MCTE 4352
ROBOTICS
SEMESTER II || SESSION 2021/2022
SECTION 1

ROBOTICS PROJECT

Prepared by:

SARAH AMANDA BT SHAHRUL AZLAN 1813982

Course Instructor:

DR TANVEER SALEH

1.0 ROBOT DESIGN

In this project, we will design a 3 degree of freedom (DOF) robot program using MATLAB software. We will be discussing the mathematical modelling of the robot that involves the forward kinematics and the inverse kinematics. Next, we will simulate the working space of the robot. The sketch of the robot is shown in Figure 1.

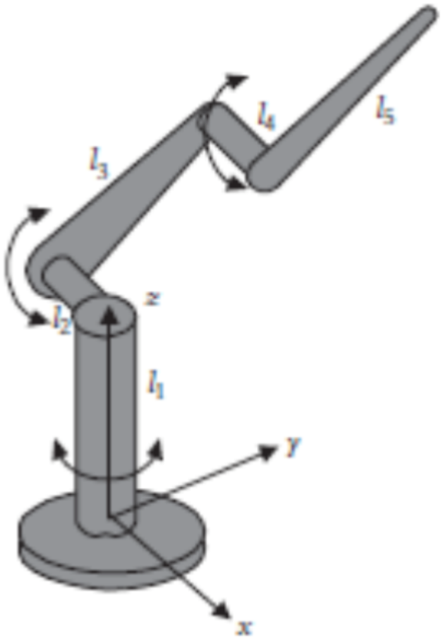


Figure 1: Robot Sketch

The robot has a specification that are as follows

Parameters	Length (mm)
L1	82
L2	0.182
L3	1.1582
L4	0.182
L5	0.7582

Table 1: Robot Specifications

2.0 MATHEMATICAL MODELLING

2.1 FORWARD KINEMATICS

Firstly, to model the system in mathematical form, we apply forward kinematics to find the end effector position of the robot. We are using Denavit-Hartenberg representation to find the related parameters. Figure 2 and Table 2 show the frame assignment and DH parameters of the robot.

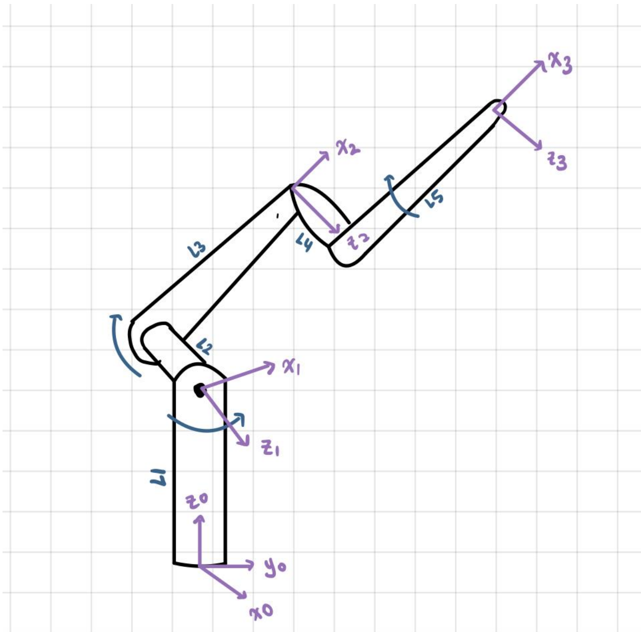


Figure 2: Frame Assignment

Joint	a_i (mm)	α_i	d_i (mm)	θ_i
0 - 1	0	90	L1	θ_1
1 - 2	L3	0	-L2	θ_2
2 -3	L5	0	L4	θ_3

Table 2: DH Parameters

From the parameters assigned in the DH table, the derivation of the position transformation matrix, uTh , could be obtained. The mathematical modelling and calculation was done using MATLAB software.

```
Editor – /Users/sarahamanda/Desktop/DH.m
firstfwd.m | DH.m | DH_Inverse.m | +

function [A] = DH (a, alpha, d, theta)
A= [cos(theta)    -sin(theta*round(cos(alpha)))     sin(theta)*round(sin(alpha))     a*cos(theta)
    sin(theta)     cos(theta*round(cos(alpha)))    -cos(theta)*round(sin(alpha))     a*sin(theta)
        0               round(sin(alpha))                round(cos(alpha))              d
        0                     0                                0                        1       ];
end
```

