MEGN544 - Project Part 1 Report

Student: Thong Quoc (Bill) Huynh

September 26, 2023

1 Section 1 - Scaling Factor

With the given example points (start of C letter and end of M letter) in both the world frame and the unscaled letter frame, I calculated a vector in each frame and then calculated the ratio of the vector norms to find the scaling factor:

$$\vec{v}_{unscaled\ letter} = \begin{bmatrix}10\\0\\0\end{bmatrix} - \begin{bmatrix}2.5\\4\\0\end{bmatrix} = \begin{bmatrix}7.5\\-4\\0\end{bmatrix}$$

$$|\vec{v}_{unscaled\ letter}| = 8.5$$

$$\vec{v}_{world} = \begin{bmatrix}0.14\\-0.3\\0.4\end{bmatrix} - \begin{bmatrix}-0.01\\-0.3\\0.48\end{bmatrix} = \begin{bmatrix}0.15\\0\\-0.08\end{bmatrix}$$

$$|\vec{v}_{world}| = 0.17$$

$$scalingfactor = \frac{|\vec{v}_{world}|}{|\vec{v}_{unscaled\ letter}|} = 0.02$$

I multiplied this scaling factor to the 2-dimensional letter points' coordinates to get scaled coordinates (X and Y) and added a third Z-axis coordinate of 0 to make the points 3-dimensional. These scaled coordinates could then be considered as coordinates in the scaled letter frame and ready for projection on to the world frame.

2 Section 2 - Transformation Matrix

I called the "scaled letter frame" "frame 1" and "world frame" "frame 2". I could then begin calculating the transformation matrix as follows:

$${}^1T_2 = \begin{bmatrix}{}^1x_2 & {}^1y_2 & {}^1z_2 & {}^1d_{12}\\0 & 0 & 0 & 1\end{bmatrix}$$

By inspection of the world frame relative to the letter frame:

$${}^1x_2 = {}^1x_1 = \begin{bmatrix}1\\0\\0\end{bmatrix}$$

$${}^1y_2 = -{}^1z_1 = \begin{bmatrix}0\\0\\-1\end{bmatrix}$$

$${}^1z_2 = {}^1y_1 = \begin{bmatrix}0\\1\\0\end{bmatrix}$$

Therefore:

$$ {}^{1}T_{2} = \begin{bmatrix} 1 & 0 & 0 & \\ 0 & 0 & 1 & {}^{1}d_{12} \\ 0 & -1 & 0 & \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Also:

$$ \begin{aligned} {}^{1}d_{12} &= {}^{1}d_{1C} + {}^{1}d_{C2} \\ &= {}^{1}d_{1C} - {}^{1}R_{2}\ {}^{2}d_{2C} \\ &= 0.02 \times \begin{bmatrix} 2.5 \\ 4 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} -0.01 \\ -0.3 \\ 0.48 \end{bmatrix} \\ &= \begin{bmatrix} 0.06 \\ -0.4 \\ -0.3 \end{bmatrix} \end{aligned} $$

Therefore:

$$ {}^{1}T_{2} = \begin{bmatrix} 1 & 0 & 0 & 0.06 \\ 0 & 0 & 1 & -0.4 \\ 0 & -1 & 0 & -0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

To project positions from the scaled letter frame (1) to the world frame (2), I needed:

$$ {}^{world}T_{scaled\ letter} = {}^{2}T_{1} = {}^{1}T_{2}^{-1} = \begin{bmatrix} 1 & 0 & 0 & -0.06 \\ 0 & 0 & -1 & -0.3 \\ 0 & 1 & 0 & 0.4 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

I multiplied ${}^{2}T_{1}$ with the 3D scaled letter frame positions to project all the points to the world frame and double checked to verify that the world positions of the C start point and M end point matched what was given.

3 Section 3 - CSM Plot

With the world positions calculated, I used MATLAB plot3 to plot solid black lines with square markers.

