A Project Report

On

Terrain Following Robot Manipulator

BY

UTKARSH RASTOGI	**2017A4PS0734H**
SURYA PRATAP SINGH	**2017A4PS1511H**
PRATHMESH MAHALLE	**2017A4PS1635H**

Under the supervision of

PROF ARSHAD JAVED

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI (RAJASTHAN)

HYDERABAD CAMPUS

(JANUARY-APRIL 2020)

Problem Statement

In today's world many countries are preparing strong armies so that they can fight with the terrorists and eliminate them and secure their country and countrymen. Most of these terrorists have started using cowardly tactics to kill the soldiers and countrymen by using Landmines post World War. These landmines have killed thousands of people all over the globe which is why it becomes important to detect them. There have been many instruments in the past to detect the landmines but most of them require direct involvement of humans to carry out the operation and in doing so we have lost many of our brave soldiers and innocent peoples and sometimes the landmine susceptible fields are not favorable for human-beings due to radiation. These days we have started using robots to carry out the above operations but they are fully dependent on the human instructions and many of them face difficulty based on the unpredictability of the landmine terrain and sometimes fail. Hence there is a great need for a semi-autonomous robot which doesn't require the human-being to be walking side-by-side and could autonomously adjust itself according to the terrain over the landmine field and with that we propose a terrain following robot manipulator which could overcome the difficulties faced by the previously developed robots.

CONTENTS

Table of Contents

TERRAIN FOLLOWING MANIPULATOR

Abstract

One of the primary interests of robotics field is to remove humans from hazardous and life-threatening working environment and send robots to work in such environments to save lives also it has always been more economically viable to send robots rather to prepare systems for human safety. Remote detection and detonation of landmines is such a field for which this report presents a design of a fixed manipulator, which can be mounted on a mobile robot. the design is based on an articulated arm with an end effector designed for collision avoidance with the surface being scanned.

Introduction

A ground penetrating radar is a highly useful technology that can be used for a variety of application that involve mapping the ground. One specific military application of it can be considered the detection of landmine. This is a very effective method for detection of landmine. However, it is also dangerous for a human being to carry out detection since a minor mistake can lead to detonation of the mine. The technology needs the equipment should be in very close contact with the ground being scanned to have a proper image that can be analyzed. Hence the task has to be performed meticulously such that the GPR is in close contact however it does not apply any pressure so as to trip a mine. Hence the task is tedious and prone to error by a human. A robot however can be designed to perform this task and is not to prone to error due to repetitiveness. In addition to that the tasks can be performed remotely by a human which reduces the risk to human life. This paper aims to design a robot manipulator that can follow a given terrain. If a GPR is attached at the end point the robot can be used for the above landmine detection task and replace the human being. To design the manipulator multiple design were considered and a 5 DOF as$R \vdash R \perp R \perp R \parallel R$ is due to its lower complexity while not sacrificing on the required dexterity. Various mathematical models of the manipulator are developed for the purpose of simulating and designing the control system of the robotic system. The algorithm has been developed define a closest plane while maintain a specific orientation necessary for effective scanning using GPR.

Forward kinematics

DH Parameter Calculations

A 5 degree of freedom design was chosen to minimize complexity while maintaining maximum functionality for the task of following the terrain.

The design can be described as $R \vdash R \perp R \perp R \parallel R$ (notation used in (1)). The following stick notation in Figure 1 can be used for reference (figure not to scale)

