DEPARTMENT OF MECHATRONICS

MCTE 4352

ROBOTICS

SEM 2, 20/21

SECTION 1

MINI PROJECT

NAME : EIMAN SALLEH BIN MOHD AZIAN

MATRIC ID : 1728753

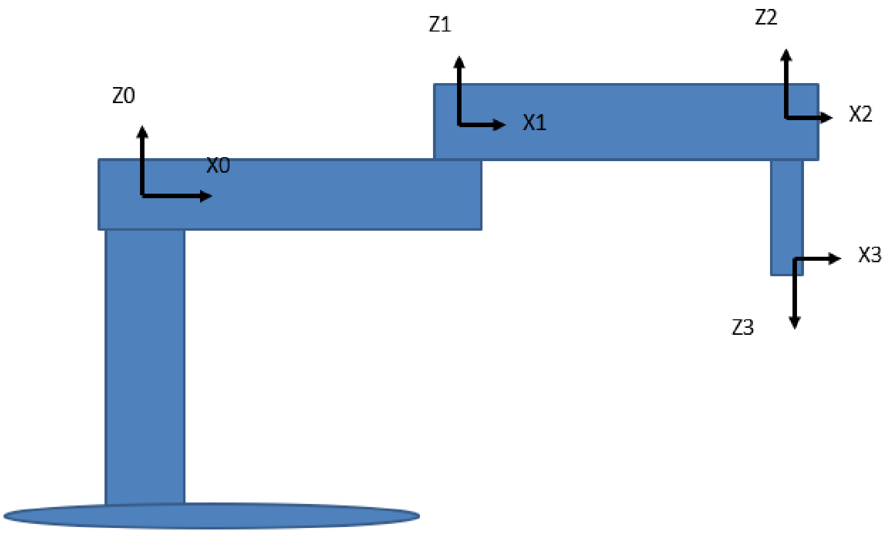
INSTRUCTOR : ASSOC. PROF. DR. TANVEER SALEH

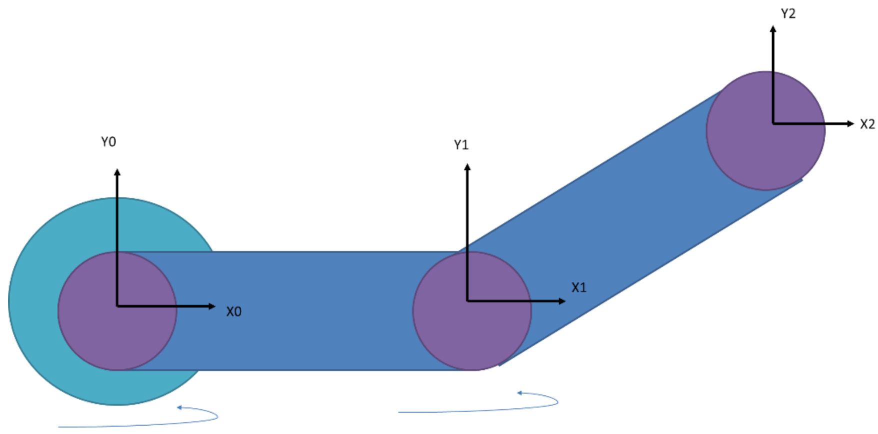
1.0 INTRODUCTION

Selective Compliance Assembly Robot Arm (SCARA) is type of industrial robot. The arm of the robot slightly compliant in the X-Y direction but rigid in the Z-direction. This kind of mechanism is advantage for many types of assembly operations, such as solder, inserting pin in a hole without binding, and more. The main feature for this Scara robot is the jointed 2-link arm layout similar to our human arms that enables arm to extend further and retract or fold up. Due to that feature, the robot is useful for moving an object, drawing and more.

In this project, student has been given a task to model own Scara robot of suitable size to write the student's name in Brush Script MT font using MATLAB.