At each world frame point, I plotted the column vectors of ${}^{world}R_{scaled\ letter}$ with another rotation around Z-axis appended to direct each local X-axis towards the next point. The angle of Z-axis rotation was found through the angle between the previous vector and the next vector at each point (cross product for sine, dot product for cosine, and then atan2).

Although MATLAB code is not needed, it is attached at the end for future reference.

Please see Figure 1 for the CSM plot as required.

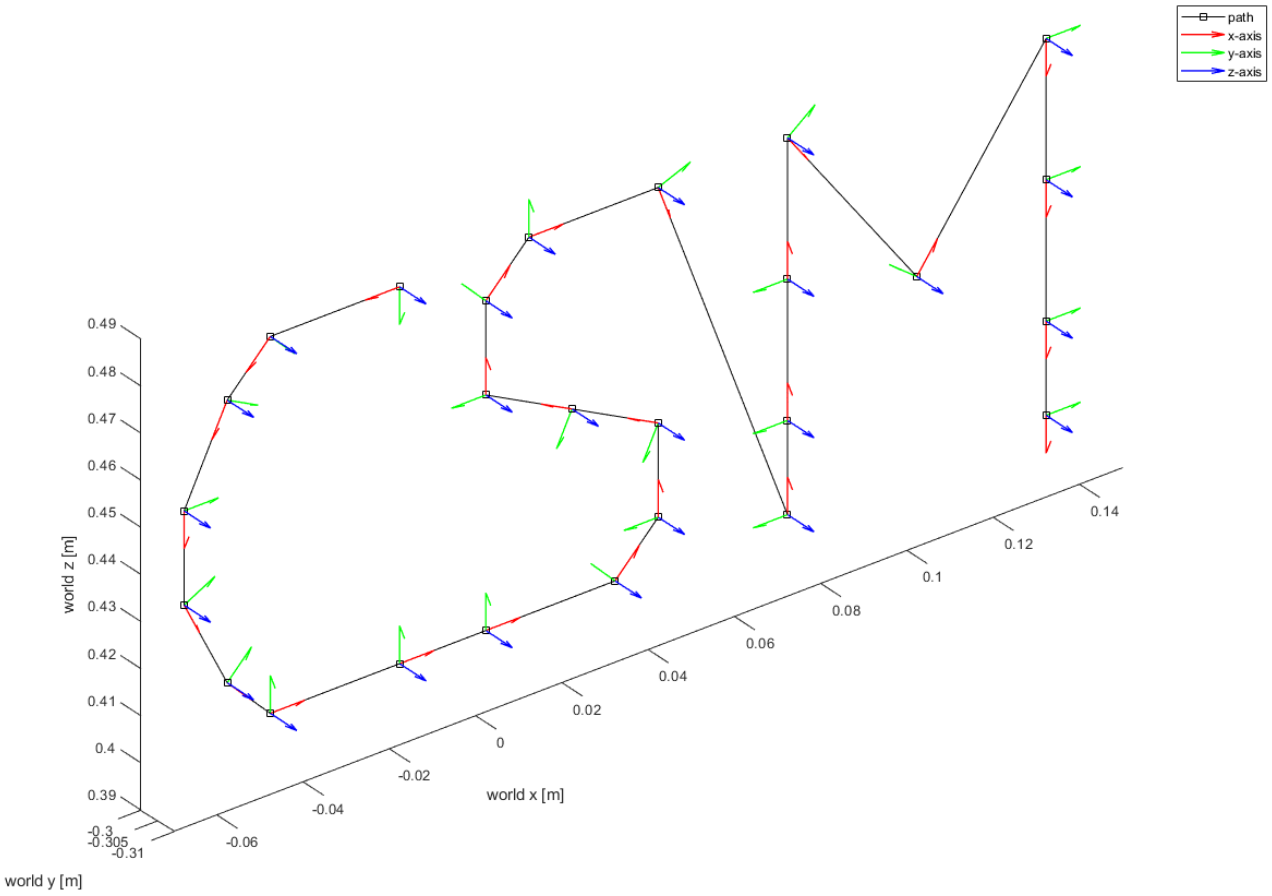
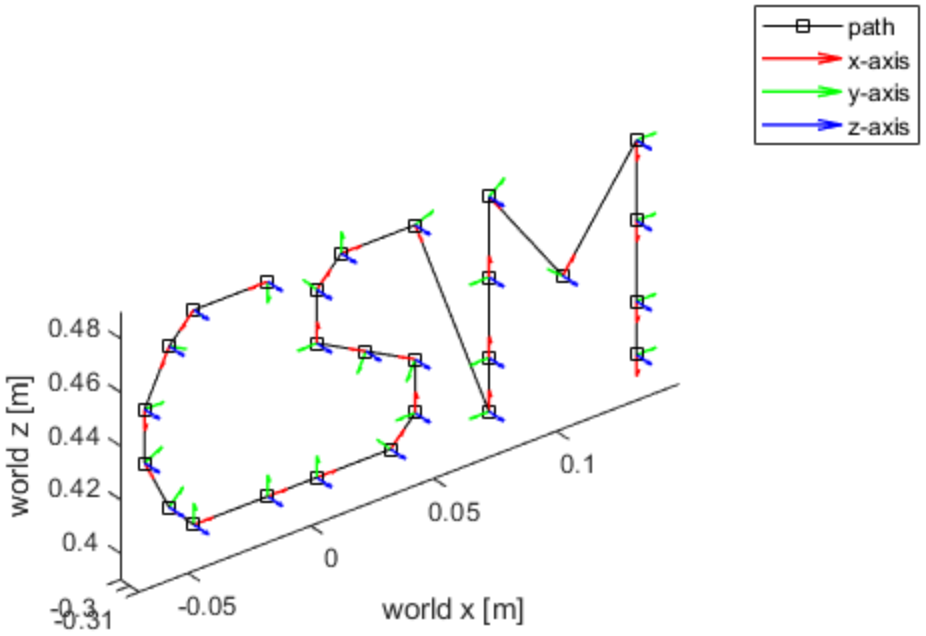