Figure 3: DH Function in Matlab

```
Editor – /Users/sarahamanda/Desktop/firstfwd.m
firstfwd.m | DH.m | DH_Inverse.m | +

1    clear
2    clc
3    syms t1 t2 t3
4
5    %% DH parameters
6    %Link 1
7    a1 = 0;       alpha1 = 90;  d1 = 82;      theta1 = t1;
8    %Link 2
9    a2 = 1.1582; alpha2 = 0;   d2 = -0.182; theta2 = t2;
10   %Link3
11   a3 = 0.7582; alpha3 = 0;   d3 = 0.182;  theta3 = t3;
12
13
14   %% Forward kinematics
15   A_01 = DH(  a1 , alpha1 , d1 , theta1);
16   A_12 = DH(  a2 , alpha2 , d2 , theta2);
17   A_23 = DH(  a3 , alpha3 , d3 , theta3);
18
19   A_03 = simplify(A_01 * A_12 * A_23);
20
```

Figure 4: Forward Kinematics Coding in Matlab

A DH function is firstly written in a separate script as shown in Figure 3. The DH function contains the formula for Denavit-Hartenberg. In another script, we inserted the DH parameters of the robot following Table 2 and finally the forward kinematics calculation is executed by calling the DH Function. The position transformation matrix, $_{0}T_{3}$, which is denoted as, A_03, in line 19, was attained.

```
Command Window
>> A_03

A_03 =

[cos(t2 + t3)*cos(t1), -sin(t2 + t3)*cos(t1),  sin(t1),   (cos(t1)*(3791*cos(t2 + t3) + 5791*cos(t2)))/5000]
[cos(t2 + t3)*sin(t1), -sin(t2 + t3)*sin(t1), -cos(t1),   (sin(t1)*(3791*cos(t2 + t3) + 5791*cos(t2)))/5000]
[        sin(t2 + t3),          cos(t2 + t3),        0, (3791*sin(t2 + t3))/5000 + (5791*sin(t2))/5000 + 82]
[                   0,                     0,        0,                                                   1]

fx >>
```

Figure 5: Transformation Matrix ${}_0T_3$

Figure 5 shows the transformation matrix output from the matlab software. Below we will simplify the transformation matrix to make it easier to read.

$$ {}_0^3A = \begin{bmatrix} n_x & O_x & q_x & P_x \\ n_y & O_y & q_y & P_y \\ n_z & O_z & q_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$n_x = C_{23} C_1 \qquad O_x = -S_{23} C_1$$

$$n_y = C_{23} S_1 \qquad O_y = -S_{23} S_1$$

$$n_z = S_{23} \qquad O_z = C_{23}$$

$$q_x = S_1 \qquad P_x = 0.7582\ C_{23} C_1 + 1.1582\ C_2 C_1$$

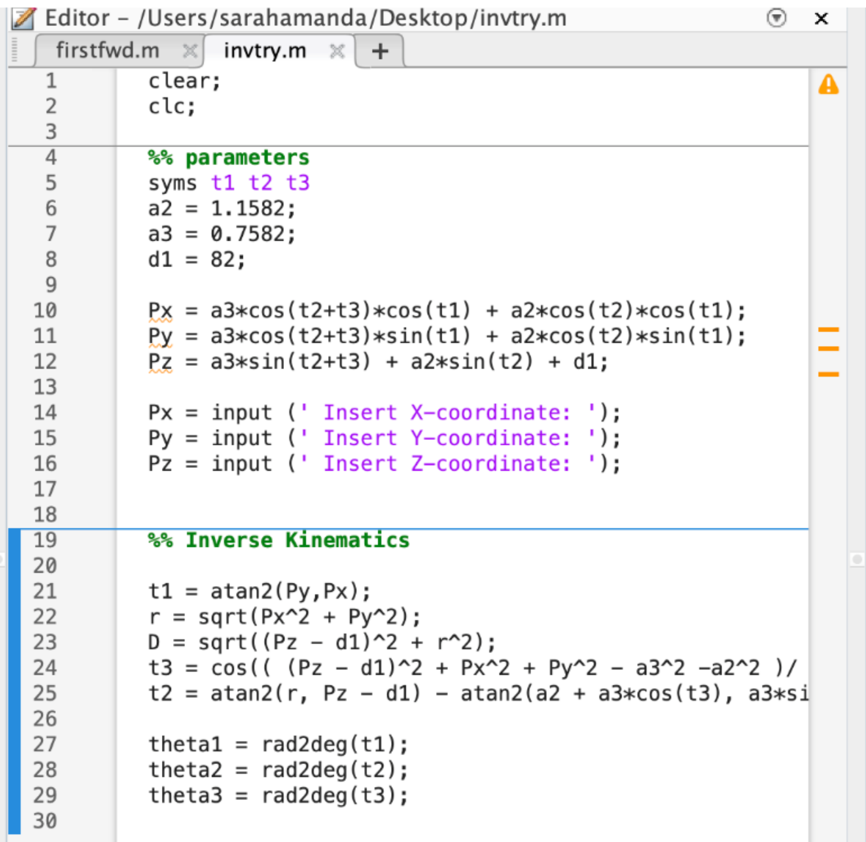
$$q_y = -C_1 \qquad P_y = 0.7582\ C_{23} S_1 + 1.1582\ C_2 S_1$$

