QUEENSLAND UNIVERSITY OF TECHNOLOGY

Mathematical Modelling of Target Road Trajectory by Polynomial Curve Fitting Algorithm in MATLAB

Supervisor: Yanming Feng

Project Type: Research

Student Name: YIXI ZHOU

Student Number: n9599291

Abstract

The growth of Autonomous vehicle technology sustains a skyrocketing trend in contemporary era. One of core problem of self-driving technology is to teach computer recognize trajectory of road. However, the basic principle of computer hardware is to deal with binary number, so a huge amount of reality issues are converted to math problem by programming, which allow computer to handle it. Therefore, this thesis addresses this issue that transform road graph to mathematical model. It formulates the problem as how to represent road trajectory by series polynomial equations. To solve this research question, it designs an experimental framework consisting of three following phases:

1. Data collecting and pre-processing: to utilize high precise GPS receiver installed in an automobile to collect the target road information in KMZ files and convert it to EXCEL files that allow MATLAB to manage it.
2. Data analysing and modelling: to exploit polynomial curve fitting algorithm in MATLAB to build target road mathematical model.
3. Evaluation: from the outcomes of phase two, to evaluate the accuracy for the road math model.

The major of the thesis is contributed from above three phases. 36 polynomial equations are constructed by MATLAB, which means only hundreds of parameters can be employed to depict the target road trajectory. It will save a great deal of space for computer memory to remember and recognize the road orbit, which assist different types of automatic transmission.

Contents

Introduction 1
  Research Background 1
  Research Questions and Objectives 3
Literature Review of Previous Work 4
Project Methodology 8
Outcomes 15
  Research Question 1 15
  Research Question 2 18
  Research Question 3 22
Discussion 30
Conclusion 34
Reflection 35
Reference 36
Appendix 38

Introduction

Research Background

With the rapidly developing of software techniques, automation have already become a trend in recent years. Substantial numbers of IT engineers are fascinated to make computer have ability to do the same work as human. In particular, the Autonomous vehicle technology has come to be a popular research area for promoting computers being able to drive cars like human, and it seem to be achieved in not far future. However, there are still many problems in self-driving technology. One of core issues is how to make computer identify the right orbit of real road. Since computers are only able to handle numbers, they cannot understand any graph or picture of road(Anderson, 2014). Therefore, we must transform a graph road path to a mathematical way that computers could calculate and understand it. That leads to the research questions of this article: how to build mathematical model for these road route?

To solve above question, a method is designed in this research. The first step is to use high-precise GPS receiver to collect the target road coordinate data. An automobile installed with a GPS receiver will be driven on the target road. The GPS receiver would record the vehicle moving track via changing of latitude and longitude. Through these data, a local two dimension coordinate could be established, which the X axis represent west-east direction and the Y axis represent north-west direction. Next, we use MATLAB programming language to analyse the GPS data and build polynomial functions. In mathematics, a polynomial is an expression containing variables and coefficients, which include addition, subtraction, multiplication and non-negative integer exponents operations of variables. The general polynomial expression is $Y = a_n x^n + a_{n-1}\, x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0$, where $a_n \ldots a_o$constants are and x is the variable that does not have particular value. The exponent on a variable is called the degree of that variable. And, the largest degree of a variable with nonzero coefficient is termed the degree of the polynomial function(Markushevich, 2014). However, the target road is an irregular shape, which is unable to describe it by single polynomial equation. Thereby, it requires to divide the target road to different parts, which allow polynomial equations to depict the road orbit. Afterwards, the curve fitting algorithm in MATLAB can be exploited to obtain the polynomial equations for describing the corresponding road part. Then, through concatenating those equations, the math depicting of target road would be acquired. At last, we will evaluate the errors and precise of the mathematical model and discuss their goodness.

Since it is a big project, we separate the project two parts. The first part is to utilize GPS receiver to gain experimental data (longitude and latitude) for fitting and convert the KMZ file (record GPS data) to table text type. The second part is to exploit the pre-processing data to establish math model for road route. My major working scope concentrate on part2.

In-scope

- Convert longitude and latitude value to meters
- Create local 2D coordinate for target road
- Divide target road to different part
- Construct mathematical model for each part of road
- Evaluate all the polynomial models accuracy

Out-scope

- Use GPS receiver to collect data
- Convert KMZ files to txt files that record the longitude, latitude and altitude

The final result of this research is including thirty-six polynomial functions for the target road. For those mathematical model, the best case only generate 0.0977 meters errors compared to experimental GPS track data. And, the worst polynomial model produces 10.3106 meters errors. The average of 36 polynomial equations' error is 2.7498 meters. By recording hundreds of parameters for these equation, computer can draw the target road trajectory. However, the disadvantage of polynomial fitting is it cannot describe a big radian curve very well. Also, it generate some errors when these thirty-six polynomial equations are concatenated together to depict the road route.

With accomplishment of this research, we could explore a low-cost approach to establish mathematical models for road geometry. This methodology not only could assist self-driving system to identity target road but also may help digital mapping industry. Although many big companies like Google have created the most widespread digital map Google map to assist navigation of automobiles and pedestrians, there are still some rural areas are not provided accuracy digital map. Nevertheless, it is expensive to employ aerial photographs via airplane or satellite to generate digital map for rural areas. Through using GPS tracks to generate road route, many small and medium corporations have ability to make digital map for rural areas. Furthermore, according to the declaration of semiconductor corporation Broadcom in ION GNSS meeting, a new chip are being tested which is able to provide mobile device for thirty centimetres navigation. With upgradation of GPS precision, researchers can construct more and more accuracy mathematical model for the target road via GPS tracks, which may be beneficial for producing high precise digital road map.

Research Questions and Objectives

This thesis address below research questions.

Question 1: What are the basic math theory of polynomial curve fitting algorithm in MATLAB?

Question 2: What are the polynomial math models for the target road route?

