**Northeastern University**
360 Huntington Ave, Boston, MA 02115

**Report on**
**Data Acquisition of RTK based GNSS system and Data Analysis**

**LAB_2** - EECE 5554 Robotics Sensing and Navigation

**Submitted by:**
**Pavan Rathnakar Shetty**

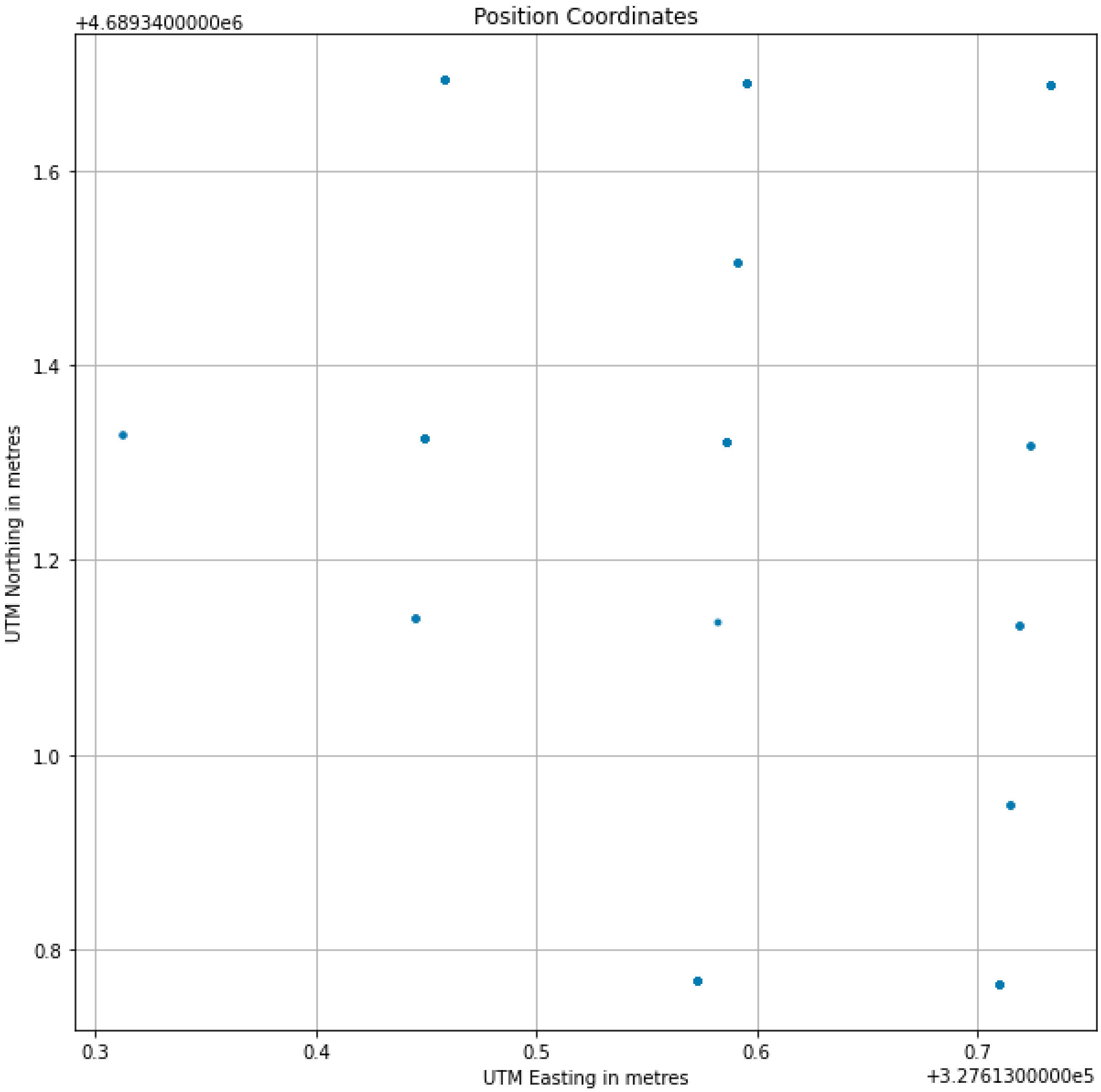
**Submitted to**
**Hanumant Singh**
**Professor, Electrical and Computer Engineering**

**Open Space data were collected near the Common Sentinel and occluded space data were collected near West Village.**

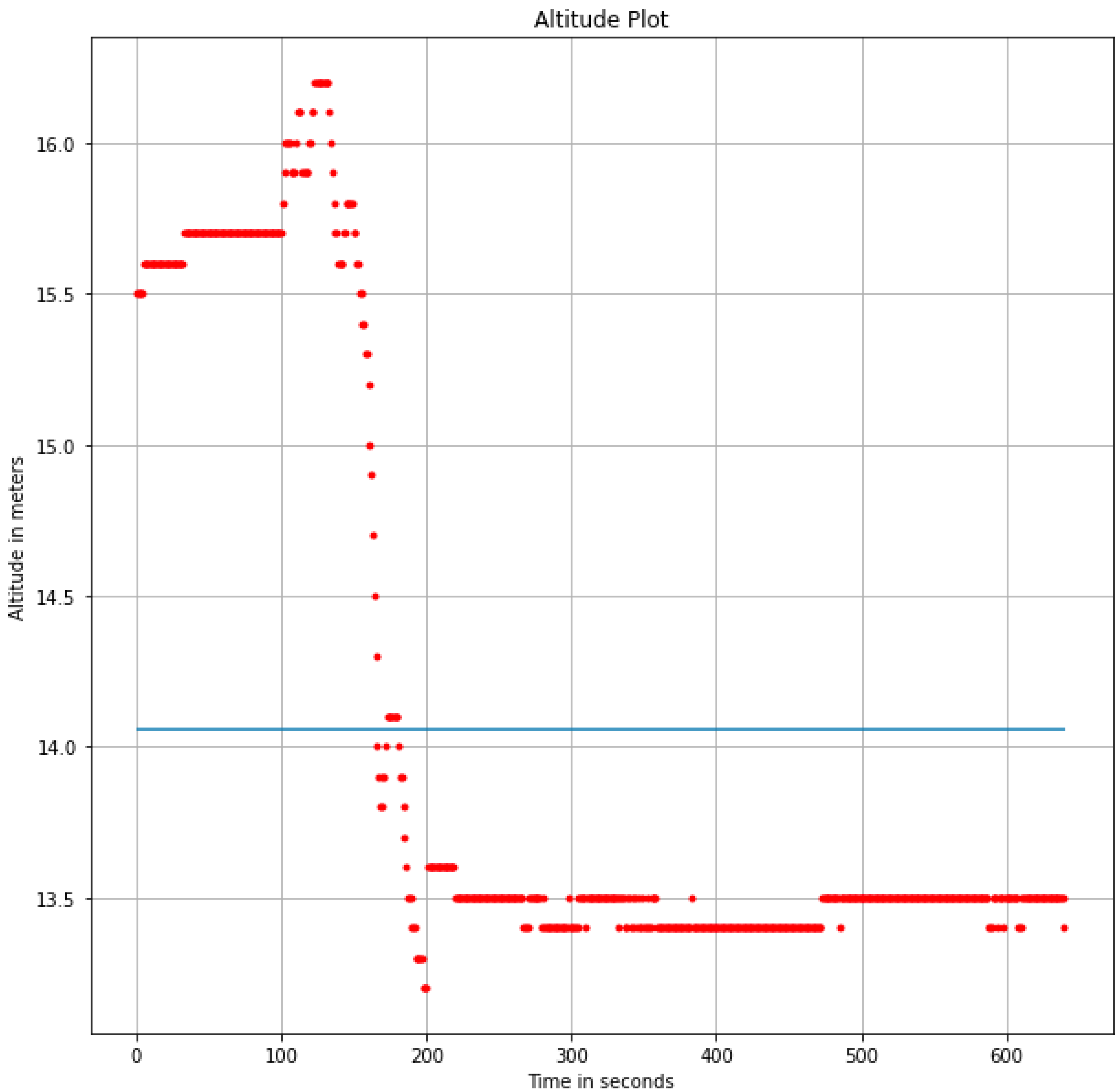
**The Data was taken from Dhruv Shah's team (Team Number 2)**

**Detailed Analysis:** Graphical plotting and Analysis of the CSV file was done using Jupyter Notebook by using Python and libraries like Matplotlib, CSV, Numpy, Statistics, and Pandas as described in the RTKDATAVISUALIZATION.ipynb file.