$$q_z = 0 \qquad P_z = 0.7582\ S_{23} + 1.1582\ S_2 + 82$$

$$
{}^{3}_{0}A = \begin{bmatrix} C_{23}C_1 & -S_{23}C_1 & S_1 & 0.7582\,C_{23}C_1 + 1.1582\,C_2C_1 \\ C_{23}S_1 & -S_{23}S_1 & -C_1 & 0.7582\,C_{23}S_1 + 1.1582\,C_2S_1 \\ S_{23} & C_{23} & 0 & 0.7582\,S_{23} + 1.1582\,S_2 + 82 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

INVERSE KINEMATICS

The figure below shows the Matlab coding to develop Inverse Kinematics of the robot. The program will prompt the user for X-coordinate, Y-coordinate, and Z-coordinate. It will then calculate using the Px, Py, and Pz formula that we had gotten from Forward Kinematics. The output will show the value of theta in degrees.


Editor – /Users/sarahamanda/Desktop/invtry.m

firstfwd.m | invtry.m

```
clear;
clc;

%% parameters
syms t1 t2 t3
a2 = 1.1582;
a3 = 0.7582;
d1 = 82;

Px = a3*cos(t2+t3)*cos(t1) + a2*cos(t2)*cos(t1);
Py = a3*cos(t2+t3)*sin(t1) + a2*cos(t2)*sin(t1);
Pz = a3*sin(t2+t3) + a2*sin(t2) + d1;

Px = input (' Insert X-coordinate: ');
Py = input (' Insert Y-coordinate: ');
Pz = input (' Insert Z-coordinate: ');


%% Inverse Kinematics

t1 = atan2(Py,Px);
r = sqrt(Px^2 + Py^2);
D = sqrt((Pz - d1)^2 + r^2);
t3 = cos(( (Pz - d1)^2 + Px^2 + Py^2 - a3^2 -a2^2 )/
t2 = atan2(r, Pz - d1) - atan2(a2 + a3*cos(t3), a3*si

theta1 = rad2deg(t1);
theta2 = rad2deg(t2);
theta3 = rad2deg(t3);
```

OUTPUT

```
Command Window
 Insert X-coordinate: 5
 Insert Y-coordinate: 6
 Insert Z-coordinate: 8
>> t1

t1 =

    0.8761

>> theta1

theta1 =

   50.1944

>> t2

t2 =

    1.1342

>> theta2

theta2 =

   64.9872
```

```
t3 =

   -0.8516

>> theta3

theta3 =

  -48.7910

fx >>
```

3.0 ROBOT WORKSPACE

Next we will be simulating the work envelope of the robot. A robot's work envelope otherwise known as its workspace is defined to be its range of motion when the manipulator moves forward, backwards, up and down. The shape created is called the workspace and it is highly dependent and determined by the length of the robot's arm and its axis design. A robot usually performs and is limited to moving in this work envelope. Hence, a work envelope of a robot is considered important when selecting a robot for a particular task or application.

The work envelope of the robot is simulated in Matlab. Figure 6 shows the code where we first initialise the parameters for d, a, theta, and alpha. Then, we will call the function for DHworkspace shown at figure 7.