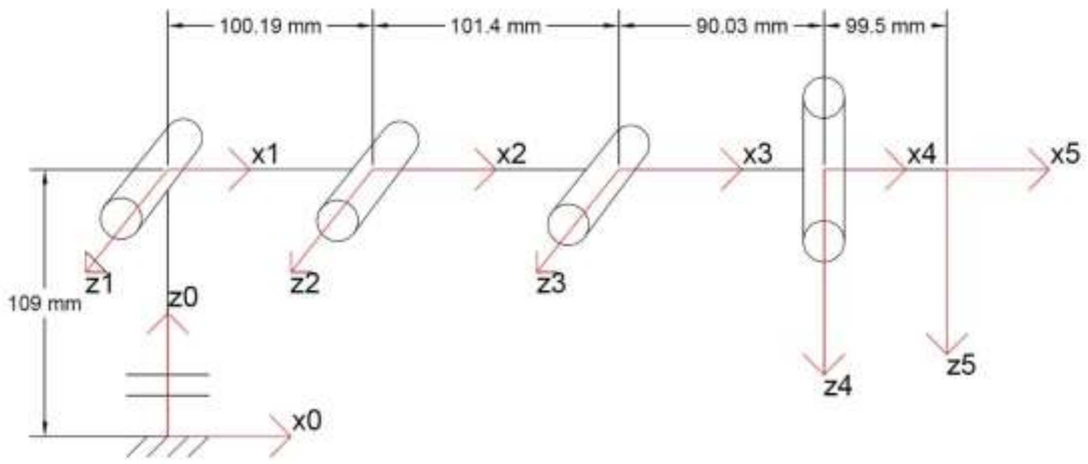


FIGURE 1

DH parameter table for the same is:

TABLE 1

Frame	α_i	a_i	d_i	θ_i
1	90°	0	l_1	θ_1
2	0°	l_2	0	θ_2
3	0°	l_3	0	θ_3
4	90°	l_4	0	θ_4
5	0°	l_5	0	θ_5

Transformation Matrices:

$$ {}^{0}_{1}T = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ \sin\theta_1 & 0 & -\cos\theta_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{1}_{2}T = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & l_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & l_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{2}_{3}T = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & l_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & l_3\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$
{}^{3}_{4}T = \begin{bmatrix} \cos\theta_4 & 0 & \sin\theta_4 & l_4\cos\theta_4 \\ \sin\theta_4 & 0 & -\cos\theta_4 & l_4\sin\theta_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{4}_{5}T = \begin{bmatrix} \cos\theta_5 & -\sin\theta_5 & 0 & l_5\cos\theta_5 \\ \sin\theta_5 & \cos\theta_5 & 0 & l_5\sin\theta_5 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{0}_{5}T = \begin{bmatrix} s_1c_{234}c_5 + s_1s_5 & -c_1c_{234}s_5 + s_1c_5 & s_{234}c_1 & \begin{matrix} l_5[s_1c_{234}c_5 + s_1s_5] + \\ l_4c_1c_{234} + c_1[l_3c_{23} + l_2c_2] \end{matrix} \\ s_1c_{234}c_5 - c_1s_5 & -s_1c_{234}s_5 - c_1c_5 & s_{234}s_1 & \begin{matrix} l_5[s_1c_{234}c_5 - c_1s_5] + \\ l_4s_1c_{234} + s_1[l_3c_{23} + l_2c_2] \end{matrix} \\ s_{234}c_5 & -s_{234}s_5 & c_{234} & \begin{matrix} l_1 + l_2s_2 + l_3s_{23} + \\ l_4s_{234} + l_5c_5s_{234} \end{matrix} \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Simulation:

The forward kinematic model was simulated to verify the validity of the manipulator following are the codes and results for the same:

DH Transform:

```
function t = DHTranform(alphai_1, lx, lz, theta)
rx=[1 0 0 0; 0 cosd(alphai_1) -sind(alphai_1) 0; 0 sind(alphai_1) cosd(alphai_1) 0;  0 0 0 1];
tx=[1 0 0 lx; 0 1 0 0; 0 0 1 0; 0 0 0 1];
tz=[1 0 0 0; 0 1 0 0; 0 0 1 lz; 0 0 0 1];
rz=[cosd(theta) -sind(theta) 0 0; sind(theta) cosd(theta) 0 0 ; 0 0 1 0; 0 0 0 1];
t=rz*tz*tx*rx;
end
```