2.0 MODELLING SCARA ROBOT





$$P = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i C\alpha_i & \alpha_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & \alpha_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

	a_i	α_i	d_i	θ_i
$0-1$	a_1	0	0	$\theta_1 *$
$1-2$	a_2	0	0	$\theta_2 *$
$2-3$	0	0	$d_3 *$	0
$3-H$	0	0	0	0

$$P_1 = \begin{bmatrix} C_1 & -S_1 & 0 & a_1 C_1 \\ S_1 & C_1 & 0 & a_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad P_2 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2 C_2 \\ S_2 & C_2 & 0 & a_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P_3 = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad P_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P_1P_2 = \begin{bmatrix} C_{12} - S_{12} & -C_1S_2 - S_1C_2 & 0 & a_2C_{12} + a_1C_1 - a_2S_{12} \\ S_1C_2 + C_1S_2 & C_{12} - S_{12} & 0 & a_2S_1C_2 + a_2C_1S_2 - a_1S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$P_3P_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

<u>End effector Transformation Matrix</u>

$$P_1P_2P_3P_4 = \begin{bmatrix} C_{12} - S_{12} & -C_1S_2 - S_1C_2 & 0 & a_2C_{12} + a_1C_1 - a_2S_{12} \\ S_1C_2 + C_1S_2 & C_{12} - S_{12} & 0 & a_2S_1C_2 + a_2C_1S_2 - a_1S_1 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

<u>Forward kinematic and inverse kinematic solution</u>

To get theta 1 and theta 2, we must use this solution:

$$x = cos\theta_1 a_1 + \cos(\theta_1 + \theta_2)\, a_2 - \sin(\theta_1 + \theta_2)\, a_2$$

$$y = sin\theta_1 cos\theta_2 a_2 + cos\theta_1 sin\theta_2 a_2 - sin\theta_1 a_1$$

$$z = 0$$

$$(x)^2 + (y)^2 = (cos\theta_1 a_1 + \cos(\theta_1 + \theta_2)\, a_2 - \sin(\theta_1 + \theta_2)\, a_2)^2 + (sin\theta_1 cos\theta_2 a_2 + cos\theta_1 sin\theta_2 a_2 - sin\theta_1 a_1)^2$$

.

.

.

$$\theta_2 = cos^{-1} \left| \frac{x^2 + y^2 + a_1 + a_2}{2a_1a_2} \right|$$

$$\theta_1 = tan^{-1} \left| \frac{a_1y + a_2ycos\theta_2 - a_2xsin(\theta_2)}{a_1x + a_2xcos\theta_2 - a_2ysin(\theta_2)} \right|$$

By using excel, this is what the student got from the calculation above. The student only choose from point 1 to point 10.

Excel

a1	a2
90	90

	x	y	num theta2	deno theta2	theta2_rad	theta2_deg	num theta1	deno theta1	theta1_rad	theta1_deg
point 1	32	41	2885	16200	1.391754885	79.74168103	1513.176118	-238.1259115	-1.41470815	-81.05680624
point 2	32	41	2885	16200	1.391754885	79.74168103	1513.176118	-238.1259115	-1.41470815	-81.05680624
point 3	37	44	3485	16200	1.353978082	77.57722966	1559.85464	179.077139	1.456493007	83.45090222
point 4	38	45	3649	16200	1.343599872	76.98260201	1630.138142	244.4225069	1.421965303	81.47261048
point 5	37	50	4049	16200	1.318179824	75.52614055	2400.410519	-194.8835325	-1.48978642	-85.35847427
point 6	26	50	3356	16200	1.362124733	78.04399838	3142.98391	-1577.625815	-1.105588654	-63.34556374
point 7	17	46	2585	16200	1.410543396	80.81818344	3290.215069	-2312.815106	-0.958101697	-54.89518357
point 8	15	44	2341	16200	1.425782438	81.6913162	3196.414181	-2373.352106	-0.932109583	-53.40594517
point 9	12	39	1845	16200	1.456659787	83.46045798	2836.777027	-2284.162161	-0.892894974	-51.15911359
point 10	15	34	1561	16200	1.474288564	84.47051248	2011.137458	-1565.677688	-0.909301093	-52.09911495

Matlab

	x	y	theta1_rad	theta1_deg	theta2_rad	theta2_deg
point 1	32	41	-0.6435	-36.86983412	2.2143	126.8700446
point 2	32	41	-0.6435	-36.86983412	2.2143	126.8700446
point 3	37	44	1.5395	88.20685256	-2.1414	-122.6931822
point 4	38	45	1.536	88.00631733	-2.1243	-121.7134244
point 5	37	50	-0.5235	-29.99434058	2.1048	120.5961567
point 6	26	50	1.7094	97.9414055	-2.222	-127.3112221
point 7	17	46	-0.5465	-31.3121435	2.3402	134.0835832
point 8	15	44	-0.5701	-32.6643239	2.3737	136.0029918
point 9	12	39	1.7975	102.9891637	-2.4368	-139.6183555
point 10	15	34	-0.7329	-41.99207681	2.4362	139.583978

By comparing between both methods, we can see that the value is different, even though it should be the same. Maybe it might have a problem at D-H parameters. Theta3 has no rotation as it is prismatic joint. Theta4 is not affected by the rotation as it is not affected by other joints.