Editor – /Users/sarahamanda/Desktop/work.m

firstfwd.m | DH.m | DH_Inverse.m | work.m | DHworkspace.m

```
clear
clc
%% Initialise DH paramaters

t1=0;
t2=0;
t3=0;
a2 = 1.1582;
a3 = 0.7582;
d1 = 82 ;
d2 = -0.182;
d3 = 0.182;
n=0;

%% DH Table
for t1= -pi/2 : 0.2 : pi/2
    for t2= 0 : 0.2 : 3*pi/4
        for t3= -3*pi/4 : 0.2 : 0


n=n+1;
%link =[  theta     d       a        alpha     ]
link1= [   t1       d1      0      deg2rad(90)  ];
link2= [   t2       d2      a2          0       ];
link3= [   t3       d3      a3          0       ];


A03= DHworkspace(link1)*DHworkspace(link2)*DHworkspace(link3);
xyz_cor(n,:) = [A03(1:3,4); t1; t2 ;t3];
        end
    end
end

%% Figure
figure (1);
scatter3(xyz_cor(:,1),xyz_cor(:,2),xyz_cor(:,3),0.5,xyz_cor(:,3),'filled','MarkerFaceAlpha', 1);
xlabel('x')
ylabel('y')
zlabel('z')
```

Figure 6: Work Envelope Matlab

Figure 7 shows the DHworkspace function that will be called into the main work envelope simulation.

Editor – /Users/sarahamanda/Desktop/DHworkspace.m

DH.m | DH_Inverse.m | work.m | DHworkspace.m

```
function DHtable = DHworkspace(matrix)

theta = matrix(1);
d_n   = matrix(2);
an_1  = matrix(3);
alpha = matrix(4);

sa = sin(alpha);
ca = cos(alpha);
st = sin(theta);
ct = cos(theta);

DHtable = [ ct    -ca*st     sa*st    an_1*ct ;
            st     ca*ct    -sa*ct    an_1*st ;
             0      sa        ca       d_n    ;
             0       0         0        1    ];
end
```

Figure 7: DHworkspace Function

Next, we will write the scatter plot coding to show the work envelope of robot in X-Y plane, Y-Z plane, X-Z plane and 3D space.

```
%% Figure
figure (1);
scatter3(xyz_cor(:,1),xyz_cor(:,2),xyz_cor(:,3),0.5,xyz_cor(:,3),'filled','MarkerFaceAlpha', 1);
xlabel('x')
ylabel('y')
zlabel('z')

figure (2)
scatter3(xyz_cor(:,1),xyz_cor(:,2),xyz_cor(:,3),0.5,xyz_cor(:,3),'filled','MarkerFaceAlpha', 1);
view(90,90);
xlabel('x')
ylabel('y')
zlabel('z')
title('Workspace in X-Y plane');

figure (3)
scatter3(xyz_cor(:,1),xyz_cor(:,2),xyz_cor(:,3),0.5,xyz_cor(:,3),'filled','MarkerFaceAlpha', 1);
view(90,0);
xlabel('x')
ylabel('y')
zlabel('z')
title('Workspace in Y-Z plane')

figure (4)
scatter3(xyz_cor(:,1),xyz_cor(:,2),xyz_cor(:,3),0.5,xyz_cor(:,3),'filled','MarkerFaceAlpha', 1);
view(0,0);
xlabel('x')
ylabel('y')
zlabel('z')
title('Workspace in X-Z plane')
```

Figure 8: Figure Coding

The scatter plot is shown in Figure 9-12 for the 3D plane, X-Y plane, Y-Z plane, and X-Z plane respectively.