Question3: How the performance of polynomial equations are evaluated for the target road?

To develop methodology for constructing road math model, the road in Springwood, Brisbane, Australia is chosen for target road. The main objectives of this project are:

Objective 1: Study surface approximation algorithms and understand its math theory.

Objective 2: Study the MATLAP programming language and their curve fitting tools for building the mathematical model of the target road route.

Objective 3: divide the GPS track data to different parts, which is able to fit appropriately to polynomial equations.

Objective 4: analyse the goodness of mathematical model and justify its benefits of this methodology.

According to above objectives, this research aims to employ the GPS track data of vehicles to gain a number of polynomial functions which can record the geometrical features of the target road. Then, justify the goodness of model and give a general methodology for constructing math model of road.

Literature Review of Previous Work

In previous work, there are many researchers have already developed different curve fitting algorithms to build mathematical model for describing dissimilar road orbit. To classify these types of issues, mathematical modelling of road route is called road alignments geometry by researchers. The reason for road alignments geometry becoming a popular problem is road traffic safety strictly relates to road geometry because the drivers' performance are strongly affected by it. The first cause is that the knowledge of road geometry could impact drivers adapt their approach of driving cars. The second reason is that road geometry can record and show the geographical features of road information which may help drivers determine their driving strategies based on road topography. In addition, with developing of Global Positioning System (GPS), more and more researchers realize that exerting GPS receiver is a low-cost way to construct mathematical model for road orbit. In 2012, Paola Di Mascio's team employ Mobile Mapping System (M.M.S.) to obtain the road alignment data. M.M.S is comprised by a GPS receiver, Inertial Navigation System and a vehicle installed above equipment. According to the data, they establish spatial coordinates x, y, z to acquire the path component and use different approximation algorithm to depict it. Paola Di Mascio's team consider the road consist of linear and circular elements. Thus, they use circumference model to represent the road curve and linear model to represent straight road. Through minimizing the sum of the squares of the distances from GPS point to circumference model, the optimized radius can be gained to solve the circumference equation(Di Mascio, Di Vito, Loprencipe, & Ragnoli, 2012). Furthermore, the optimized linear model ($y = ax + b$) is able to be gotten via using the same approach as circumference model. By employing above two math model, the straight and curve road can be depicted pretty well. However, these two models have some limitations to describe irregular road such as mountain road. These kind of road consist of different sinuous tracks, which is difficult to express by linear and circular model. Therefore, the model may fail to depict rural area road. Many sinuous path probably create high errors for these two mathematical models. Moreover, similar with Mobile Mapping System, data log vehicles has widely been utilized in gather roadway geometry information. For example, Jiménez, Aparicio, & Estrada employ the data log vehicle to collect experimental road trajectory data to develop curve-fitting algorithms for drawing digital map in order to support the Advanced Driver Assistance Systems (ADAS) in 2009(Jiménez, Aparicio, & Estrada, 2009).

In 2013, Dewang Chen's team utilise Constraint K-Segment Principal Curve Algorithms (CKPCS) to create Railway GPS digital map. The basic theory of K-segment principal curve algorithms is to divide the curves to a series of combined segments that can be portrayed by a linear model. The first step is to set two fixed endpoints as initial solution. Then, a setting error (E) is given to compare with the mistake of initial solution. If the solution error is bigger than E, a vertex is added in middle position to divide the curve to two part. All data points are separated to their corresponding vertices and segments based on the principle of nearest neighbour. Next, the sum of the two segments' linear model error (AD) is compared again to the E. If the solution error is still larger than E, another vertex will be added to optimize all vertices for minimizing the sum of errors (AD). Via repeating above processes, a proper solution where its error (AD) is smaller than E could be discovered by CKPCS algorithm. However, the elementary CKPCS algorithm has high computational complexity, which cannot work well in complex curve and huge data sets. Therefore, Dewang Chen's propose another practical CKPCS generation algorithms named Max Point Method (MPM) to decrease the complexity. This method add one step after the comparing error between initial solution and E, which is heuristically adding one vertex to generate a new solution(Chen & Chen, 2013). Subsequently, if the computed AD of is larger than K times of the setting error (E), repeat this step until AD is less than K times. Also, a heuristic approach is utilized to make the vertex locate a place where its AD is maximum among all line segments. As the MPM K-segment principal curve algorithm reduce substantial numbers of AD computing times, this algorithm are better work on large datasets and complicated curves in railway. However, because the orbit of a railway is relatively easy and straight, this improvement CKPCS algorithm is insufficient to portray very complex road trajectory. If it is applied to construct mathematical model for a complex route, lots of linear model may be generated to reach the setting error (E). As a result, it will increase the complexity of math model. Another disadvantage is the CKPCS algorithm apply many linear model to represent the trajectory, which perhaps lose the radian attribute of road orbit.

To avoid above issues, this research decide to use polynomial function to establish mathematical models for target road. There are four significant reasons to select polynomial model for this project. The first benefit is polynomial models have a simple form, which just contain several parameters. Thereby, the trajectory can be stored in computer by only several coefficients, which would save massive memory space in computer. The second advantage is that polynomial model are the most frequently used empirical models for fitting function in history. It appear in a widespread variety of areas of mathematics and science, such as chemistry, physics, economics, social science and data mining(polynomial, 2018). For instance, Peng's team utilized polynomial to depict the relationship between voltage and power in solar cell in 2013(Peng, Sun, Meng, Wang, & Xu, 2013). In 2015, Ji Chao's team propose an original mechanism for image recognition according to polynomial curve fitting(Ji, Yang, & Wang, 2015). Furthermore, in Predictive Data Mining, polynomial curve fitting is applied to establish a regression model for the continuous attributes, which is able to predict the trend of continual data(Hassan, Farhan, Mangayil, Huttunen, & Aho, 2013). Thus, most researchers have understand well on the properties of polynomial, which promote better understanding in outcome of this research. Having moderate flexibility of shapes is the third merit of polynomial models. Through changing the degree and coefficient of polynomial function, not only straight route could be represented but also the wining and tortuous trajectory could be portrayed adeptly. Furthermore, the polynomial function can clearly record the curve radian compared with K-segment principal curve algorithm. The final good point is that polynomial models are computationally simple to exert, which may economize the computation resources.