3.0 METHODOLOGY

1. Installation of the Robotics Toolbox

   The student install the Robotics Toolbox from https://petercorke.com/toolboxes/robotics-toolbox/. Choosing the 1st edition which is RTB9+MVTB3 (2011). After the installation, the student try to understand the functions and command of the RTB toolbox.

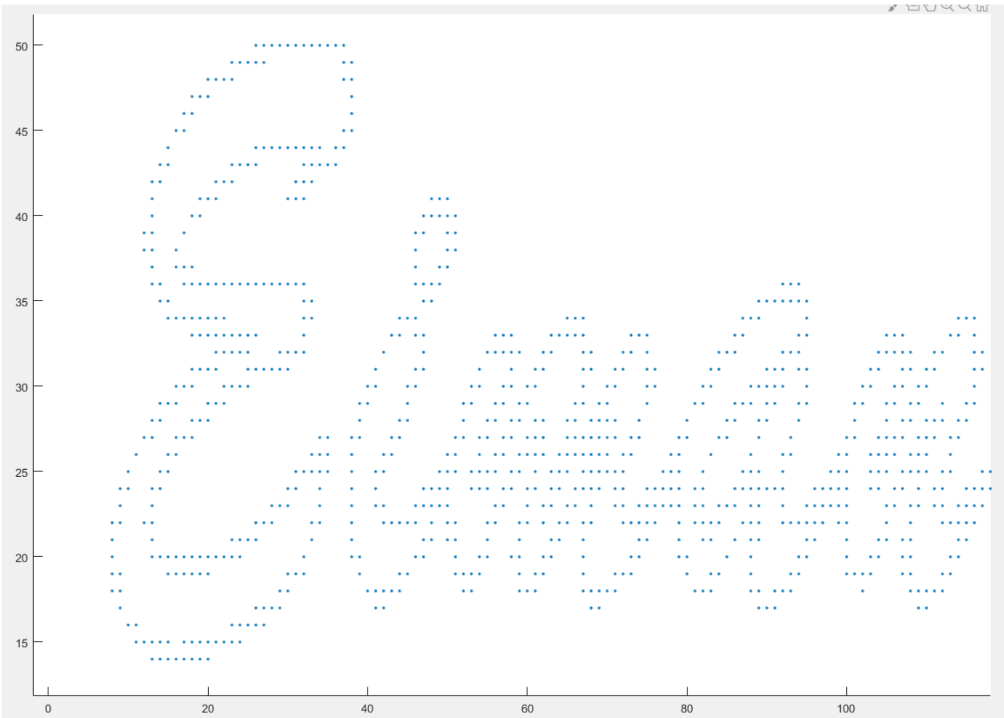
2. Typing the name with font given

   The student typed his name on Adobe Illustrator with font of Brush Script MT. Then it will display the name with the outline. The student choose the outline to export to image which the file name is name_1728753.png.

3. Exporting image to x-y coordinates

   After exporting the name to image, the student then exporting to MATLAB by using the code given. The student got the code through a website https://www.mathworks.com/matlabcentral/fileexchange/54799-convert-image-pixels-to-xy-coordinates.

```
Im = imread('./Images/name_1728753.png');
CoordinateMatrix = pic2points(Im);
scatter(CoordinateMatrix (:,1), CoordinateMatrix (:,2),'.');
```



4. <u>Choosing coordinates for each letter</u>

   Finally, the student managed to get the x-y coordinate. Then, he choose the suitable points or coordinates manually and then he exported to Microsoft Excel. The points can be referred to appendix below.

5. <u>Simulation using ONE LETTER</u>

   At first, there were several mistakes. Then student decided using one letter for easier simulation. So, he started with letter E. After several time try and error, he managed to make a trajectory path of letter E.

6. <u>Add more points after the First Letter</u>

   After achieve the first letter of the student's name, he add more point (other letters), to simulate the full name.