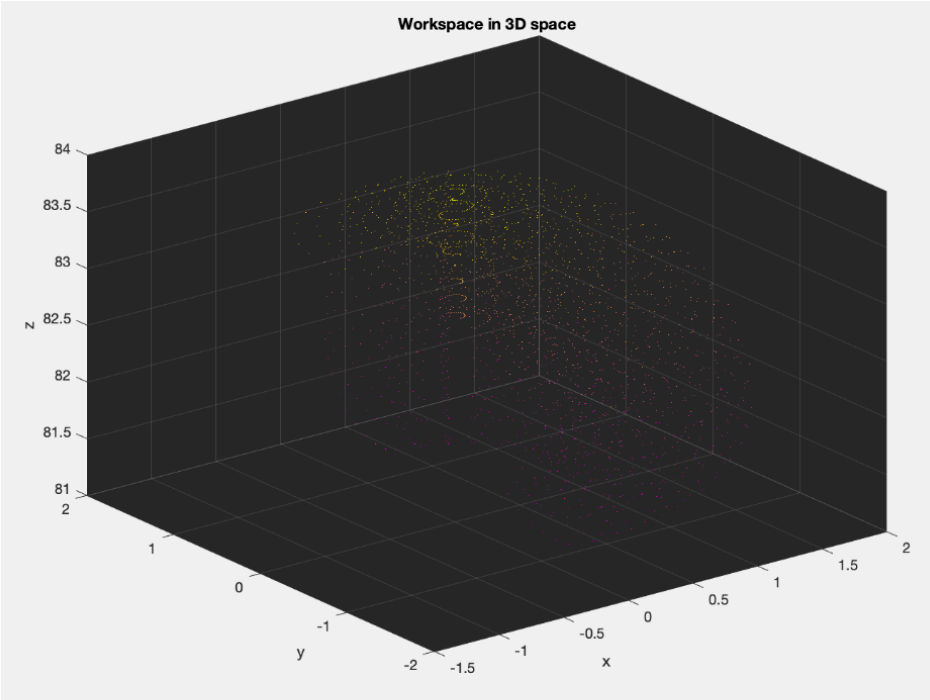


Figure 9: Work Envelope in 3D plane

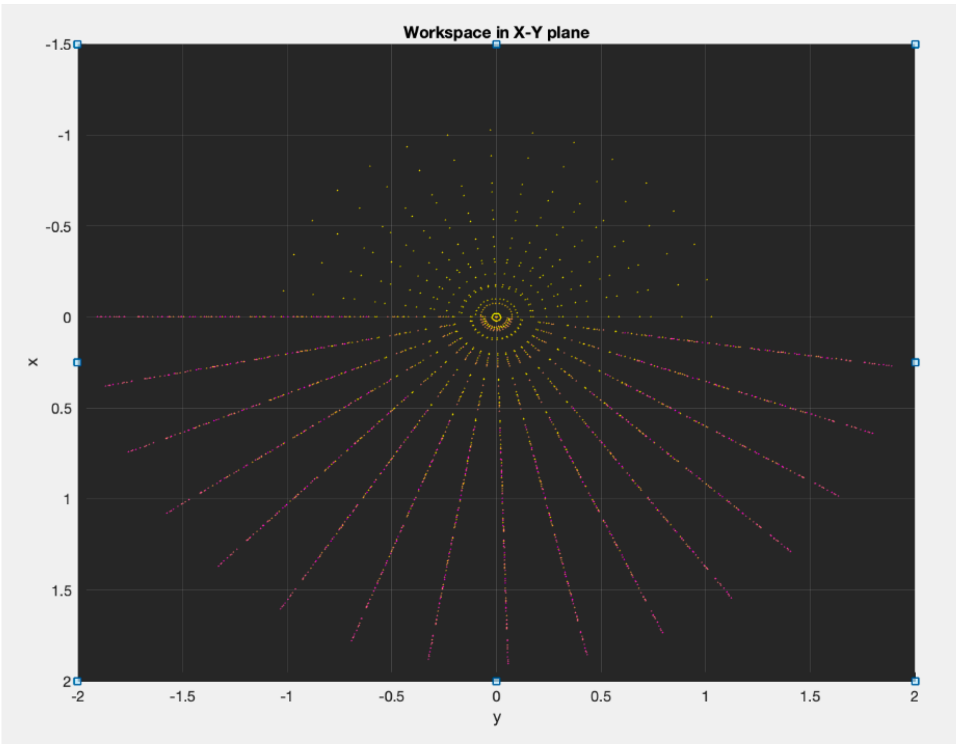


Figure 10: Work Envelope in X-Y plane