Visualization

```
% innitialise necessary variables:
l1=0;l2=0; l3=0;l4=0;l5=0;%assumed in mm
t1=0;t2=0; t3=0; t4=0; t5=0;%assumed in degree
w=10;
hold on
grid on
%manipulator diagram
%defining variable values
l1=109;l2=100.19; l3=101.4;l4=90.03;l5=99.5;%assumed in mm
%position
t1=0;t2=-45; t3=-45; t4=0; t5=30;
%base coordinates
o0=[0;0;0];
%setting up display view
hold on
```

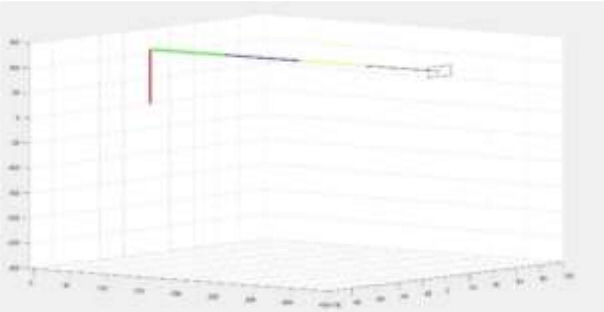


FIGURE 2(A)

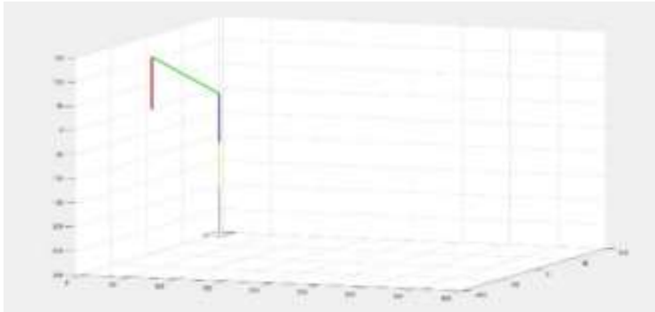


FIGURE 2 (B)

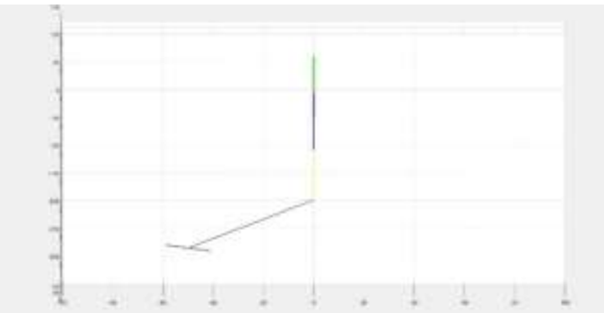


FIGURE 2(C)

FIGURE 2(D)

a: Default position| b: plane ground position| c: roll control| d: pitch control|

Workspace Plot

```
% innitialise necessary variables:
l1=0;l2=0; l3=0;l4=0;l5=0;%assumed in mm
t1=0;t2=0; t3=0; t4=0; t5=0;%assumed in degree
w=10;
hold on
grid on
%manipulator diagram
```

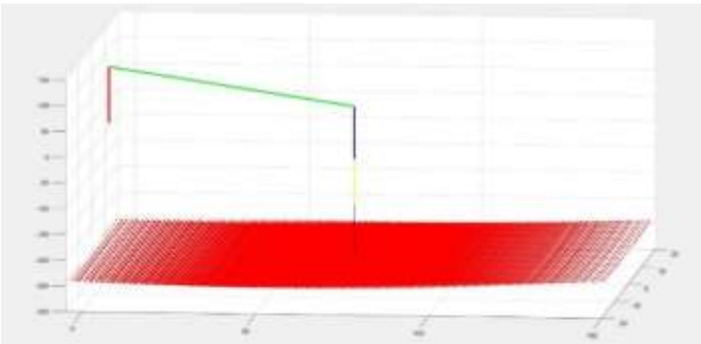


FIGURE 3

Structure and Dynamics

Structure of Jay Robotix (Version 1.3.0) robot manipulator:

A robotic manipulator with 5-DOF can perform a lot of operations especially if the end-effector is chosen accordingly and still it is very lightweight and cost-effective. The manipulator available to us is a 4-DOF manipulator and has a mass of only 7kg. It has a horizontal reach of 29.16 cm and a vertical reach of 40.06 cm from the ground. The manipulator consists of 6 Servos including the 4 which provide the Degrees of freedom to the Arm and 1 for the end-effector's rotation and 1 for the end-effector's opening and closing. The wireframe model of the robotic manipulator as prepared using SolidWorks software is shown in Figure 1, which includes the important aspects of the manipulator and the working range (in terms of the angles) of each of the links is presented in Table 2.

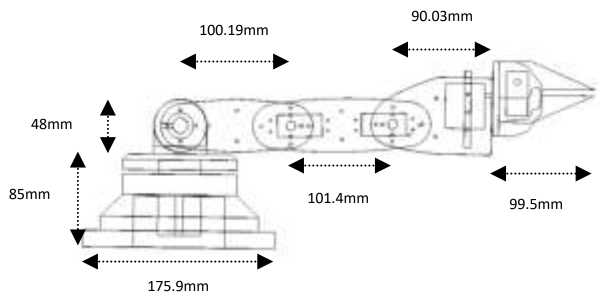


FIGURE 4-WIREFRAME MODEL-IMPORTANT ASPECTS OF JAY ROBOTIX (1.3.0)

TABLE 2-WORKING RANGE OF EACH AXIS (MECHANICAL CONSTRAINT)

Rotational limit (in degrees)	Axis movement
0 to 180	Axis 1 rotation (Link 1)
90 to -90	Axis 2 arm (Link 2)
130 to -130	Axis 3 arm (Link 3)
120 to -120	Axis 4 arm (Link 4)

The modified structure of the manipulator so as to carry out this project will require an extra link at the end of link 4 with its joint being revolute and the end-effector also needs to be replaced with a plate-like end-effector. The revolute axis of this new link will be in the same direction as that of the twist joint axis as shown in Figure 2 below.

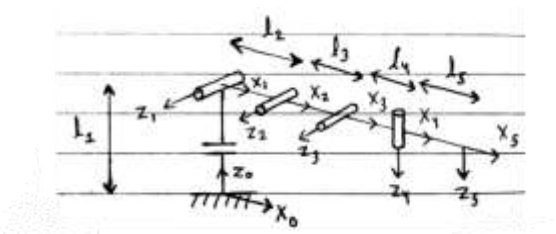


FIGURE 5-FRAME ASSIGNMENT MODEL

Mechanical Design of Jay Robotix (Version 1.3.0)

Robot manipulator modal design was prepared in SolidWorks software. To know the dynamic performance of the robot manipulator, the centre of mass and moment of inertia of each of the links were required. And since there was no precise information about the moment of inertia or centre of mass so we simulated a model in the SolidWorks software environment to get them. These two values for each of the links were derived from the "mass properties" menu in SolidWorks. The schematic model which we simulated in the SolidWorks software can be seen in Figure 3.

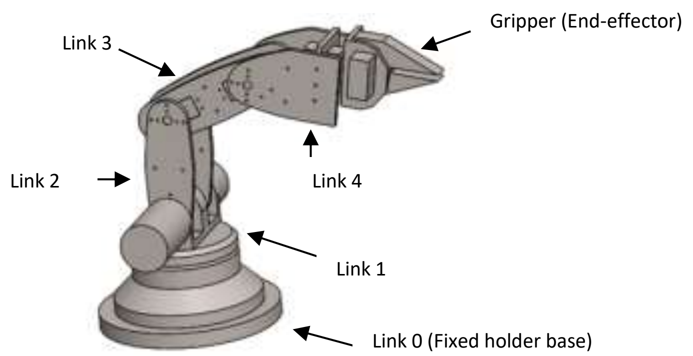


FIGURE 6-SCHEMATIC OF JAY ROBOTIX (1.3.0) MANIPULATOR SIMULATED IN SOLIDWORKS

The above robot manipulator's stimulated model consists of 4 links, one fixed holder base and one gripper (end-effector). And since the manipulator is an open-kinematic chain so the number of links is equal to the Degree of freedom of the system hence it is a 4 degree of freedom system. The model includes 1 Twist joint and 3 Rotary joints.