4.0 MATLAB CODING & SIMULATION

There are several phases that we must have inside the MATLAB code of "Eiman_1728753_robotics.m"

Phase 1: Modelling the Scara Robot

```
%% Set the length of each link
a1 = 90;
a2 = 90;
d1 = 90; % Height of the Robot
d3 = 9; % Length of prismatic joint

%% Scara Robot Model
%% L = Link([Theta d a alpha 0/1])
L1 = Link([0 d1 a1 0 0], 'standard');
L2 = Link([0 0 a2 pi 0], 'standard');
L3 = Link([0 d3 0 0 1], 'standard'); % Prismatic joint
L4 = Link([0 0 0 0 0], 'standard'); % End effector

%Set how much go up & down
L3.qlim = [0 10];
Rob = SerialLink([L1 L2 L3 L4],'name','Robot');
```

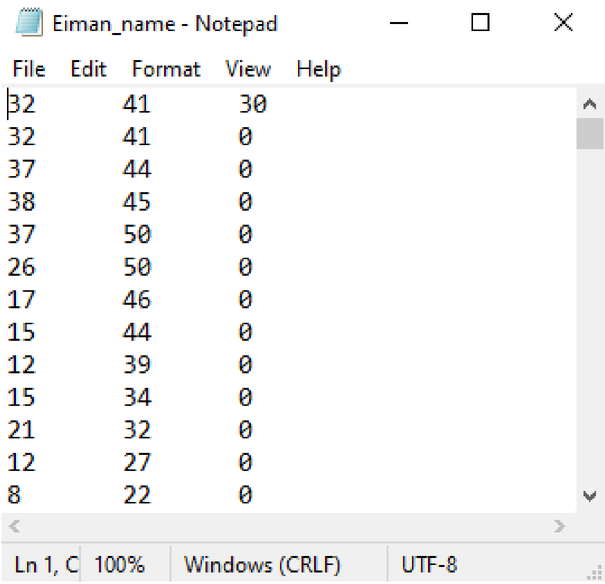
Firstly, the student set the length for each link by referring the DH parameters. To model a Scara robot, student must initialize the link. The Link function will be Link([Theta d a alpha 0or1]). 0 represent for revolute while 1 represent for prismatic. L1, L2, and L4 are specified as revolute joint, while L3 are specified as prismatic joint.

Then, the L3, the student must set at how much the prismatic joint moves up and down. So, it was set from 0 to 10 using L3.qlim.

After that, the student set the robot using SerialLink function and named the robot as 'Robot'.

Phase 2: Points of name

```
%% Points of Name (x,y,z)
name = dlmread('./nama/Eiman_name.txt');
```


Eiman_name - Notepad

File Edit Format View Help

32	41	30
32	41	0
37	44	0
38	45	0
37	50	0
26	50	0
17	46	0
15	44	0
12	39	0
15	34	0
21	32	0
12	27	0
8	22	0

Ln 1, C 100% Windows (CRLF) UTF-8

All the coordinates that the robot need to get was put inside a text file,

‘Eiman_name.txt’ from notepad.

The first column is x-coordinate, second is y-coordinate, and third is z-coordinate.

Each row represents each point. So, there are 157 points in this project.

The dlmread function will read the text file and store inside ‘name’ matrix.

Phase 3: Transformation Matrix

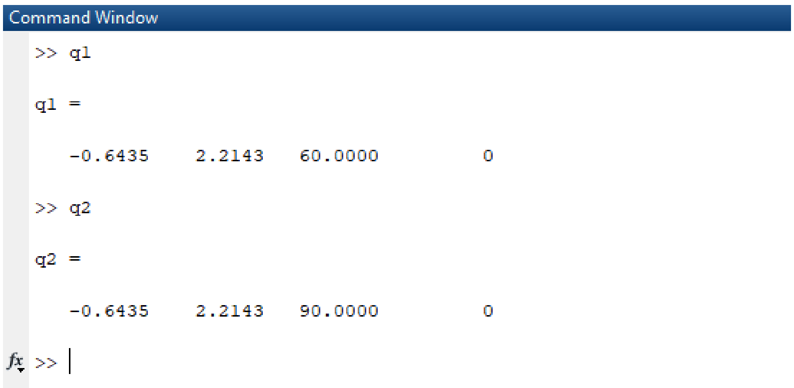
```
%% Transformation matrix
P1 = transl(name(1,1),name(1,2),name(1,3));
P2 = transl(name(2,1),name(2,2),name(2,3));
P3 = transl(name(3,1),name(3,2),name(3,3));
P4 = transl(name(4,1),name(4,2),name(4,3));
P5 = transl(name(5,1),name(5,2),name(5,3));
.
.
P157 = transl(name(157,1),name(157,2),name(157,3));
```

In this phase, it keep all the transformation matrix of each points from the phase 2 in terms of translation matrix.