Curve fitting is the process of building a mathematical function $f(x)$ that has the best fit to a series of data points like$(x_1, y_1), (x_2, y_2), (x_2, y_2) \dots (x_n, y_n)$. Looking at the concept of curve fitting offers that the approach to get the best fit is the most critical step in curve fitting algorithms(Yang & Gordon, 2014). In the prior research, three methods had been proposed to find the best fit, including Maximum Error, Average Error and Root-mean Square. The basic principle of these approaches is to minimize the equation errors which is the distance between data points and$f(x)$. In data fitting area, the error is also called residuals, which represent the difference between an actual data points and the fitted value supplied by a mathematical model(Chatterjee, 2011).

1. **Maximum Error**

$$\boldsymbol{E_{\infty}(f(x)) = max\,|f(x_k) - y_k| \quad (1 < k < n)}$$

The maximum error is to find the maximum distance $E_\infty(f)$ between a data point (x_k, y_k) and $f(x)$. The distance is calculated by above equation. Then, try to minimize the maximum error $E_\infty(f)$ to acquire the suitable function. However, this technique is readily influenced by outlier's data point because it only focus on biggest error. If the maximum data point is far away with other data points, it could cause that the errors of other data points is raised because of the minimizing of maximum error.

2. **Average Error**

$$\boldsymbol{E_{Average}\big(f(x)\big) = \frac{1}{n}\sum_{k=1}^{n}|f(x_k) - y_k|}$$

To refrain from the outlier point impacting the fitting result, this algorithm concentrate on minimizing the average error generated by $f(x)$. Based on the above equation, $|f(x_k) - y_k|$ means the distance from the excited data point (x_k, y_k) to the math equation $f(x)$ when x is x_k. Next, let k from 1 to n to indicate all data points $(x_1, y_1), (x_2, y_2), (x_2, y_2) \dots (x_n, y_n)$.

Then, sum of the errors of all data points are gotten and divide it by the number of data points(Khartov & Zani, 2018). Finally, this equation will gain the average value of all data points' errors.

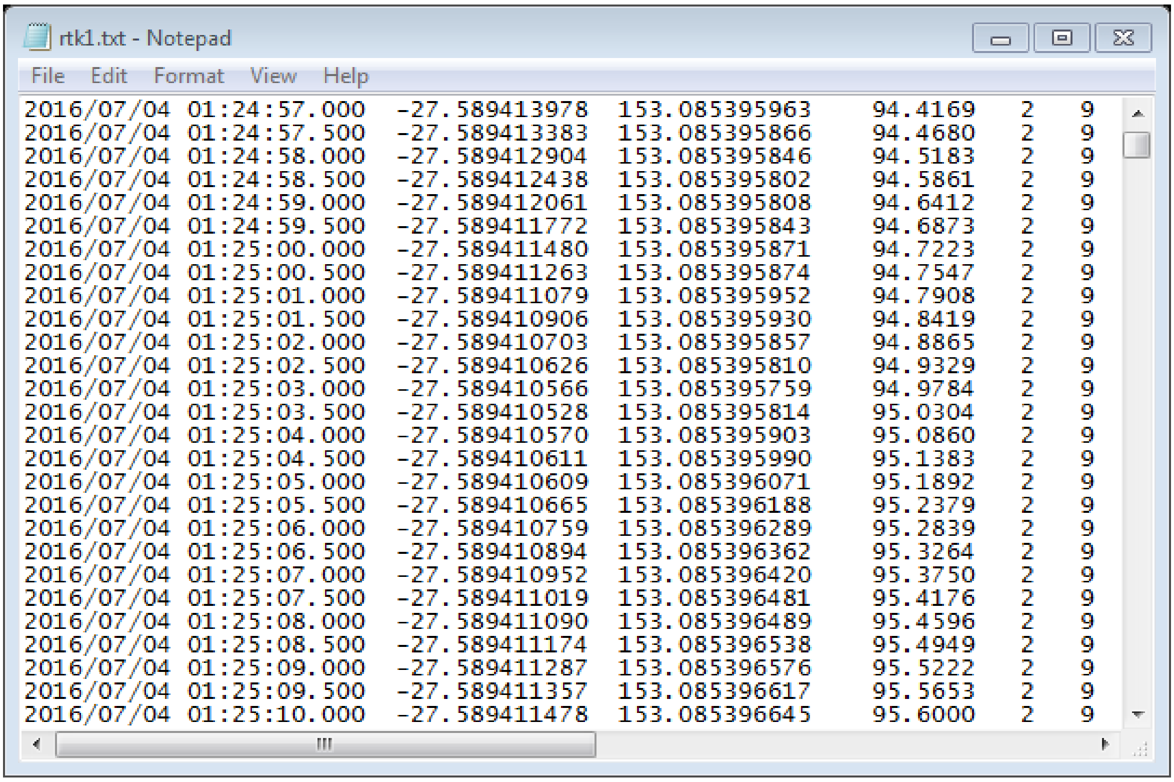
3. **Root-Mean square (Least-Squares)**

$$\boldsymbol{E_{Square}(f(x)) = \sqrt{\frac{1}{n}\sum_{k=1}^{n}|f(x_k) - y_k|^2}}$$

The approach of least squares is a standard way to approximate data points. Different with the Average Error method, it minimizes the sum of the squares of the errors generated by $f(x)$. Since the errors is squared, it may amplify the difference between observed value and fitted value given by a model(Xu, Chen, & Liang, 2018). This method are widely exploited in regression analysis, and it provide high quality in polynomial fitting result. Thereby, Root-Mean square method would be employed in this research.

