1 dq2 real

q = [q1;q2];

% Coordinates of point B and C
[B, C] = joint_position(q, L1, L2);

% Rotation of frame 1 about z0
R10 =   [cos(q1) -sin(q1) 0;
        sin(q1) cos(q1) 0;
        0 0 1];

% Rotation of frame 2 about y1
R21 =   [cos(q2) 0 sin(q2);
        0 1 0;
        -sin(q2) 0 cos(q2)];

% Rotation of frame 2 in terms of frame 0
R20 = R10*R21;

% Differentiate the rotational matrices
dR10 = diff(R10, q1)*dq1;
dR20 = diff(R20, q1)*dq1 + diff(R20, q2)*dq2;

% Angular velocities in terms of frame 0
Omega1 = simplify(dR10 * R10.');
omega1 = [Omega1(3,2);Omega1(1,3);Omega1(2,1)];

Omega2 = simplify(dR20 * R20.');
omega2 = [Omega2(3,2);Omega2(1,3);Omega2(2,1)];

% Linear velocity
vB = simplify(cross(omega1,B));
vC = simplify(cross(omega2,C));
```

```
% Function used to calculate position of point B and C
function [B, C] = joint_position(q, L1, L2)
    q1 = q(1);
    q2 = q(2);

    % Rotation of frame 1 about z0
    R10 =   [cos(q1) -sin(q1) 0;
            sin(q1) cos(q1) 0;
            0 0 1];

    % Rotation of frame 2 about y1
    R21 =   [cos(q2) 0 sin(q2);
            0 1 0;
            -sin(q2) 0 cos(q2)];

    % Origin of frame 1 in terms of frame 0 (point B)
    o10 = [L1*cos(q1); L1*sin(q1); 0];

    % Origin of frame 2 in terms of frame 1 (point C)
    o21 = [L2*cos(q2); 0; -L2*sin(q2)];

    % Homogenous transformation of frame 1 in terms of frame 0
    H10 = [R10 o10; zeros(1,3) 1];
    % Homogenous transformation of frame 2 in terms of frame 1
    H21 = [R21 o21; zeros(1,3) 1];
    % Homogenous transformation of frame 2 in terms of frame 0
    H20 = H10*H21;

    % B coordinates in terms of frame 0
    B = o10;
    % Extract C coordinates in terms of frame 0
    C = [H20(1,4); H20(2,4); H20(3,4)];
end
```

```
Not enough input arguments.

Error in joint_position (line 3)
    q1 = q(1);
```

Problem 3

a) With a vector $\vec{r}$ you can either:

- rotate $\vec{r}$ to a vector $\vec{p}$ with the same coordinates in a different fame. R^c_d will then describe the rotation of $\vec{r}$ within frame c, $r'^c = R^c_d r^c$.
- transform the coordinate frame such that you describe the same vector $\vec{r}$, but with a different basis vectors: $r^d = R^c_d r^c$

b) The rotation from a to b is a simple rotation of $\frac{3}{2}\pi$ around the a_3 axis:

$$R^a_b = R_z\left(\tfrac{3}{2}\pi\right) = \begin{bmatrix} \cos\frac{3}{2}\pi & -\sin\frac{3}{2}\pi & 0 \\ \sin\frac{3}{2}\pi & \cos\frac{3}{2}\pi & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In axis-angle terms this is:

$\theta = \frac{3}{2}\pi$, $\vec{k} = \vec{a}_3$

c) Rotation of $\frac{\pi}{2}$ around $\vec{b}_1$

$$R_c^b = R_x\left(\frac{\pi}{2}\right) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\frac{\pi}{2} & -\sin\frac{\pi}{2} \\ 0 & \sin\frac{\pi}{2} & \cos\frac{\pi}{2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

Axis-angle : $\theta = \frac{\pi}{2}$, $\vec{k} = \vec{b}_1$

d) $$R_a^c = (R_c^a)^T = (R_b^a \cdot R_c^b)^T$$

$$= \left(\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \right)^T = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}^T$$

$$= \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}$$

Using the Matlab implementation of Sheppards method from last assignment yields:

$$\theta = \frac{2}{3}\pi \quad , \quad \vec{k} = \begin{bmatrix} -\sqrt{3}/3 \\ \sqrt{3}/3 \\ \sqrt{3}/3 \end{bmatrix}$$

e) I get the wrong output and don't know why

Code below:

```
clc; clear all; close all;