Table 3 lists the general features of the 4-DOF robot manipulator and the dynamic specifications of each of its links. The table includes the information like length, weight, moment of inertia and center of mass of each of the links with reference to the individual coordinate systems.

The above design will be modified once the necessary changes have been introduced in the structure.

TABLE 3

Dynamic characteristics of all components of Jay Robotix (1.3.0) robot

Center of mass expressed in base coordinate frame/m			Moment of inertia for center of mass and expressed in center of mass coordinate frame/(kg·mm^2)									Mass/ kg	Length/ m	Links of Jay Robotix (1.3.0) robot simulated by SolidWorks
z	y	x	I_{zz}	I_{zy}	I_{zx}	I_{yz}	I_{yy}	I_{yx}	I_{xz}	I_{xy}	I_{xx}			
0.039	0	0	2466.08	0	0	0	1558.21	0	0	0	1561.09	0.77	0.175	 Link 0
-0.072	0	0.017	1227.38	0	-71.96	0	1231.99	0	-71.96	0	241.83	0.38	0.104	 Link 1
-0.048	0	0	5.05	0	0	0	44.81	0	0	0	39.93	0.03	0.138	 Link 2
–0.047	0	0	6.24	0	0	0	64.07	0	0	0	57.99	0.029	0.162	 Link 3
-0.031	0	-0.001	12.28	0	2.12	0	41.64	0	2.12	0	29.52	0.034	0.110	 Link 4

End Effector Design

Overview

All the sensors required for collision avoidance will be on the end effector itself. It houses 9 micro ultrasonic sensors arranged in a 3x3 grid, to model the surface beneath it for further calculations of closest distance and orientation. Based on these calculations the manipulator adjusts the end effector which avoid collision and tries to maintain a constant distance from the surface to continue scanning the surface for landmines while avoiding any collisions from the undulations from the surface.

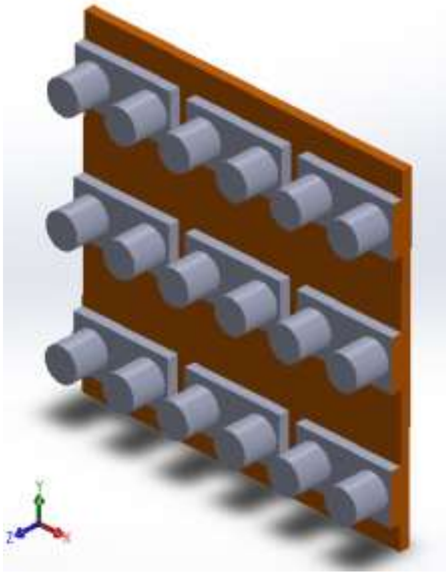


FIGURE 7: END EFFECTOR CAD

Sensor Placement and Surface Mapping

An approximate digital reconstruction of the surface is necessary for the calculation of proximity of the surface with respect to the end effector. In order to facilitate this the ultrasonic sensors were arranged in 3x3 array as shown in the figure. These 9 sensors return 9 coordinate points, whose x and y coordinates are assumed to be the center point of the ultrasonic sensor IC and the z coordinate will be the depth returned by these sensors.