**Stationary Data**:
In occluded space:
**utm_easting vs utm_northing:**



**Altitude vs Time:**




SDE: Standard Deviation of utm_easting
SDN: Standard Deviation of utm_northing
SDA: Standard Deviation of Altitude.

```
SDE #standard deviation of utm easting
```

0.09800500577610387

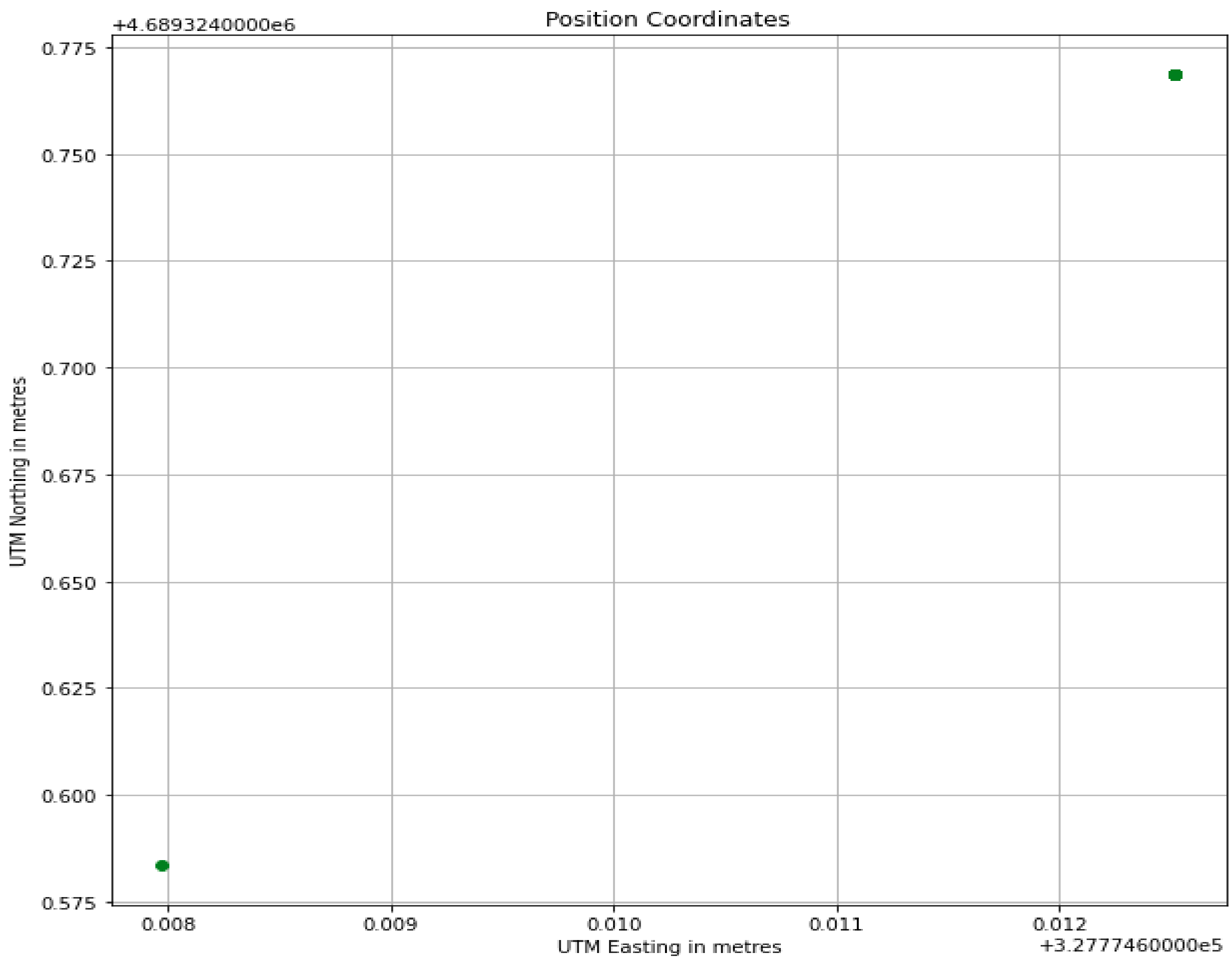
```
SDA #standard deviation of altitude
```

0.9878523088434642

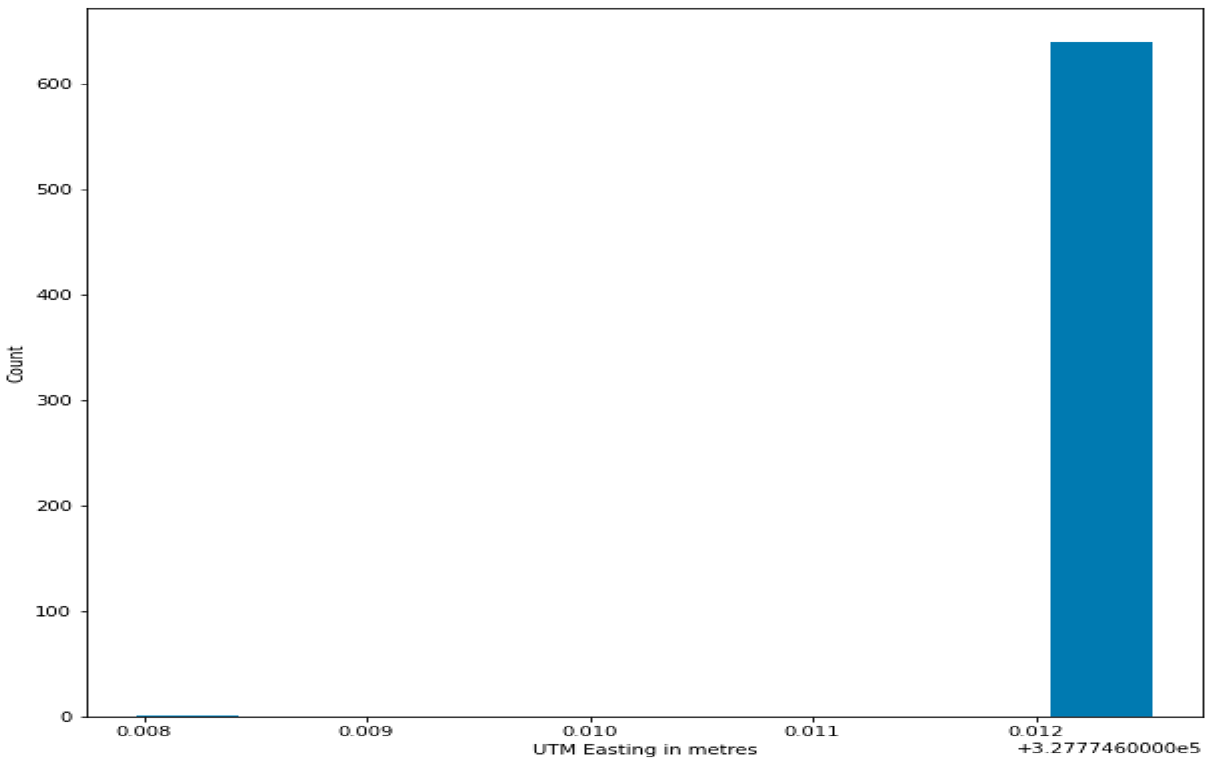
```
SDN #standard deviation of utm northing
```

0.38581442693642787

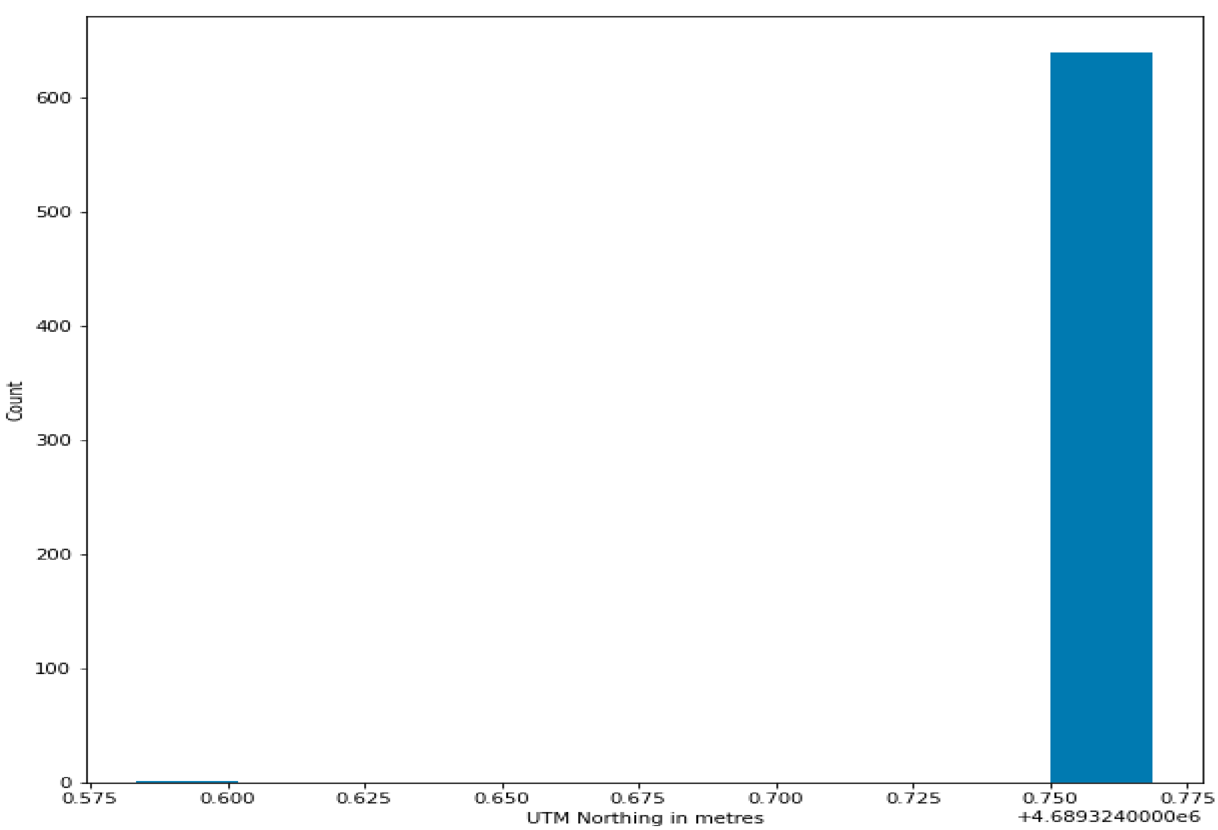
**Stationary Data**:
In open space:
**utm_easting vs utm_northing:**