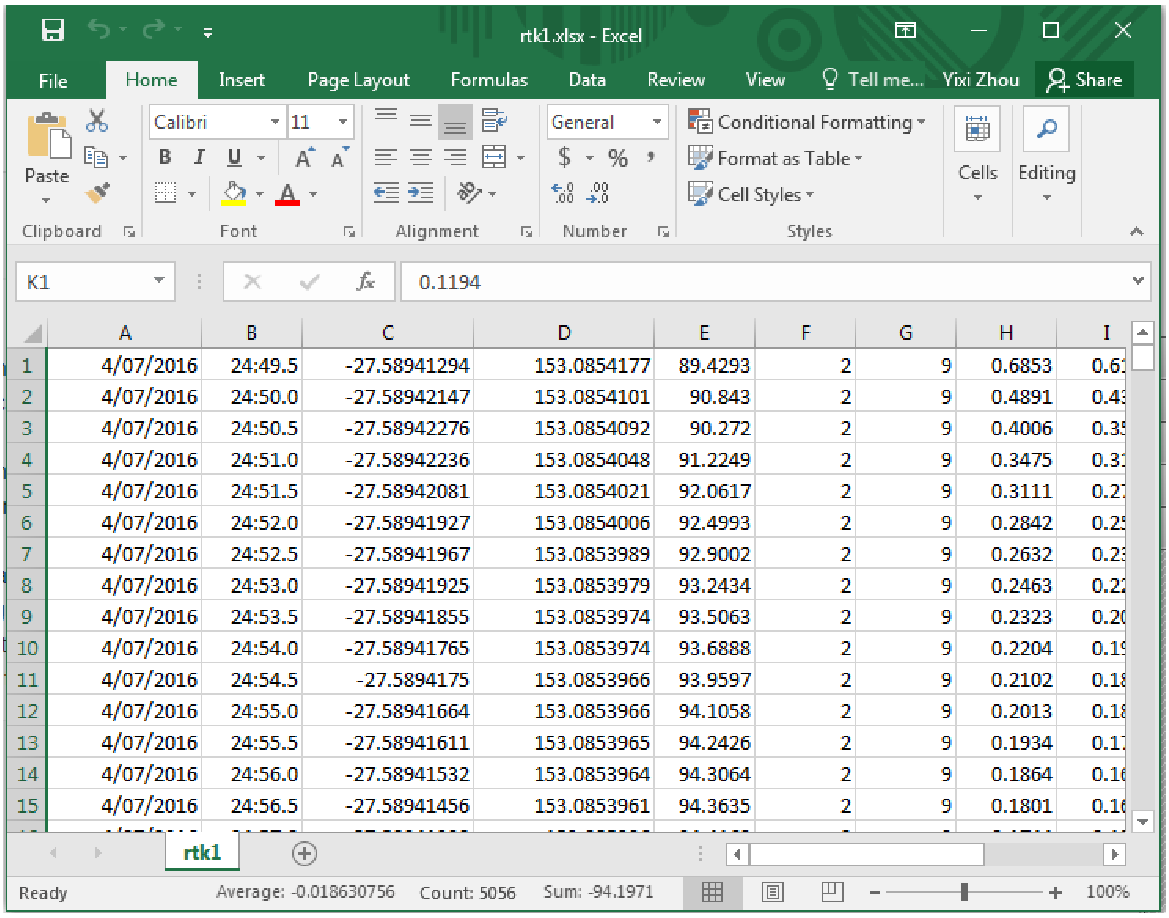
Project Methodology

There are three phases of this project, which is data collecting and pre-processing, Data analysing and modelling and Evaluation. In the data collecting and pre-processing phase, a data log vehicle installed a high-precise GPS receiver is used to collect the track data. The vehicle is driven on the target road to generate the GPS coordinate track data (latitude and longitude). After accomplishing the driving process, the GPS receiver would produce a KMZ file to record the latitude and longitude value of each time point. For making the GPS data easily be analysed by software, the KMZ file is converted to a text file showing in the below picture:



2016/07/04	01:24:57.000	-27.589413978	153.085395963	94.4169	2	9
2016/07/04	01:24:57.500	-27.589413383	153.085395866	94.4680	2	9
2016/07/04	01:24:58.000	-27.589412904	153.085395846	94.5183	2	9
2016/07/04	01:24:58.500	-27.589412438	153.085395802	94.5861	2	9
2016/07/04	01:24:59.000	-27.589412061	153.085395808	94.6412	2	9
2016/07/04	01:24:59.500	-27.589411772	153.085395843	94.6873	2	9
2016/07/04	01:25:00.000	-27.589411480	153.085395871	94.7223	2	9
2016/07/04	01:25:00.500	-27.589411263	153.085395874	94.7547	2	9
2016/07/04	01:25:01.000	-27.589411079	153.085395952	94.7908	2	9
2016/07/04	01:25:01.500	-27.589410906	153.085395930	94.8419	2	9
2016/07/04	01:25:02.000	-27.589410703	153.085395857	94.8865	2	9
2016/07/04	01:25:02.500	-27.589410626	153.085395810	94.9329	2	9
2016/07/04	01:25:03.000	-27.589410566	153.085395759	94.9784	2	9
2016/07/04	01:25:03.500	-27.589410528	153.085395814	95.0304	2	9
2016/07/04	01:25:04.000	-27.589410570	153.085395903	95.0860	2	9
2016/07/04	01:25:04.500	-27.589410611	153.085395990	95.1383	2	9
2016/07/04	01:25:05.000	-27.589410609	153.085396071	95.1892	2	9
2016/07/04	01:25:05.500	-27.589410665	153.085396188	95.2379	2	9
2016/07/04	01:25:06.000	-27.589410759	153.085396289	95.2839	2	9
2016/07/04	01:25:06.500	-27.589410894	153.085396362	95.3264	2	9
2016/07/04	01:25:07.000	-27.589410952	153.085396420	95.3750	2	9
2016/07/04	01:25:07.500	-27.589411019	153.085396481	95.4176	2	9
2016/07/04	01:25:08.000	-27.589411090	153.085396489	95.4596	2	9
2016/07/04	01:25:08.500	-27.589411174	153.085396538	95.4949	2	9
2016/07/04	01:25:09.000	-27.589411287	153.085396576	95.5222	2	9
2016/07/04	01:25:09.500	-27.589411357	153.085396617	95.5653	2	9
2016/07/04	01:25:10.000	-27.589411478	153.085396645	95.6000	2	9