Phase 4: Inverse Kinematic and Forward Kinematic

```
%% Inverse kinematics
q0 = [0 0 0 0]; %initial value
q1 = Rob.ikine(P1,q0,'mask',[1,1,1,0,0,0]);
q2 = Rob.ikine(P2,q1,'mask',[1,1,1,0,0,0]);
q3 = Rob.ikine(P3,q2,'mask',[1,1,1,0,0,0]);
q4 = Rob.ikine(P4,q3,'mask',[1,1,1,0,0,0]);
q5 = Rob.ikine(P5,q4,'mask',[1,1,1,0,0,0]);
.
.
q157 = Rob.ikine(P157,q156,[1,1,1,0,0,0]);


%% Forward kinematics
r1 = Rob.fkine(q1);
r2 = Rob.fkine(q2);
r3 = Rob.fkine(q3);
r4 = Rob.fkine(q4);
r5 = Rob.fkine(q5);
.
.
r157 = Rob.fkine(q157);
```

For inverse kinematic, the student use 'ikine' functions from RTB toolbox. It will calculate each inverse kinematics of each points, P by getting each configuration of the thetas through q.



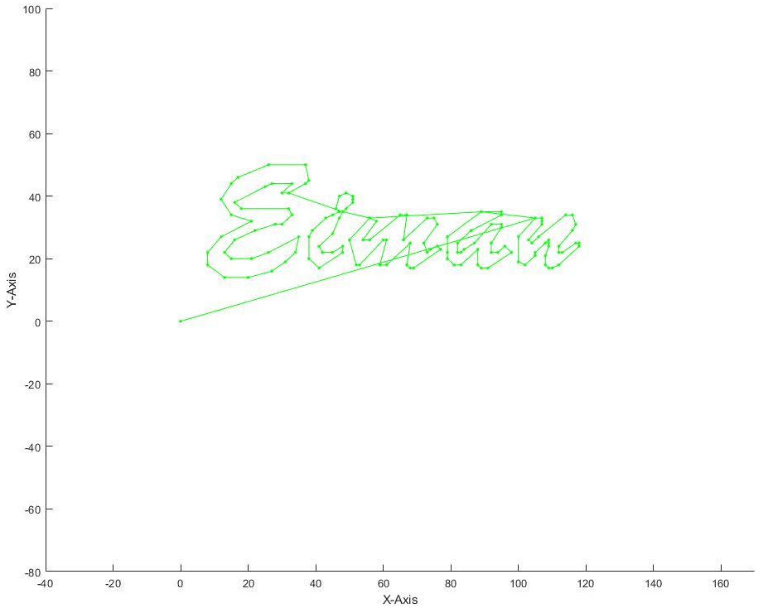
For forward kinematic, the student use ‘fkine’ function from the RTB toolbox, It will calculate each forward kinematics and get the configuration of x,y,z coordinates of the frame.

```
Command Window
r1 =
         0         1         0        32
         1         0         0        41
         0         0        -1        30
         0         0         0         1
>> r2

r2 =
         0         1         0        32
         1         0         0        41
         0         0        -1 2.348e-11
         0         0         0         1
fx >>
```

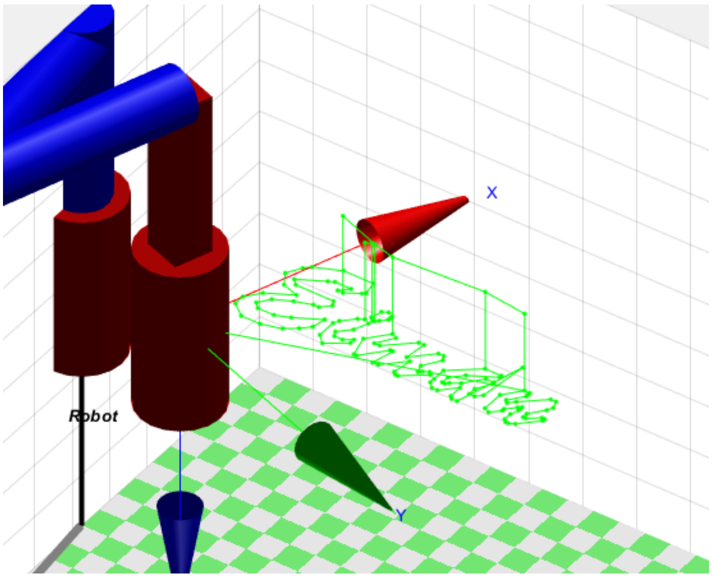
Phase 5: Trajectory path and Robot simulation

```
%% Trajectory path
trajectorypath = dlmread('Eiman_name.txt');
[nx,ny] = size(trajectorypath);

figure
hold on

for i = 1:nx-1
    v=[trajectorypath(i,:);trajectorypath(i+1,:)];
    plot3(v(:,1),v(:,2),v(:,3),'g');
    plot3(v(:,1),v(:,2),v(:,3),'g.')
end




%% Animation of Scara Robot
% Time settings to either speed up/slow down the animation
t = (0: .05: 0.1)';

% All transformation animation
line1 = jtraj(q0,q1,t); Rob.plot(line1, 'workspace', axis_matrix);
line2 = jtraj(q1,q2,t); Rob.plot(line2, 'workspace', axis_matrix);
line3 = jtraj(q2,q3,t); Rob.plot(line3, 'workspace', axis_matrix);
.
.
line157 = jtraj(q156,q157,t); Rob.plot(line157, 'workspace',
axis_matrix);
```