**Histogram of utm_easting and utm_northing:**

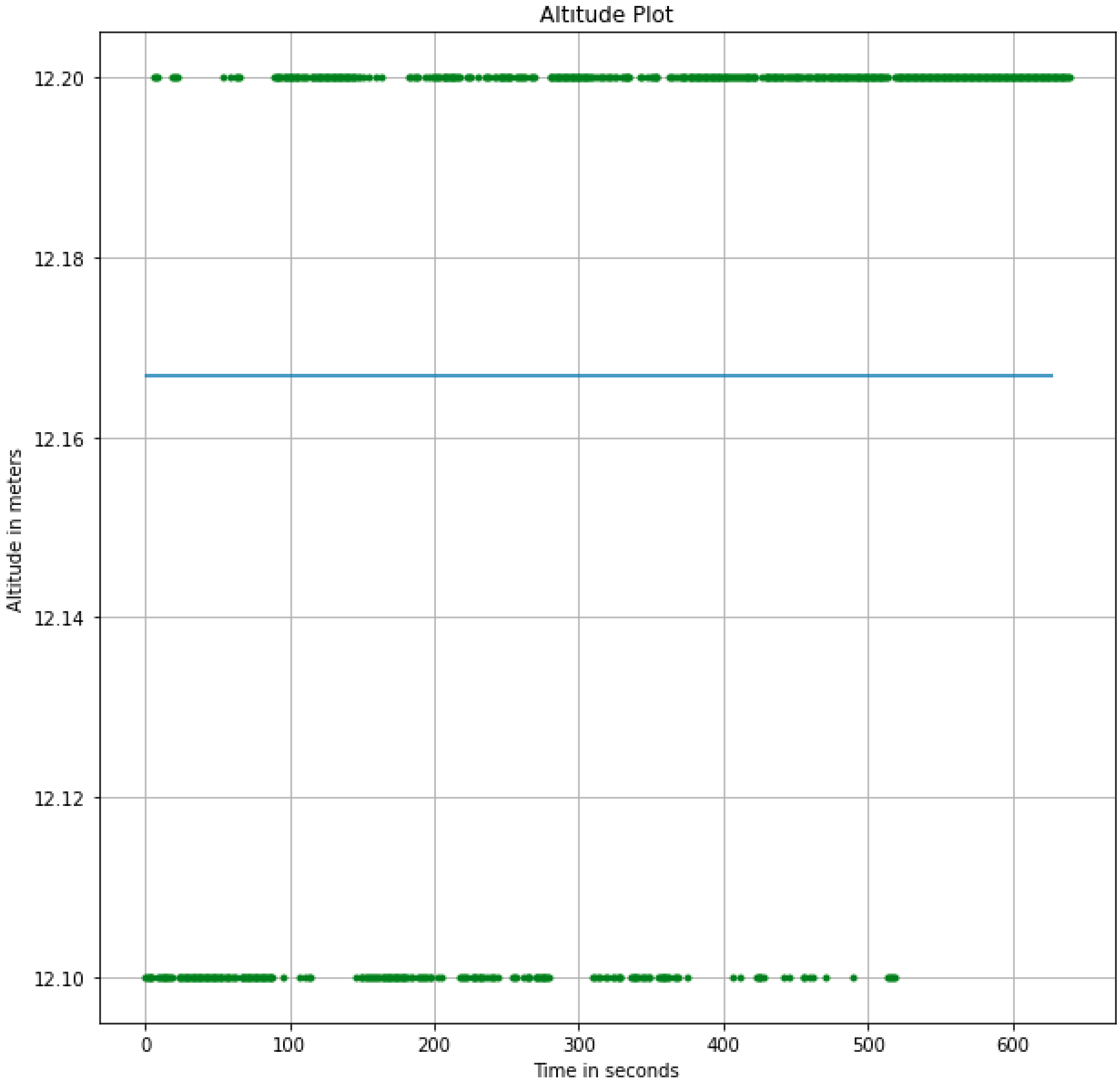


Count
600
500
400
300
200
100
0
0.008
0.009
0.010
0.011
0.012
+3.2777460000e5
UTM Easting in metres

Count
600
500
400
300
200
100
0
0.575
0.600
0.625
0.650
0.675
0.700
0.725
0.750
0.775
+4.6893240000e6
UTM Northing in metres

**Altitude vs Time:**




```
SDE #standard deviation of utm easting
```

```
0.00017985494506217 17
```

```
SDN #standard deviation of utm northing
```

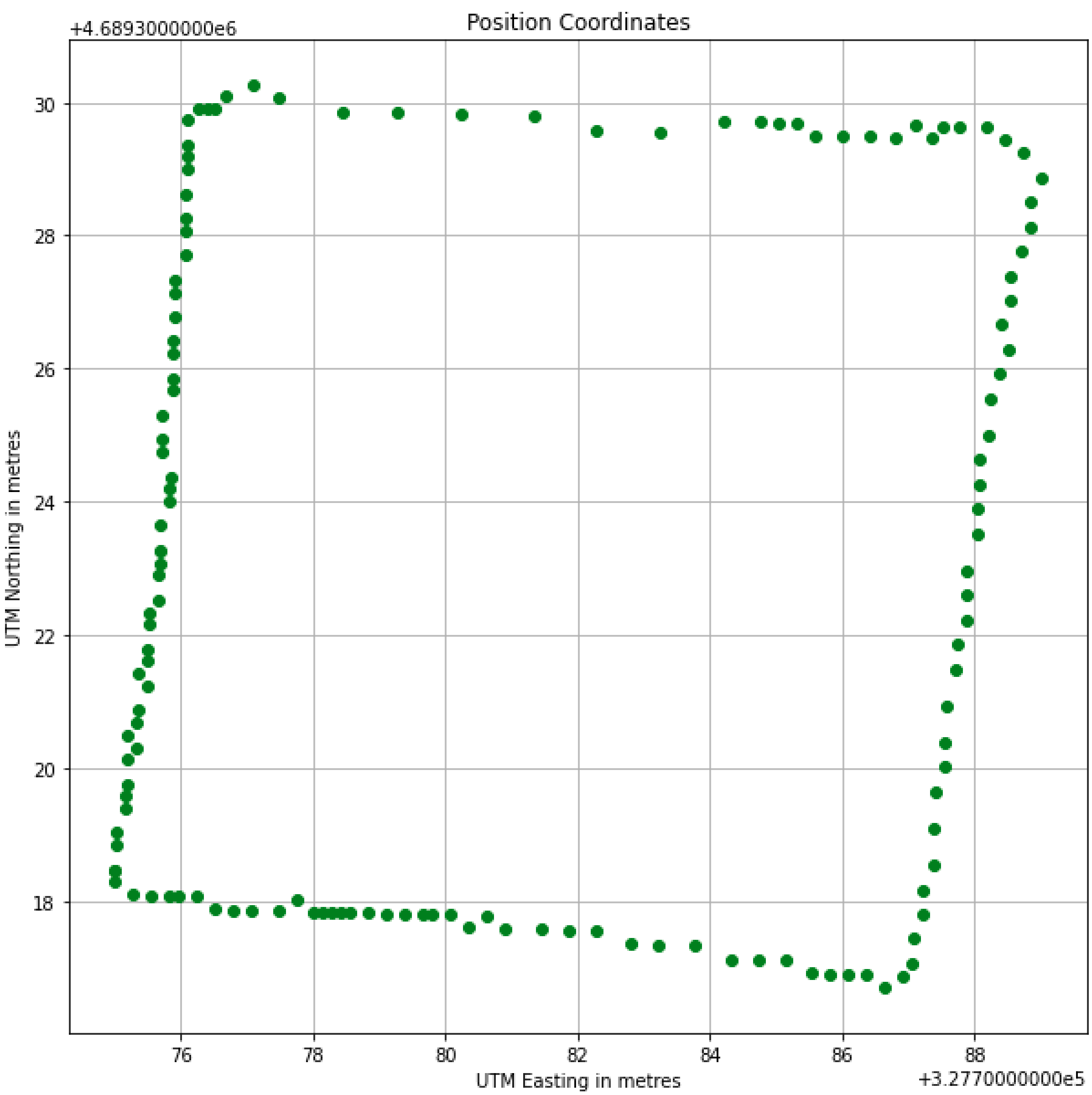
```
0.007315559727745076
```

```
SDA #standard deviation of altitude
```