The software tool for pre-processing and analysing data is MATLAB, which is a high-level technical computing language for algorithm development, numerical calculation and data visualization and analysis(Gilat, 2016). To make the data become simpler to deal with, the rtk1.txt text file is import to EXCEL software, which can transform rtk1.txt to rtk1.xlsx file.



According to above picture, the A and B column record the every time point. The C column and the D column record the corresponding latitude and longitude value for each time point. Since Excel give each column a name, it is simple to import required data to MATLAB workspace. Next, through exploiting the "xlsread( )" function in MATLAB, I can import the required data to MATLAB workspace. Because only the latitude and longitude value is needed, I just extract the data in the C and D column. The code is showing below picture:

```
latitude = xlsread('rtk1.xlsx','C1:C5056');
longitude = xlsread('rtk1.xlsx','D1:D5056');
```

Afterwards, I come to the data analysing and modelling phase of the project. The first step is to construct local coordinate system, so the first data point is set to base point 0. Furthermore, the latitude and longitude value need to be transformed to meters, because most people have concept of metric. We usually describe the distance via meters instead of degree of latitude and longitude. Therefore, I establish local coordinate, which the Y axis represent the longitude and X axis represent latitude. One degree longitude is equal to

111111 meters, so I use the second point longitude degree to subtract the base point longitude degree for acquiring the difference of two point in Y axis. Then, the third point longitude degree is subtracted the degree of base point to get the difference too. Via repeating above process until all data points longitude degree have been subtracted with base point. Next, multiply all longitude difference with 111111 to convert the degree to the meters in Y axis, like the below equations.

$$\Delta Y_1 = (\varphi_1 - \varphi_0) * 111111$$
$$\Delta Y_2 = (\varphi_2 - \varphi_0) * 111111$$
$$\Delta Y_3 = (\varphi_3 - \varphi_0) * 111111$$
$$\ldots$$
$$\Delta Y_n = (\varphi_n - \varphi_0) * 111111$$

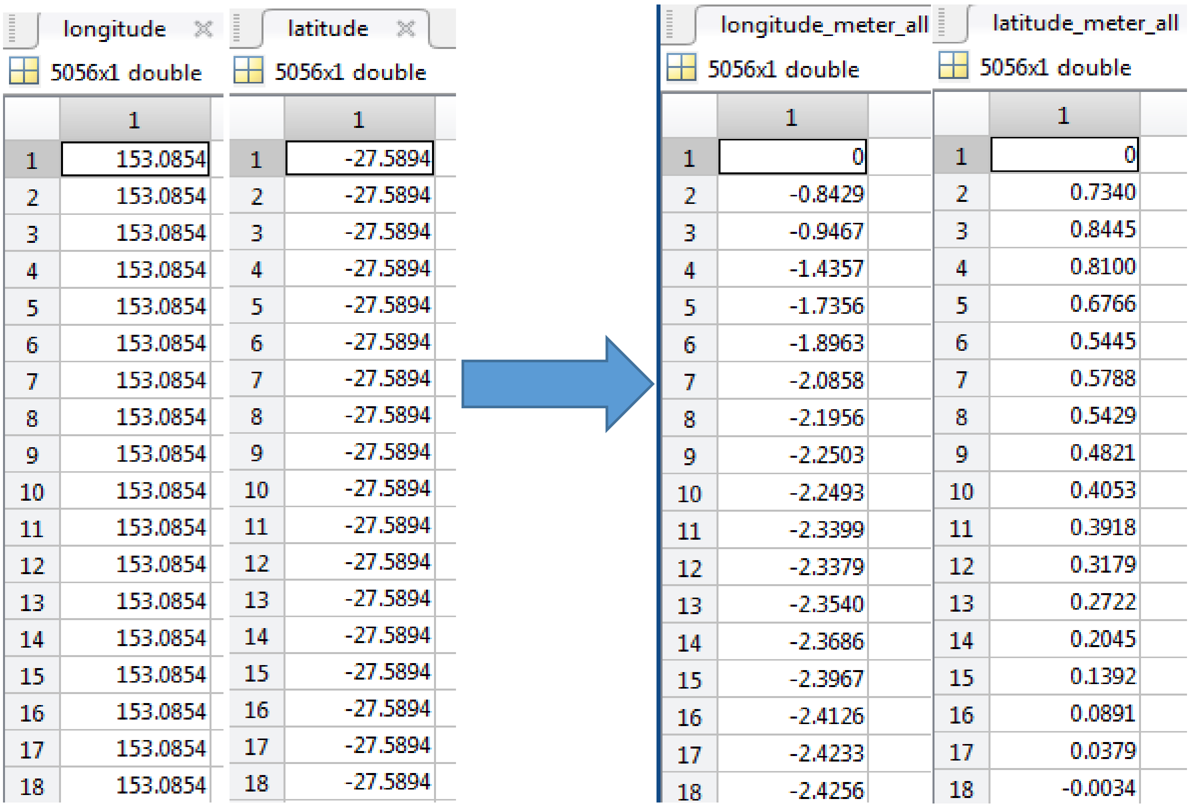
Also, one latitude degree is equal to the multiplication between 111111 and cos value of current latitude degree. Thus, same as the above progress, we can obtain the difference of latitude value in X axis. The equations is showing below:

$$\Delta X_1 = (\gamma_1 - \gamma_0) * 111111 * \cos\gamma_1$$
$$\Delta X_2 = (\gamma_2 - \gamma_0) * 111111 * \cos\gamma_2$$
$$\Delta X_3 = (\gamma_3 - \gamma_0) * 111111 * \cos\gamma_3$$
$$\ldots$$
$$\Delta X_n = (\gamma_n - \gamma_0) * 111111 * \cos\gamma_n$$