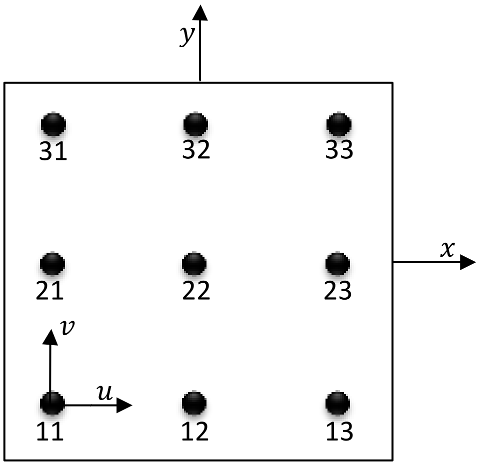


FIGURE 8: CENTERS OF ULTRASONIC SENSORS ARRANGED ON THE END EFFECTOR FRAME

These 9 points form a point cloud from which a Bezier Surface Patch is constructed. The closest point from the end effector will be the minima of the Surface patch.

A Bezier surface Patch can be formulated as follows:

Let $P_{ij} = \begin{bmatrix} x_{ij} \\ y_{ij} \\ z_{ij} \end{bmatrix}$ be a point from the point cloud and let u and v be the scalars in the directions as shown in the figure. Then any point $P(u, v)$ on the Bezier Surface Patch can be calculated as follows:

$$P(u,v) = \sum_{j=0}^{2}\sum_{i=0}^{2} B_i^2(u)B_j^2(v)P_{ij}$$

Where,

$$B_i^n(u) = \binom{n}{i} u^i(1-u)^{n-i}$$

However, for the calculation of the orientation that the end effector needs to assume, doesn't require the formation of Bezier Surface Patch, it can be easily calculated with the point cloud itself.

Calculation of the Orientation

The new orientation that the end effector would need to assume to avoid collision can be calculated by finding the Best Fit Plane using the least square method (3).

Let $P_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}$, such that $i \in \{1, 2, \dots, 9\}$, represent a point from the set of cloud points and let $ax + by + z = c$ be the best fit plane.

Let $\bar{P}$ represent the centroid of the plane then, $\bar{P} = \begin{bmatrix} \bar{x} \\ \bar{y} \\ \bar{z} \end{bmatrix}$, where $\bar{x} = \frac{1}{9}\sum_{i=1}^{9} x_i$. It can be observed that the way end effector is constructed it makes $\bar{x} = \bar{y} = 0$. Hence

$$\bar{P} = \begin{bmatrix} 0 \\ 0 \\ \bar{z} \end{bmatrix}$$

Let the total squared error be e. Then,

$$e = \sum_{i=1}^{9} (z_i - c + ax_i + by_i)^2$$

Minimizing e with respect to a, b, and c we get,

$$\begin{bmatrix} \sum x_i^2 & \sum x_i y_i & -\sum x_i \\ \sum x_i y_i & \sum y_i^2 & -\sum y_i \\ \sum x_i & \sum y_i & -\sum 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} -\sum x_i z_i \\ -\sum y_i z_i \\ -\sum z_i \end{bmatrix}$$

The solution for $\begin{bmatrix} a \\ b \\ c \end{bmatrix}$ has to be calculated every time the data is retrieved from the sensors. Given the complicated nature of the equation, this seems to be a tedious task. But it should be noted that the matrix on LHS is constant and depends on the arrangement of the ultrasonic sensor array on the end effector and is therefore constant, hence, it has to be calculated only once.

Let the unit normal vector to the best fit plane be $\hat{\eta}$, then

$$\hat{\eta} = \frac{1}{\sqrt{a^2 + b^2 + 1}} \begin{bmatrix} a \\ b \\ 1 \end{bmatrix}$$

For end effector to assume that orientation, it has to rotate such that z axis of the end effector coincides with $\hat{\eta}$.