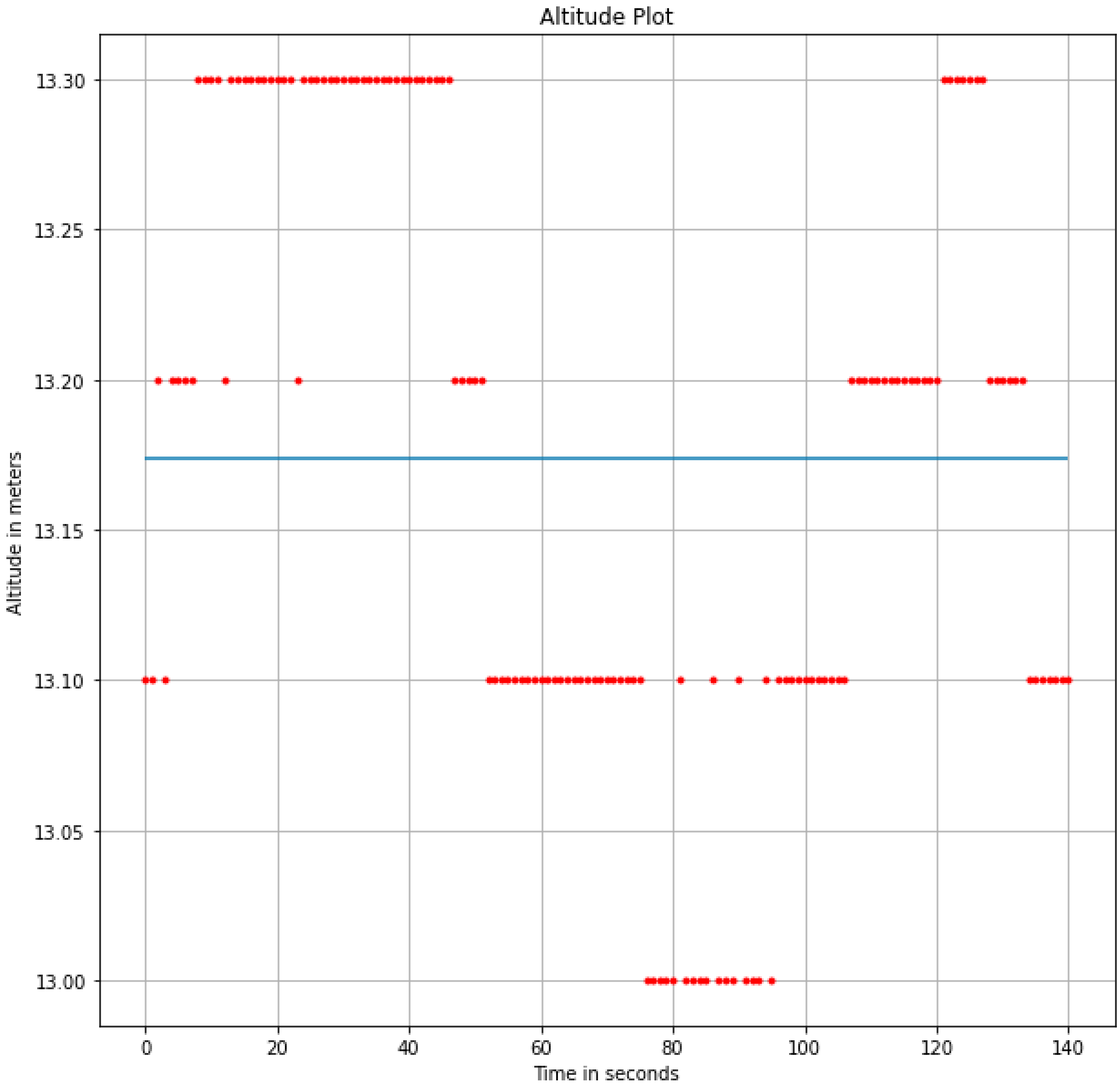
```
0.04715886272306954
```

SDE: Standard Deviation of utm_easting
SDN: Standard Deviation of utm_northing
SDA: Standard Deviation of Altitude.

**Moving Data:**
In occluded space:
**utm_easting vs utm_northing:**



**Altitude vs Time:**




SDA: Standard Deviation of Altitude

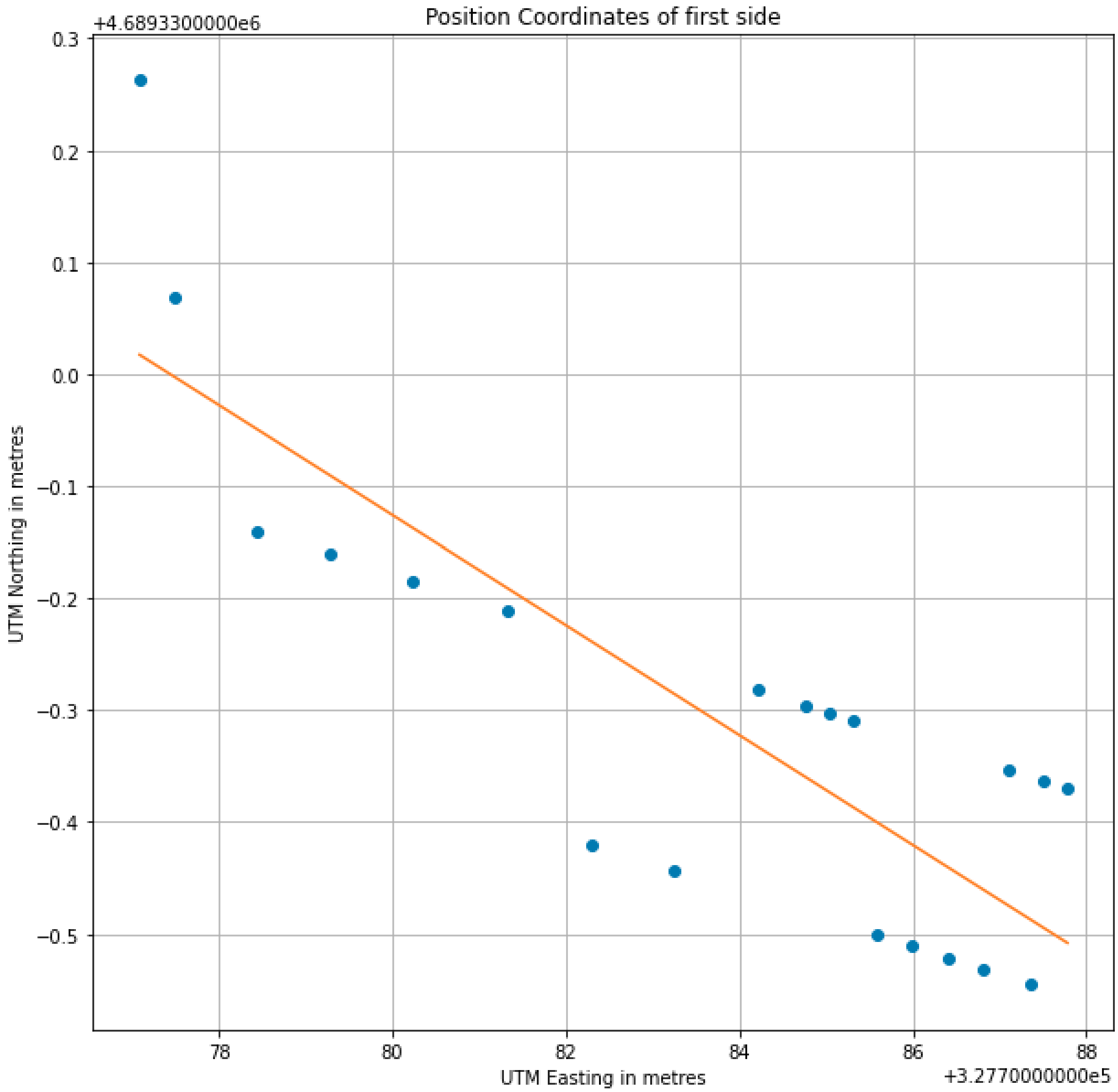
```
SDA
```

```
0.04715886272306954
```

**I have divided the square-looking path into four lines and analyzed each side for better understanding and visualization.**

I have plotted the position coordinates with a line of best fit and calculated Mean Square Error for the same.

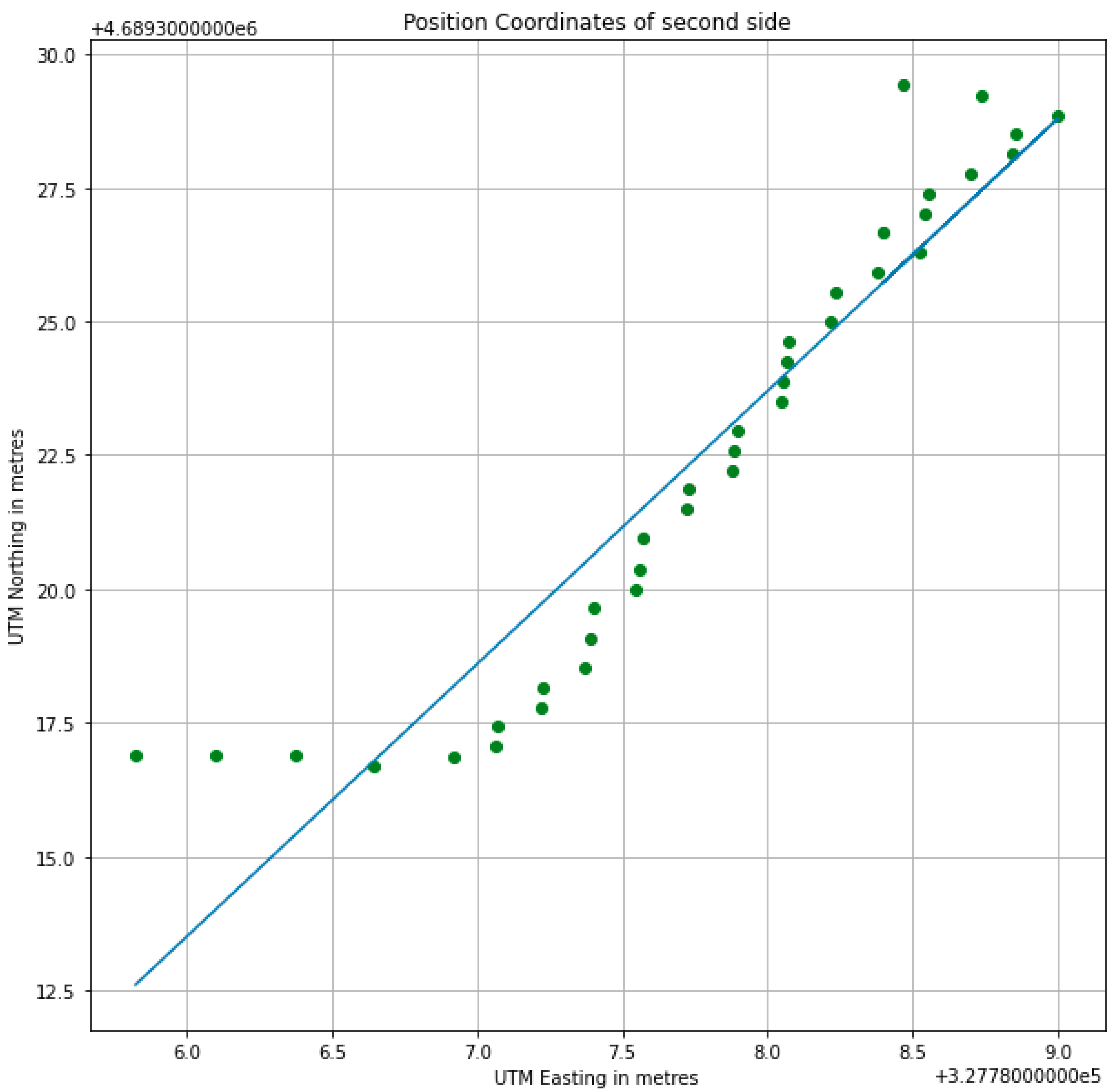
Following are the 4 side analysis:




MSE: Mean Square Error

```
#calculation of mean square error
np.array(new_y1)
MS=np.square(np.array(new_y1)-np.array(n311))
sum = 0
for i in MS:
    sum=sum+i
MSE=sum/20
print(MSE)
```

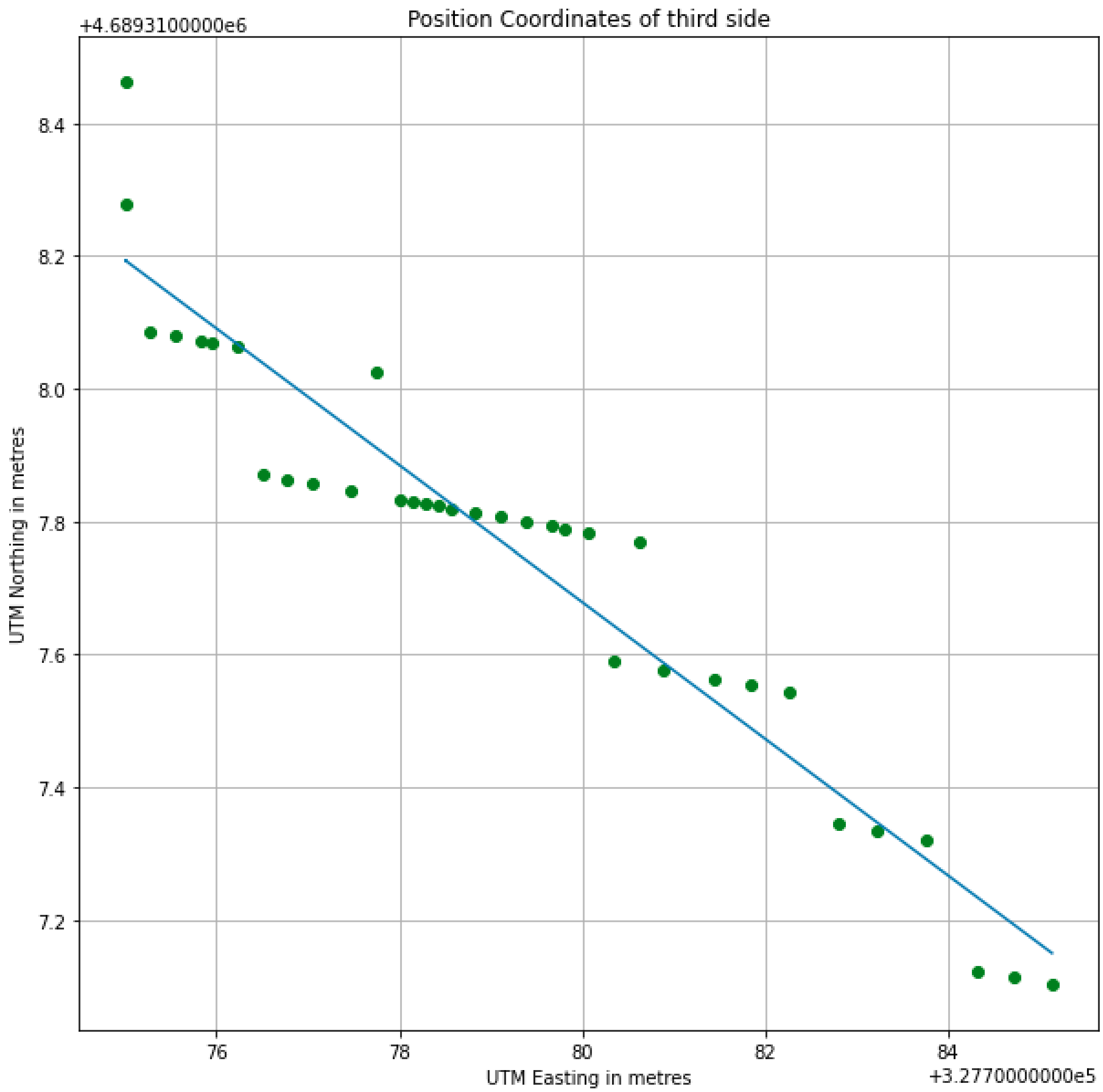
```
0.012079938681173252
```




MSE: Mean Square Error

```python
#mean square error
np.array(new_y4)
MS=np.square(np.array(new_y4)-np.array(n312))
sum = 0
for i in MS:
    sum=sum+i
MSE=sum/37
print(MSE)
```

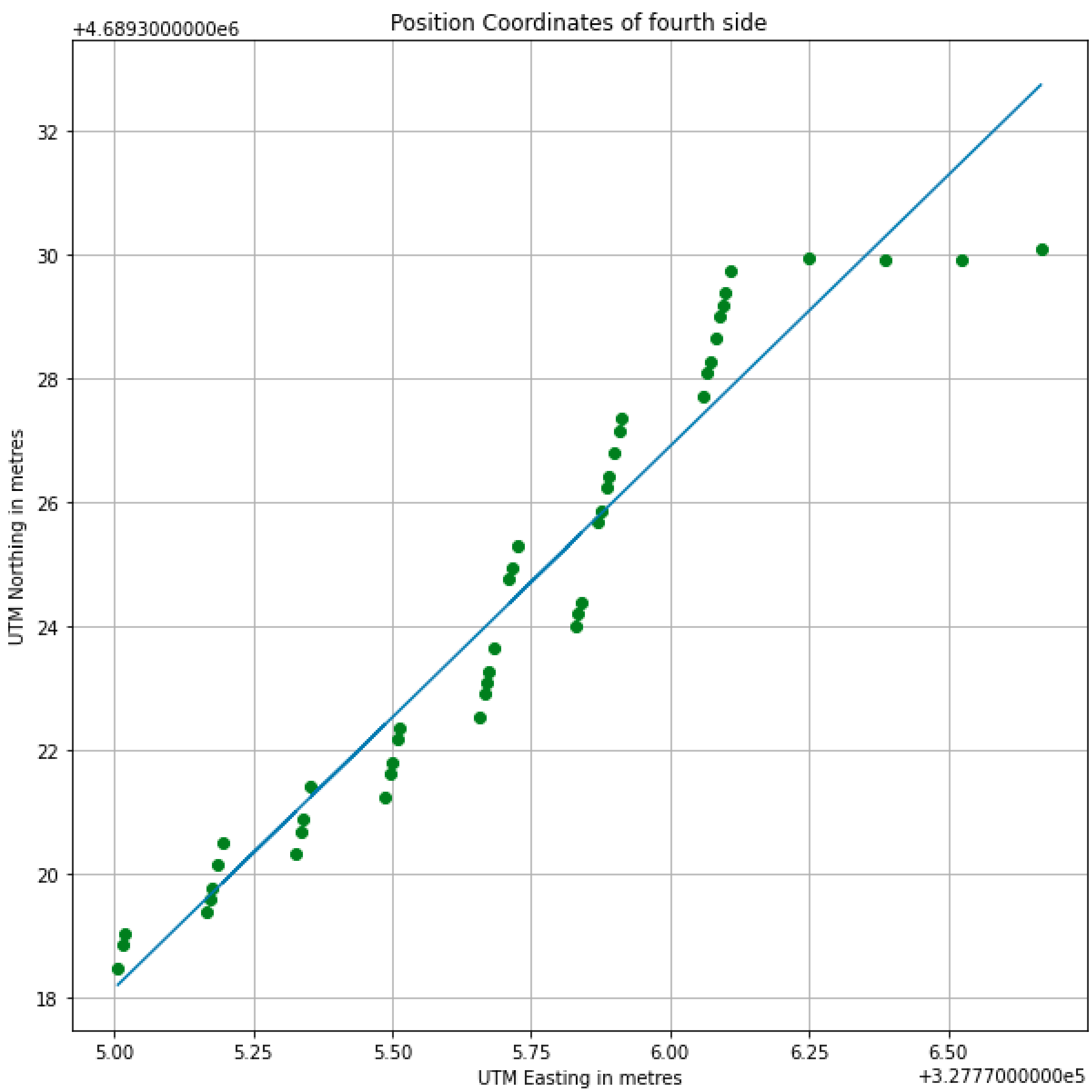
```
1.9802618450766978
```




MSE: Mean Square Error

```
#mean square error
np.array(new_y4)
MS=np.square(np.array(new_y4)-np.array(n313))
sum = 0
for i in MS:
    sum=sum+i
MSE=sum/35
print(MSE)
```

```
0.008045813065482696
```