As the MATLAP provide the matrix to manipulate the data, two simple command line are able to gain the transformation results.

```
longitude_meter_all = (longitude-longitude(1)).*111111;
latitude_meter_all = (latitude - latitude(1)).*111111.*cos(latitude);
```

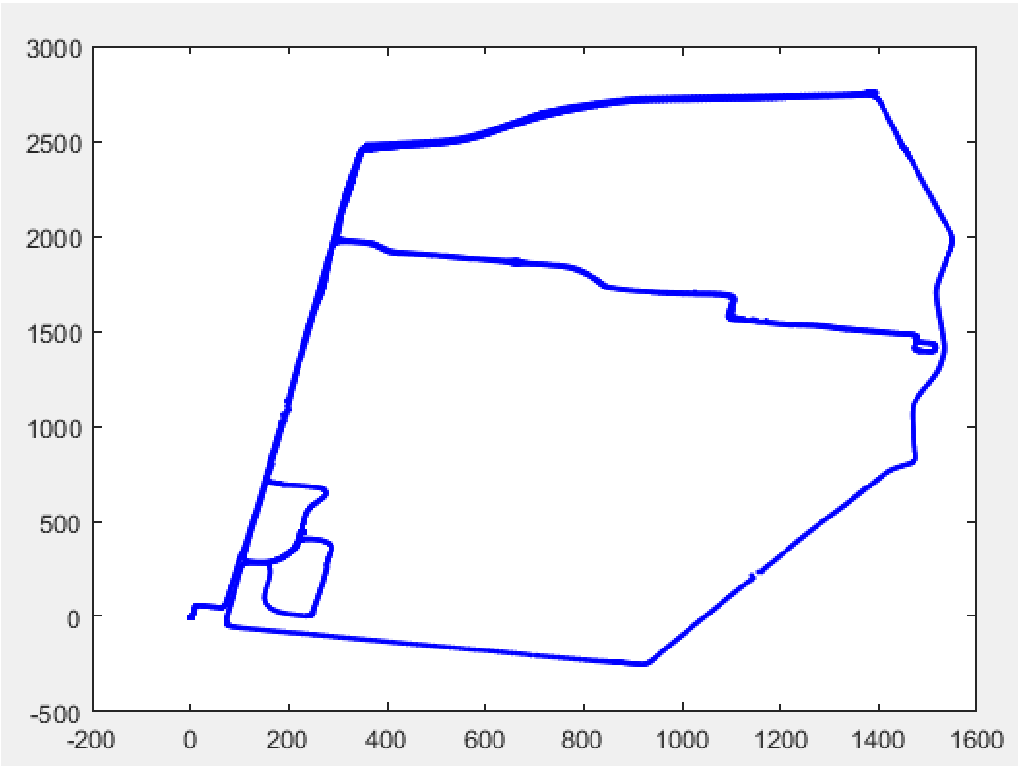
The below picture demonstrate the outcome of transformation:



	longitude (5056x1 double)	latitude (5056x1 double)	longitude_meter_all (5056x1 double)	latitude_meter_all (5056x1 double)
	1	1	1	1
1	153.0854	-27.5894	0	0
2	153.0854	-27.5894	-0.8429	0.7340
3	153.0854	-27.5894	-0.9467	0.8445
4	153.0854	-27.5894	-1.4357	0.8100
5	153.0854	-27.5894	-1.7356	0.6766
6	153.0854	-27.5894	-1.8963	0.5445
7	153.0854	-27.5894	-2.0858	0.5788
8	153.0854	-27.5894	-2.1956	0.5429
9	153.0854	-27.5894	-2.2503	0.4821
10	153.0854	-27.5894	-2.2493	0.4053
11	153.0854	-27.5894	-2.3399	0.3918
12	153.0854	-27.5894	-2.3379	0.3179
13	153.0854	-27.5894	-2.3540	0.2722
14	153.0854	-27.5894	-2.3686	0.2045
15	153.0854	-27.5894	-2.3967	0.1392
16	153.0854	-27.5894	-2.4126	0.0891
17	153.0854	-27.5894	-2.4233	0.0379
18	153.0854	-27.5894	-2.4256	-0.0034

With the transformation data, we can build a metric local coordinate to display all the data points. In MATLAP, the plot ( ) function is able to visualise all data points to demonstrate the GPS tracks, which the below graph show the command line and GPS trajectory.

```
plot(latitude_meter_all, longitude_meter_all,'.b');
```



The route shown in above graph is my target road that need to create polynomial mathematical models. Since the trajectory is irregular which cannot be depicted by a single polynomial equation, I decide to divide the path to several segments. Subsequently, I will construct the polynomial equations to describe each corresponding segment. By combining those equations together, the final mathematical model would be acquired. For promoting the each segment to have a suitable polynomial function, I separate the whole road route based on the significant inflection point. Then, I zoom out above figure to gain the X value of inflection point. Furthermore, I divide whole dataset according to the inflection point X value. For example, through zooming out the above picture, I find the X value of the first infection point is 10. Hence, I extract the data points between the base point (0, 0) and the infection point where their X value is from 0 to 10. Next, store these data to a cell type variable named “latitude_meters”. The cell variable structure can save the heterogeneous data, which allow me to store different scale matrix. This MATLAB command line show below picture:

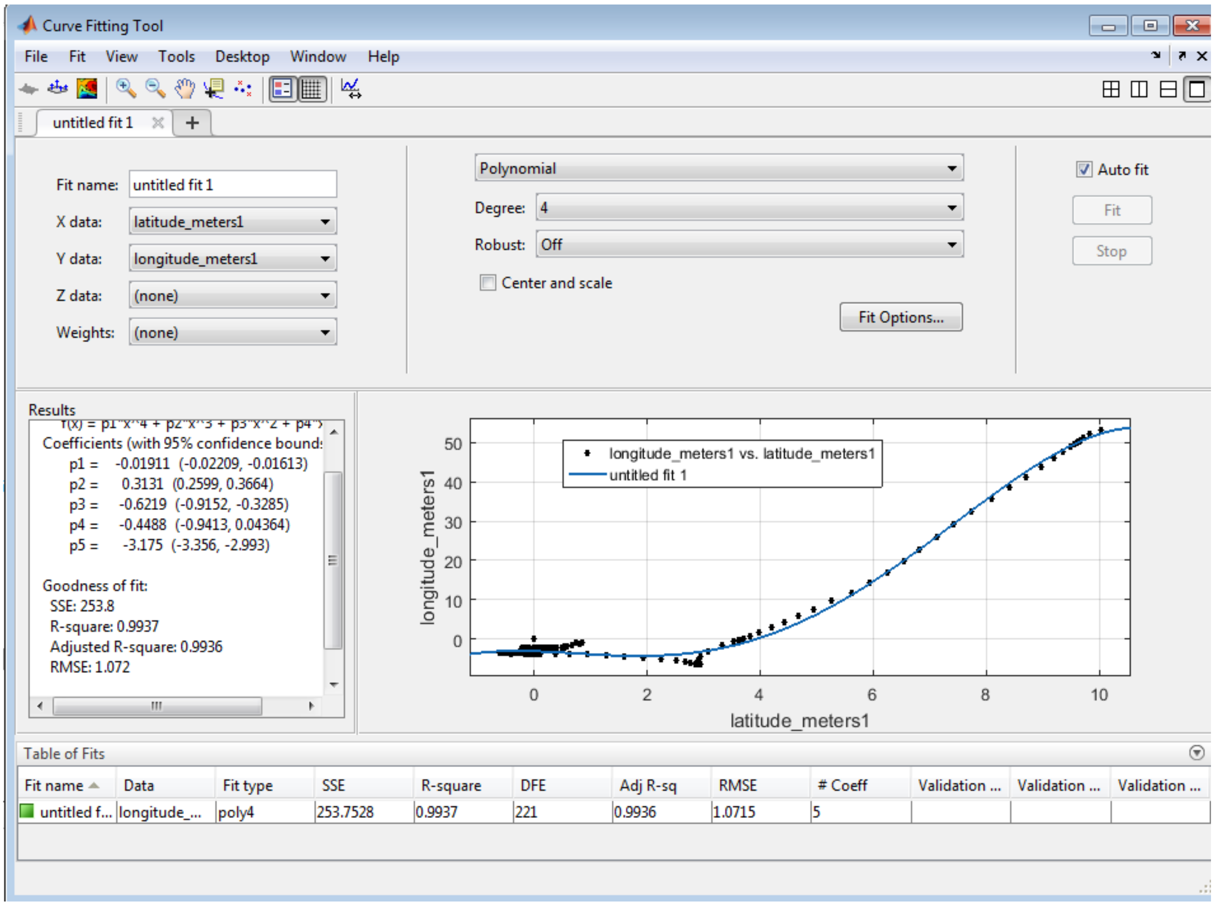
```
latitude_meters{1} = latitude_meter_all(1:226);
longitude_meters{1} = longitude_meter_all(1:226);
```

After getting the segment dataset, I utilize the curve fitting tool (showing below picture) offered by MATLAB to determine the most suitable polynomial function. By changing the degree of polynomial function, I can minimize the error of model. However, if the degree is too high, the polynomial would become more complex. Thus, I need to adjust the degree to build a proper polynomial model. When the polynomial degree is determined, I employ the below command line to construct the mathematical model.

```
[xData, yData] = prepareCurveData( latitude_meters{1}, longitude_meters{1});
ft = fittype( 'poly4' );
[fitresult{1}, gof(1)] = fit( xData, yData, ft );
```

In addition, the polynomial model is saved to a cell type variable named “fitresult” and the model’s goodness information is stored in a struct type variable named “gof”.

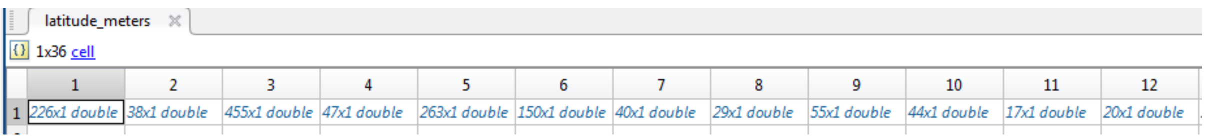
Curve fitting tool

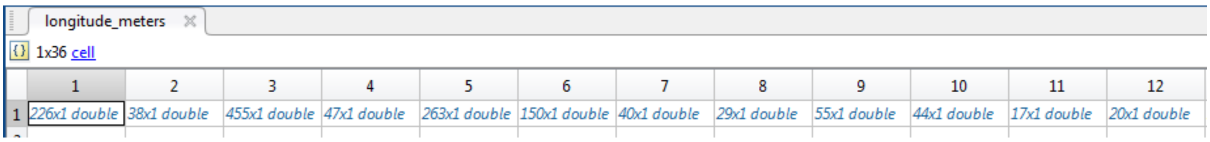


The first polynomial model had been established via above procedure. Afterwards, the second infection point is able to be found by repeating the zoom out method. As same as the foregoing approach, the second polynomial model could be built by under instructions