For the trajectory path, is for plotting the path line of the end effector. ‘trajectorypath’ is read from ‘Eiman_name.txt’ just like phase 2 where the name matrix read from the same text file, to get the ‘trajectory matrix’. The trajectory path line can be referred to the figure below.



Finally, for transformation animation, is where the student code for the robot animations using the configuration of theta that the student received from the inverse kinematic. By using the ‘jtraj’ function, the student able to get the animation from one point to another. The animation can be refer to the video link below.



Appendix

Link of explanation: https://youtu.be/F5qYIERyIkQ

Link of Video Result in 2D: https://drive.google.com/file/d/1unD-12-K2KxQpntWfhAflbfhosm3wHHl/view?usp=sharing

Link of Video Result in 3D:

https://drive.google.com/file/d/1KUbTgACWatfVLgnoayWYZv3KmdaOZ0Z1/view?usp=sharing

E	X	Y	Z
	32	41	30
	32	41	0
	37	44	0
	38	45	0
	37	50	0
	26	50	0
	17	46	0
	15	44	0
	12	39	0
	15	34	0
	21	32	0

12	27	0
8	22	0
8	18	0
13	14	0
20	14	0
27	16	0
31	19	0
34	22	0
35	27	0
26	22	0
21	20	0
15	20	0
13	22	0
16	26	0
22	29	0
28	31	0
30	31	0
33	34	0
32	36	0
18	36	0
16	38	0
25	43	0
27	44	0
33	44	0
30	41	0
32	41	0
32	41	30

i	X	Y	Z
	48	35	30
	48	35	0
	49	36	0
	51	38	0
	51	39	0
	51	40	0
	49	41	0
	47	40	0
	46	36	0
	47	35	0
	47	35	30
	45	34	30
	45	34	0
	43	33	0

39	29	0
38	27	0
38	20	0
41	17	0
48	22	0
48	24	0
45	22	0
42	22	0
41	24	0
45	28	0
47	33	0
48	35	0
48	35	30

m	X	Y	Z
	56	33	30
	56	33	0
	50	26	0
	52	18	0
	53	18	0
	60	26	0
	61	26	0
	59	18	0
	61	18	0
	68	25	0
	67	18	0
	68	17	0
	69	17	0
	77	23	0
	76	24	0
	74	23	0
	73	22	0
	72	25	0
	76	31	0
	75	33	0
	73	33	0
	66	26	0
	67	34	0
	65	34	0
	56	26	0
	54	26	0
	58	32	0
	56	33	0

56	33	30

a	X	Y	Z
	89	35	30
	89	35	0
	79	27	0
	79	22	0
	79	20	0
	81	18	0
	83	18	0
	88	23	0
	88	18	0
	89	17	0
	91	17	0
	98	42	0
	96	24	0
	94	22	0
	92	22	0
	92	25	0
	95	30	0
	95	31	0
	92	31	0
	87	25	0
	84	23	0
	83	22	0
	82	22	0
	82	25	0
	86	29	0
	95	34	0
	95	35	0
	89	35	0
	89	35	30

n	X	Y	Z
	105	33	30
	105	33	0
	100	27	0
	100	19	0
	102	18	0
	105	21	0
	105	22	0

108	25	0
109	26	0
109	25	0
109	24	0
108	21	0
108	18	0
109	17	0
110	17	0
112	18	0
118	24	0
118	25	0
117	25	0
113	22	0
112	22	0
112	24	0
116	29	0
117	31	0
116	34	0
114	34	0
106	27	0
104	25	0
103	26	0
107	31	0
107	32	0
107	33	0
105	33	0