Figure 1: CSM world frame plot with local coordinate frames drawn.

```
% Project Part 1 - plotting CSM
% MEGN-544
% Name: Thong Quoc (Bill) Huynh
% CWID: 10921057

clc; clear; close all;


% Load 2D points
load('points2D.mat');

% figure(1)
% plot(points_all(:,1),points_all(:,2))
% axis([-1 max(points_all(:,1))+1 -1 max(points_all(:,2))+1])

% figure(2)
% plot(points_C(:,1),points_C(:,2),'r',...
%      points_S(:,1),points_S(:,2),'g',...
%      points_M(:,1),points_M(:,2),'b')
% axis([-1 max(points_all(:,1))+1 -1 max(points_all(:,2))+1])

save('points2D','points_C','points_S','points_M','points_all')


% Find scale
C_start_letter = [2.5; 4; 0];
C_start_world = [-0.01; -0.3; 0.48];

M_end_letter = [10; 0; 0];
M_end_world = [0.14;-0.3; 0.4];

vec_letter = M_end_letter - C_start_letter;
vec_world = M_end_world - C_start_world;

scale = norm(vec_world) / norm(vec_letter)

C_start_letter_scaled = scale.*C_start_letter;
M_end_letter_scaled = scale.*M_end_letter;

temp = size(points_all);
length = temp(1);
points_all_scaled_3D = [scale.*points_all zeros(length,1)];


% Transformation matrix (from letter (1) frame to world (2) frame)
x12 = [1;
       0;
       0]; % x of world = x of letter frame
y12 = [0;
       0;
       -1]; % y of world = -z of letter frame
z12 = [0;
```

```
       1;
       0]; % y of world = -z of letter frame
R12 = [x12 y12 z12];
R21 = R12';
d_1_1C = C_start_letter_scaled;
d_2_2C = C_start_world;
d_1_12 = d_1_1C - R12*d_2_2C;
T12 =  [R12 d_1_12;  % Brings frame 1 (letter) to frame 2 (world)
        0 0 0 1];
T21 = inv(T12)       % Translates from frame 1 (letter) to frame 2 (world)


% Transform points to world frame
points_all_scaled_3D_augmented = [points_all_scaled_3D ones(length,1)]';
points_all_world_3D_augmented = T21*points_all_scaled_3D_augmented;
points_all_world_3D = points_all_world_3D_augmented(1:3,:)';
points_all_world_X = points_all_world_3D(:,1);
points_all_world_Y = points_all_world_3D(:,2);
points_all_world_Z = points_all_world_3D(:,3);

% Plot path
plot3(points_all_world_X, points_all_world_Y, points_all_world_Z, ...
    '-sk')
axis equal
xlim([min(points_all_world_X)-0.02 max(points_all_world_X)]+0.01)
ylim([-0.31 -0.3])
zlim([min(points_all_world_Z)-0.02 max(points_all_world_Z)]+0.01)
xlabel('world x [m]')
ylabel('world y [m]')
zlabel('world z [m]')
hold on

% Plot local coordinates (quiver3)

% quiver3 for first point's local coordinate, separately
p = points_all_world_3D(1,:); % point in current iteration
rad = pi;
R = R21*rotZ(rad);
v = R(:,1);
quiver3(p(1),p(2),p(3),v(1),v(2),v(3),0.008,'color','r','LineWidth',1,'MaxHeadSize',10)
v = R(:,2);
quiver3(p(1),p(2),p(3),v(1),v(2),v(3),0.008,'color','g','LineWidth',1,'MaxHeadSize',10)
v = R(:,3);
quiver3(p(1),p(2),p(3),v(1),v(2),v(3),0.008,'color','b','LineWidth',1,'MaxHeadSize',10)

for i = 2:(length-1)
    p_prev = points_all_world_3D(i-1,:); % point in previous iteration
    p = points_all_world_3D(i,:); % point in current iteration
    p_next = points_all_world_3D(i+1,:); % point in next iteration

    vec_prev = p - p_prev;
    vec_next = p_next - p;
    plane_normal = [0;1;0]; % CSM board's plane normal is the world y-axis
```

```
    sine = (cpMap(vec_next')*vec_prev')'*plane_normal/
(norm(vec_next')*norm(vec_prev'));
    cosine = vec_next*vec_prev'/(norm(vec_next')*norm(vec_prev'));
    rad = atan2(sine, cosine);
    R = R*rotZ(rad);

    v = R(:,1);

 quiver3(p(1),p(2),p(3),v(1),v(2),v(3),0.008,'color','r','LineWidth',1,'MaxHeadSize',10)
    v = R(:,2);

 quiver3(p(1),p(2),p(3),v(1),v(2),v(3),0.008,'color','g','LineWidth',1,'MaxHeadSize',10)
    v = R(:,3);

 quiver3(p(1),p(2),p(3),v(1),v(2),v(3),0.008,'color','b','LineWidth',1,'MaxHeadSize',10)
end

% quiver3 for last point's local coordinate, separately
p = points_all_world_3D(length,:); % point in current iteration
v = R(:,1);
quiver3(p(1),p(2),p(3),v(1),v(2),v(3),0.008,'color','r','LineWidth',1,'MaxHeadSize',10)
v = R(:,2);
quiver3(p(1),p(2),p(3),v(1),v(2),v(3),0.008,'color','g','LineWidth',1,'MaxHeadSize',10)
v = R(:,3);
quiver3(p(1),p(2),p(3),v(1),v(2),v(3),0.008,'color','b','LineWidth',1,'MaxHeadSize',10)

legend('path','x-axis','y-axis','z-axis')
hold off


scale =

    0.0200


T21 =

    1.0000         0         0   -0.0600
         0         0   -1.0000   -0.3000
         0    1.0000         0    0.4000
         0         0         0    1.0000
```



Published with MATLAB® R2023a