MSE: Mean Square Error

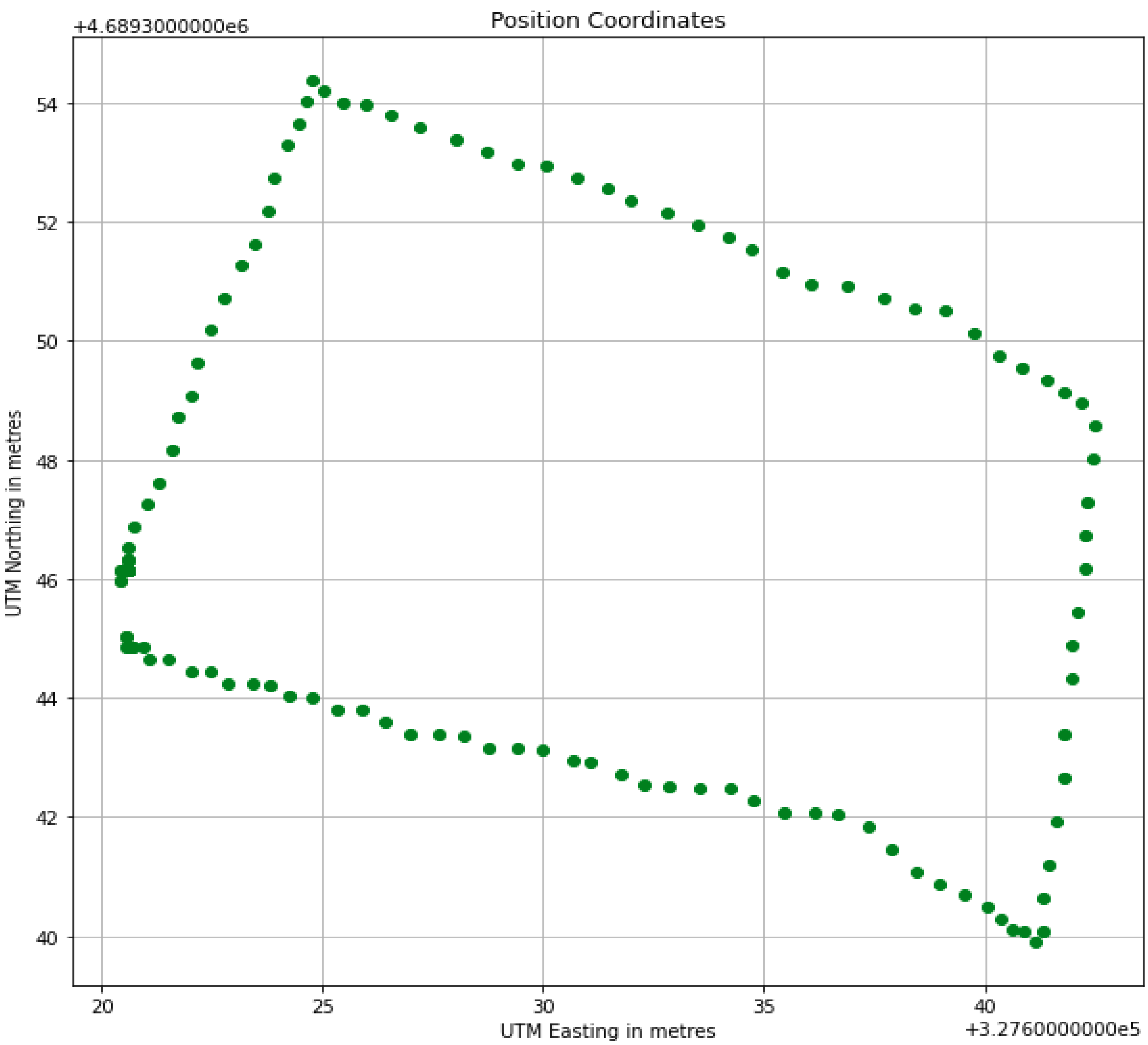
```
#mean square error
np.array(new_y4)
MS=np.square(np.array(new_y4)-np.array(n314))
sum = 0
for i in MS:
    sum=sum+i
MSE=sum/47
print(MSE)
```

0.8861922971220054

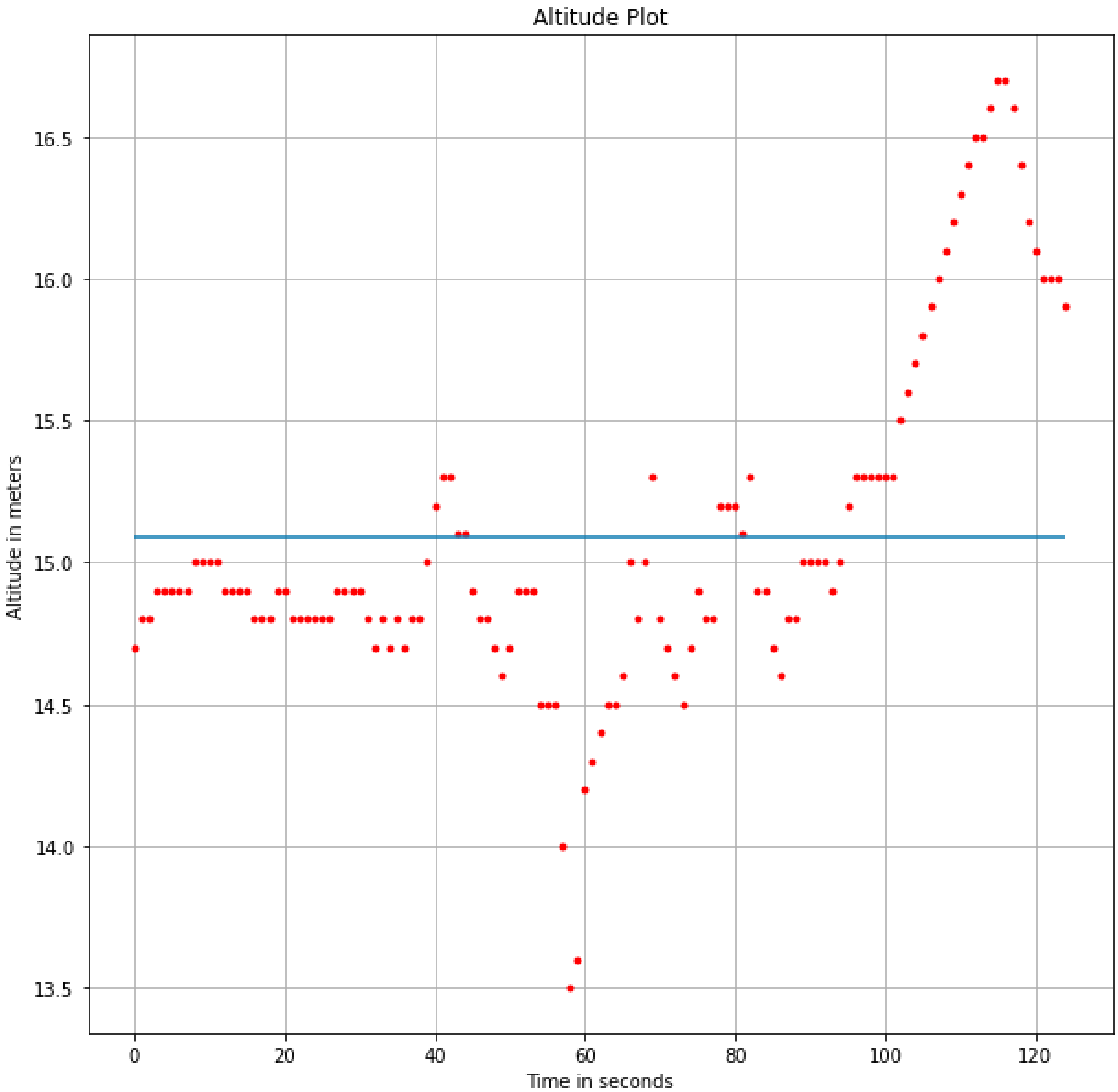
**Moving Data:**

In open space:

**utm_easting vs utm_northing:**



**Altitude vs Time:**




SDA: Standard Deviation of Altitude

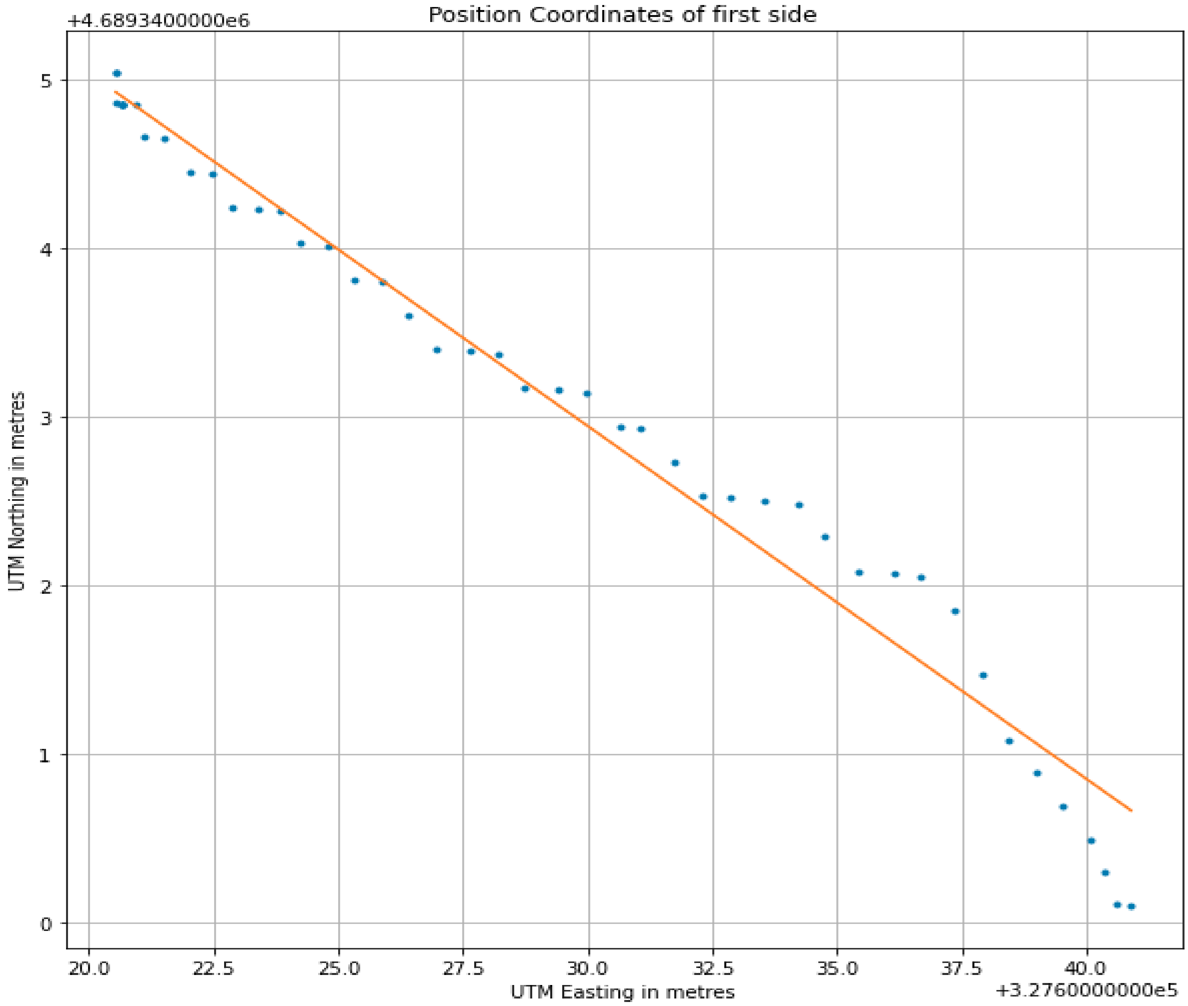
```
SDA
```

```
0.6006791854817075
```

**I have divided the square-looking path into four lines and analyzed each side for better understanding and visualization.**

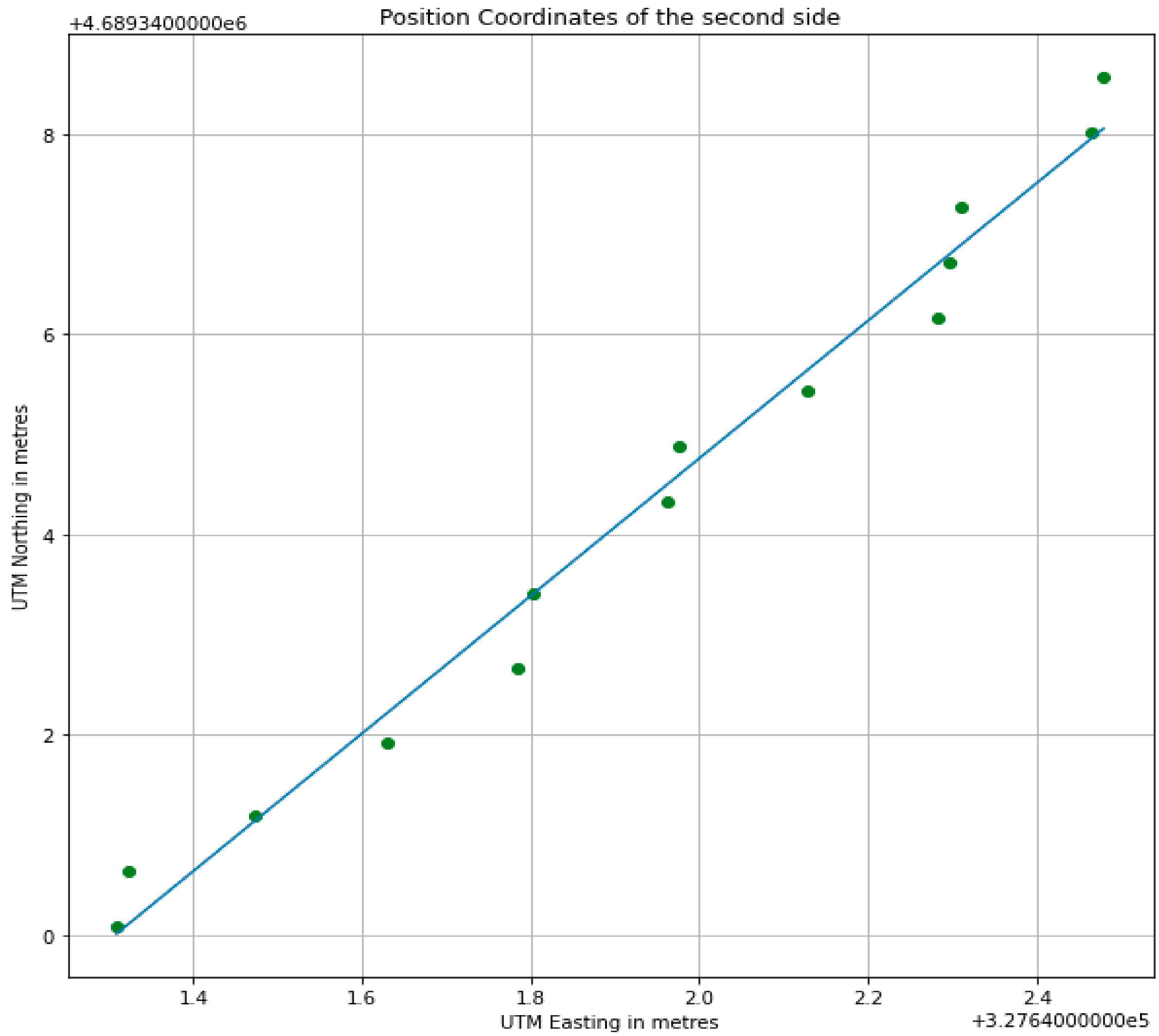
I have plotted the position coordinates with a line of best fit and calculated Mean Square Error for the same.

Following are the 4 side analysis:



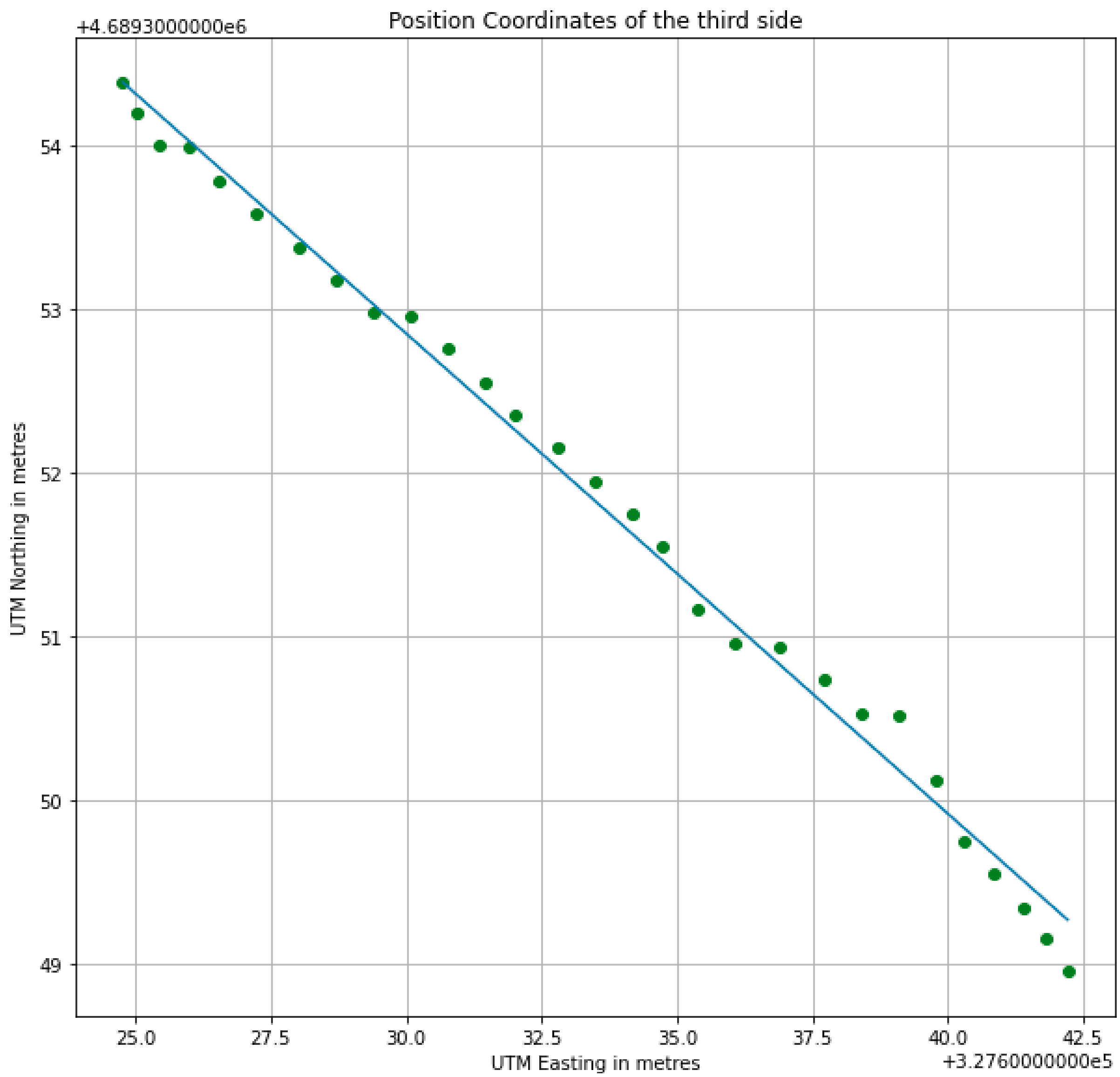

```
#calculation of mean square error
np.array(new_y1)
MS=np.square(np.array(new_y1)-np.array(n111))
sum = 0
for i in MS:
    sum=sum+i
MSE=sum/46
print(MSE)
```

```
0.057957408541276074
```