%% Make rotation matrix function R = R(k, theta) (fill in)
skew = @(w) [0, -w(3), w(2); w(3), 0, -w(1); -w(2), w(1), 0]; % may become helpful
R_func = @(k, theta) cos(theta)*eye(3) + sin(theta)*skew(k) + (1-cos(theta))*k*k'; %
calculate rotation matrix from angle and axis

%% Rotation axis and angles for a->b, b->c (fill in)

kab = [0; 0; 1]; %rotation axis from a to b
thetaab = 3*pi/2; %rotation angle from a to b
kbc = [0; -1; 0]; % rotation axis from b to c
thetabc = pi/2; %rotation angle from b to c

Rab = R_func(kab, thetaab); % rotation matrix from a to b
Rbc = R_func(kbc, thetabc); % rotation matrix from b to c

%% rotation matrix, axis and angle from c->a (fill in)
Rca = (Rab*Rbc).'; % rotation matrix from c to a

[thetaca, kca] = Shepperd(Rca);
thetaca = double(thetaca);
kca = double(kca);

stopAngle = thetaab+thetabc+thetaca;   % what angle to stop rotating (total angle)

%% Calculate rotation matrices (nothing to fill in)
n = 200; % number of discretization points
theta_t = linspace(0, 2*pi, n+1); % we rotate uintill 2pi, but will stop before
dtheta = theta_t(2) - theta_t(1); % angle increments

R_t1 = R_func(kab, dtheta); % small part of a->b rotation
R_t2 = R_func(kbc, dtheta); % small part of b->c rotation
R_t3 = R_func(kca, dtheta); % small part of c->a rotation

Rlist = cell(n+1, 1);
Rlist{1} = eye(3);

for i=(2:n+1)
    if (i <= n/4)
        Ri = Rlist{i-1}*R_t1;
    elseif (i > n/4) && (i <n/2)
        Ri = Rlist{i-1}*R_t2;
    else
        Ri = Rlist{i-1}*R_t3;
    end
    if (theta_t(i) >= stopAngle) % stopAngle, to stop rotating further
        Ri = Rlist{i-1};
    end
    Rlist{i} = Ri;
end
%% Real-time animation (fill in)
ScaleFrame = 5;   % Scaling factor for adjusting the frame size (cosmetic)
FS         = 15;  % Fontsize for text
```

```
SW          = 0.035; % Arrows size
time_display = 0; % initialise time_display
for i =(1:n+1)
    state_animate = Rlist{i}; % what is the 'state' that we animate?
    p     = [5;5;5];  % Position of the single body

    %3D below this point
    figure(1);clf;hold on
    MakeFrame( zeros(3,1),eye(3),ScaleFrame,FS,SW,'a', 'color', 'k')
    MakeFrame( p,state_animate,ScaleFrame,FS,SW,'b', 'color', 'r')
    DrawRectangle(p,state_animate ,'color',[0.5,0.5,0.5]);
    FormatPicture([0;0;2],0.5*[73.8380   21.0967   30.1493])

    if time_display == 0
        display('Hit a key to start animation')
        pause
        tic
    end
    time_display = toc; %get the current clock time
end
```

```
function [theta, k] = Shepperd(R)
    syms z0 z1 z2 z3

    z = [z0 z1 z2 z3];

    % Equation 6.223-6.225
    eq1 = z0*z1 == R(3,2) - R(2,3);
    eq2 = z0*z2 == R(1,3) - R(3,1);
    eq3 = z0*z3 == R(2,1) - R(1,2);
    eq4 = z2*z3 == R(3,2) + R(2,3);
    eq5 = z3*z1 == R(1,3) + R(3,1);
    eq6 = z1*z2 == R(2,1) + R(1,2);

    eq = [eq1 eq2 eq3 eq4 eq5 eq6];

    % Find zi
    Ri = [trace(R) R(1,1) R(2,2) R(3,3)];
    [rii,i] = max(Ri);
    zi = sqrt(1 + 2*rii - trace(R));

    % Substitute zi into the equations and solve for remaining zj
    eq = subs(eq, z(i), zi);
    sol = solve(eq, z);

    % Make array with solutions and insert zi
    solz = [sol.z0 sol.z1 sol.z2 sol.z3];
    solz(i) = zi;

    % Convert to euler parameters
    eta = solz(1)/2;
    epsilon = [solz(2:end)]/2;

    % Convert to angle axis
    theta = acos(eta)*2;
    k = epsilon/sin(theta/2);
end
```