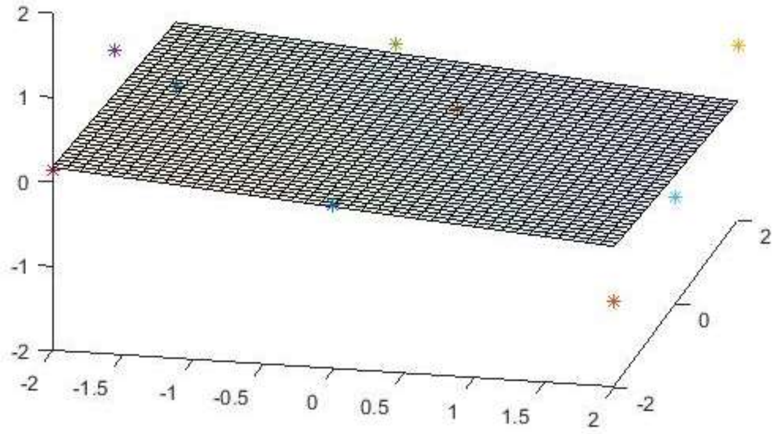


FIGURE 9: BEST FIT PLANE PLOT WITH CLOUD POINTS

Calculation of End-effector center position

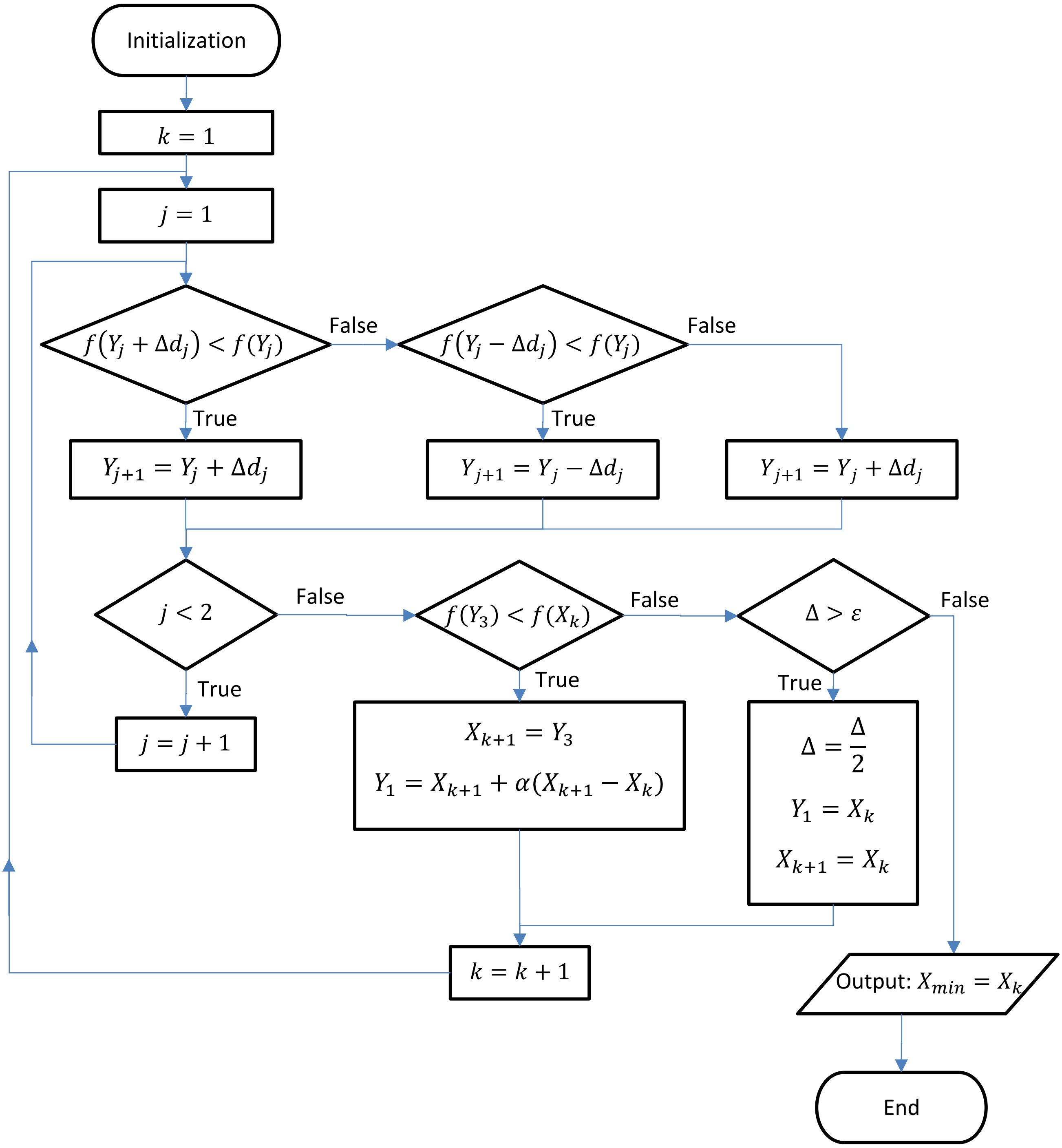
Every point P on our Bezier surface patch is parametrized with parameters u and v, and the positions of the centres of the sensors do not change with respect the end effector frame, therefore the x and y coordinates of P do not change with time and are only functions of u and v and hence has to be calculated only once. However, the z coordinate changes as the end effector moves. Hence to find the closest point to the end effector, we only need to find the minimum of z, where