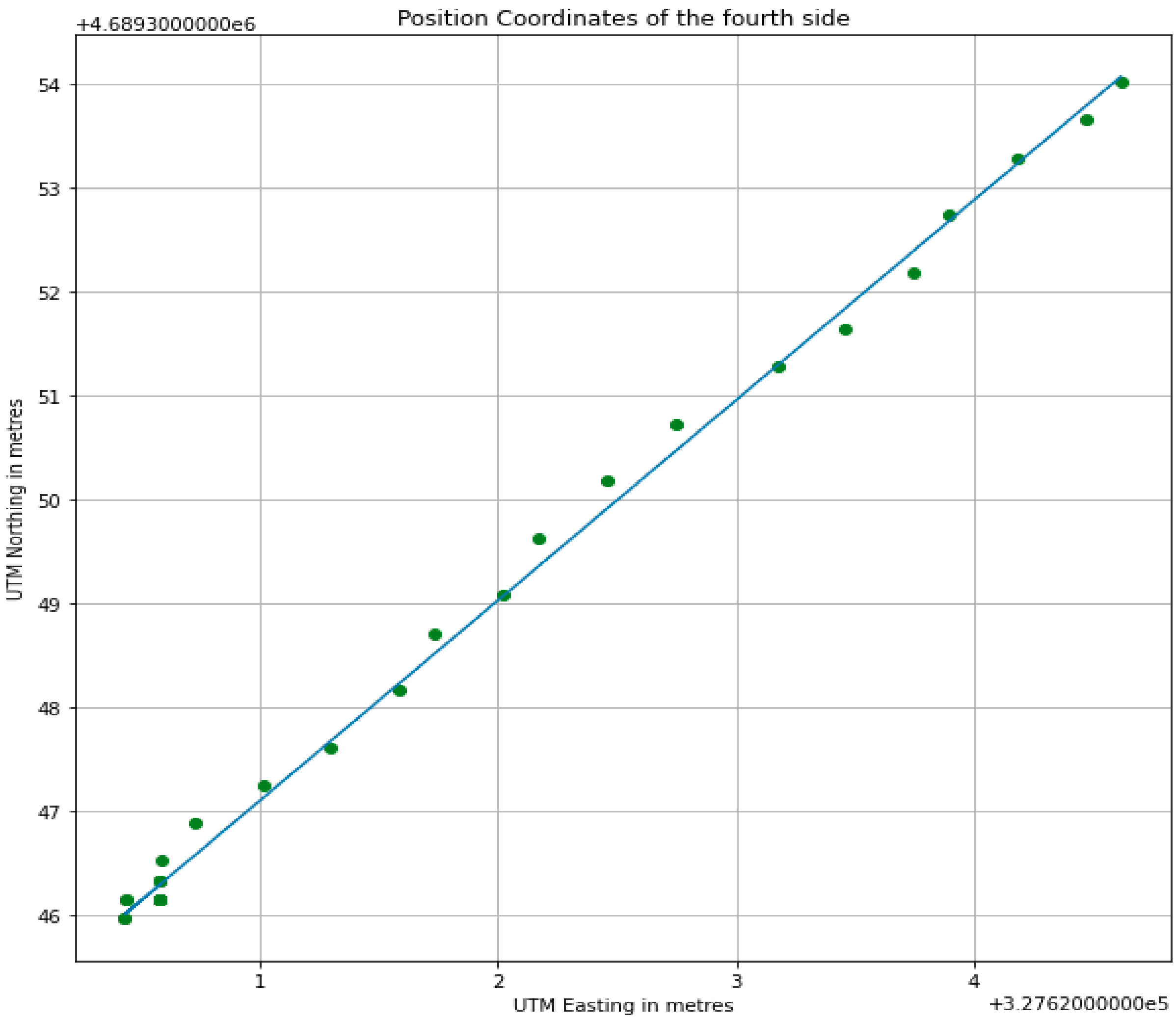

```
#calculation of mean square error
np.array(new_y2)
MS=np.square(np.array(new_y2)-np.array(n112))
sum = 0
for i in MS:
    sum=sum+i
MSE=sum/14
print(MSE)
```

```
0.1155429212713511
```




```
#calculation of mean square error
np.array(new_y3)
MS=np.square(np.array(new_y3)-np.array(n113))
sum = 0
for i in MS:
    sum=sum+i
MSE=sum/29
print(MSE)
```

```
0.020428921544716143
```




```
#calculation of mean square error
np.array(new_y4)
MS=np.square(np.array(new_y4)-np.array(n114))
sum = 0
for i in MS:
    sum=sum+i
MSE=sum/34
print(MSE)
```

```
0.02182972052108329
```

SDA: Standard Deviation of Altitude

SDN: Standard Deviation of UTM northing

SDE: Standard Deviation of UTM easting

MSE: Mean Square Error of UTM easting vs UTM northing

**Stationary data in occluded space:**

| SDA | SDE | SDN |
|---|---|---|
| 0.987 | 0.098 | 0.385 |

**Stationary data in open space:**

| SDA | SDE | SDN |
|---|---|---|
| 0.047 | 0.0001 | 0.00731 |

Since there is only one data deviated from the remaining set as shown in the histogram and the negligible values of SDE and SDN, the deviated value can be rejected assuming it is subjected to random noise.

**Moving data in occluded space:**

**SDA: 0.1025**

**RMS**

| First Side | Second Side | Third Side | Fourth Side | **Mean of the entire path** |
|---|---|---|---|---|
| 0.0120 | 1.98026 | 0.0080 | 0.88619 | **0.7216** |

**Moving data in open space:**

**SDA: 0.6006**

**RMS**

| First Side | Second Side | Third Side | Fourth Side | **Mean** |
|---|---|---|---|---|
| 0.0579 | 0.1155 | 0.0204 | 0.0218 | **0.0539** |

**Observation and Analysis:**

The nature of GPS errors are observed to be non-uniform. Transmitter errors, Receiver errors, Signal propagation errors, and interference of signals.
Good error measurement for the data can be the Mean Square Error for moving data and Standard Deviation for stationary data. Error metric for altitude remains the same for both moving and stationary data.

To account for better accuracy, the RTK system has correction measurements from the fixed GPS called the base sent to the Rover. As deploying an RTK system provides live corrections to data in real-time, RTK saves time by ensuring no post-processing is required, unlike the PPK system.
However, the RTK system requires a constant and consistent connection. Outages do happen and signals can be disrupted. If signals are lost, then it can result in a loss of correction data with the potential for inaccurate gaps during the acquisition. This does happen in cars using GPS and connection is reestablished. RTK works well in flat terrains without trees, mountains, or other obstructions and landmarks to get in the way of the communication between the rover and the base which is under stable setup conditions.

The error in the measurement/noise is due to the following factors:
1) PDOP, HDOP, and VDOP: DOP stands for Dilution of Precision. Dilution of Precision is a term used to describe the strength of the current satellite configuration, or geometry, on the accuracy of the data collected by a GPS receiver at the time of use. Thus, PDOP is the Position of DOP and can be thought of as 3D positioning or the mean of DOP, and most often referred to in GPS; HDOP is Horizontal of DOP; VDOP is Vertical of DOP. If there are more satellites available spread evenly throughout the sky, the better our positional accuracy will be (*and the lower the PDOP value*).
2) Any kind of signal disturbance from external factors like Buildings or any signal Interference.
3) Gps are unable to receive or communicate signals from the required number of satellites for better accuracy which results in inconsistency in measurement. GNS fix values are consistent in the data we collected and there is a deviation from the majority set when the number reduces in the GNS fix attribute.

I would prefer using PPK setup over RTK setup if I won't have the time to set up a solid connection, places with extreme conditions where it’s difficult to set up a stable network.

**Stationary open vs occluded:** The data we collected for open is almost constant except for 1 value which is considered redundant. While the setup is solid in both cases, apart from the sensor noise, the only added disturbances are the interference because of buildings. Hence the Stationary Data of open space is more accurate than the occluded space data.
**Moving open vs occluded:** Disturbances are the interference because of buildings. Hence the moving Data of open space is more accurate than the occluded space data.
**Stationary vs Moving:** Connection is much more stable in stationary setup, and the path taken in moving is subjected to corrections and change than stationary, Stationary data has stable correction data from the base to the rover.

**Altitude** correction or error estimates in stationary data in occluded space have a bigger standard deviation from stationary data in open space and moving data where the standard deviation is too negligible, suggesting stable data and good measurements. This is mainly because of the signal interference due to building and system error for the trial.

My error metrics and graphical analysis validate and justify the above reasons.