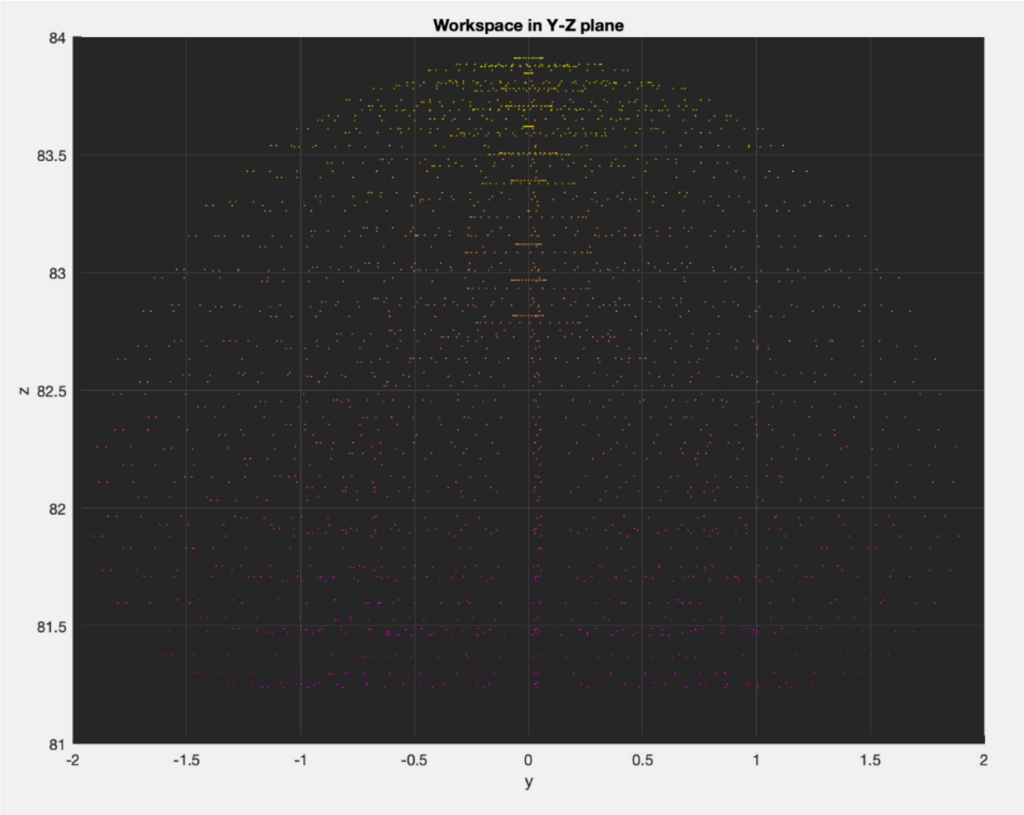


Figure 11: Work Envelope in Y-Z plane

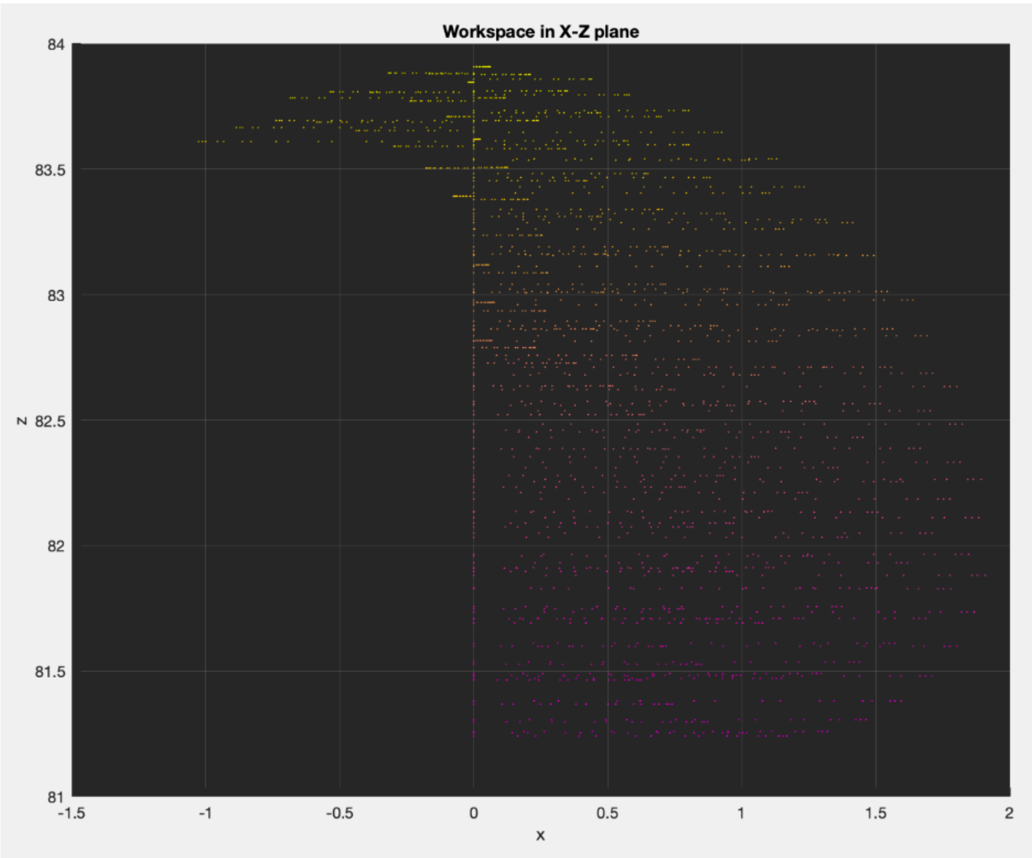


Figure 12: Work Envelope in X-Z plane