$$z = f(u, v)$$

This can be achieved by any numerical minimization search algorithm. Hookes and Jeeves Method with Discrete steps (4) will be our algorithm of choice. The algorithm works as follows:

Initialization:

let $X = \begin{bmatrix} u \\ v \end{bmatrix}, d_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, d_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and choose a termination scalar $\varepsilon > 0$, initial step size $\Delta > \varepsilon$, and an acceleration factor $\alpha > 0$. Choose a starting point X_1, and let $Y_1 = X_1$ and then follow the algorithm stated below.



The closest point z_{min} occurs at the output of the algorithm X_{min}. Let $X_{min} = \begin{bmatrix} u_{min} \\ v_{min} \end{bmatrix}$, then the corresponding point P_{min} can be calculates as

$$P_{min} = P(u_{min}, v_{min}) = \begin{bmatrix} x_{min} \\ y_{min} \\ z_{min} \end{bmatrix}$$

Compiling position and orientation of End-effector

Let us assume that D is the distance that the end effector needs to maintain from the closest point for safe operation. Then the new distance of the origin of the end effector frame and centroid of the best fit plane, that the end effector will have to maintain, will be $D +$ the vertical distance of the peak from the best fit plane.

Let the point that origin of the frame of end effector will translate towards will be P_{new}, then

$$P_{new} = \bar{P} - (D + z_{min} - c + ax_{min} + by_{min}) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

And the new orientation can be achieved by rotating the end effector frame about some vector M and by some angle θ such that the z axis of the frame coincides with $\hat{\eta}$, then

$$M = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \hat{\eta}$$

and,

$$\theta = \cos^{-1}(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \cdot \hat{\eta})$$

References

1. **Jazar, Reza N.** *Theory of Applied Robotics.* s.l. : Springer.

2. *The Dexterous Workspace of Simple Manipulators.* **ZONE-CHANG LAI, CHIA-HSIANG MENQ.** 1, s.l. : IEEE Journal on Robotics and Automation, 1998, Vol. 4. 99-103.

3. **Eberly, David.** Least Squares Fitting. [Online] July 15, 1999. http://www.sci.utah.edu/~balling/FEtools/doc_files/LeastSquaresFitting.pdf.

4. **Bazaraa, Mokhtar S., Sherali, Hanif D. and Shetty, C. M.** *Nonlinear Programming Theory and Algorithms.* s.l. : Wiley Intersience, 2006.

5. **R K Mittal, I J Nagrath.** *Robotics and Control.* s.l. : MCGrawHill.

6. *Forward and Inverse Kinematic Models for an.* **Dayal R. Parhi, BBVL Deepak, Devedutta Nayak and Anand Amrit.** 2, s.l. : I.J.A.I.C.R, 2012, Vol. 4.