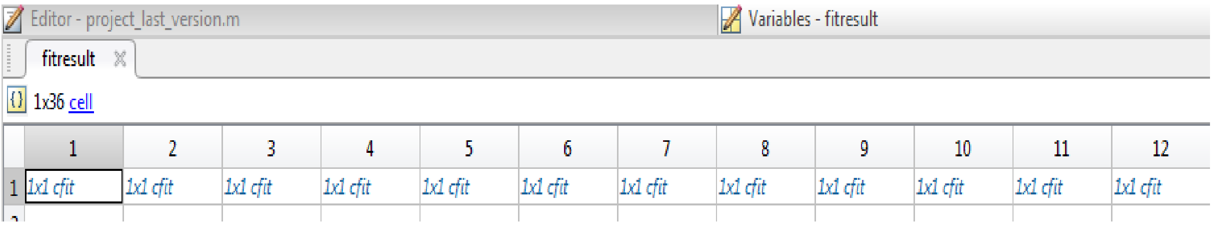
```
latitude_meters{2} = latitude_meter_all(226:263);
longitude_meters{2} = longitude_meter_all(226:263);
[xData, yData] =
prepareCurveData( latitude_meters{2},longitude_meters{2});
ft = fittype( 'poly5' );
[fitresult{2}, gof(2)] = fit(xData, yData, ft);
```

With adopting the forgoing approach, I separate whole dataset to 36 segments and store in cell type variable (display in under picture).





Then, 36 polynomial functions are built via above datasets, and it is saved to a cell type variable displaying in under picture.



Since every polynomial function work on a specific range of X value, the "`linspace` ( )" function is employed to record the X value range of each model. The code is showing below:

```
x{1} = linspace(0,10);
x{2} = linspace(10,66);
x{3} = linspace(66, 351);
…
```

Moreover, as the X value is known, it is simple to use the above polynomial model to draw the orbit of target road. The code is showing under position:

```
y{1} = fitresult{1}(x{1});
y{2} = fitresult{2}(x{2});
y{3} = fitresult{3}(x{3});
…
```

To combine 36 polynomial function for describing the target road trajectory, the hold on/off command line is used in MATALB, which is demonstrated on under picture.

```
hold on
plot(latitude_meter_all, longitude_meter_all,'.b');
plot(x{1},y{1},'-r',x{2},y{2},'-r',x{3},y{3},'-r',x{4},y{4},'-r',x{5},y{5},'-r',x{6},y{6},'-r');
plot(x{7},y{7},'-r',x{8},y{8},'-r',x{9},y{9},'-r',x{10},y{10},'-r',x{11},y{11},'-r',x{12},y{12},'-r');
plot(x{13},y{13},'-r',x{14},y{14},'-r',x{15},y{15},'-r',x{16},y{16},'-r',x{17},y{17},'-r',x{18},y{18},'-r');
plot(x{19},y{19},'-r',x{20},y{20},'-r',x{21},y{21},'-r',x{22},y{22},'-r',x{23},y{23},'-r',x{24},y{24},'-r');
plot(x{25},y{25},'-r',x{26},y{26},'-r',x{27},y{27},'-r',x{28},y{28},'-r',x{29},y{29},'-r',x{30},y{30},'-r');
plot(x{31},y{31},'-r',x{32},y{32},'-r',x{33},y{33},'-r',x{34},y{34},'-r',x{35},y{35},'-r',x{36},y{36},'-r');
hold off
```

After finishing building the polynomial models of target road, the second phase of project was accomplished. Next, we come to the third phase of research that is evaluation. The main way to evaluate the mathematical model performance is to justify the errors of these polynomial equations. The related information is stored in struct type variable named "gof".

I can access the information via the MATLAB Workspace, so I would evaluate model accuracy based on the below data.

gof

1x36 struct with 5 fields

Fields	sse	rsquare	dfe	adjrsquare	rmse
1	253.7528	0.9937	221	0.9936	1.0715
2	1.5474	0.9967	32	0.9962	0.2199
3	2.2871e+04	0.9999	451	0.9999	7.1212
4	49.4440	0.9974	43	0.9972	1.0723
5	398.9270	0.9995	258	0.9995	1.2435
6	764.2968	0.9990	144	0.9990	2.3038
7	193.9454	0.9973	36	0.9970	2.3211
8	14.3885	0.9954	24	0.9946	0.7743
9	2.1156e+03	0.9965	50	0.9962	6.5048
10	265.3677	0.9918	37	0.9905	2.6781
11	22.0205	0.9991	13	0.9989	1.3015
12	55.2558	0.9909	16	0.9892	1.8584
13	73.4793	0.9995	27	0.9994	1.6497
14	187.6205	0.9998	253	0.9998	0.8612
15	2.8047e+03	0.9998	141	0.9998	4.4600
16	99.7993	0.9966	20	0.9954	2.2338
17	456.3222	0.9955	26	0.9951	4.1894
18	3.1919e+03	0.9946	42	0.9940	8.7176
19	458.6689	0.9977	27	0.9976	4.1216

Outcomes

Research Question 1

Through using the MATLAB command line, the polynomial mathematical model can be easily obtained. However, the behind math theory is hidden in the MATLAB programming language, so how the curve fitting algorithm to work for getting suitable polynomial. This is the first research question: What are the basic math theory of polynomial curve fitting algorithm in MATLAB?

According to the literature review, we have already known the basic principle is minimize the errors. The least-squares method is chosen to get proper math model, so I assume the required polynomial is $f(x) = a_n x^n + a_{n-1}\, x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0$. The least-squares equation demonstrated in below equation.

$$\boldsymbol{E_{Square}(f(x)) = \sqrt{\frac{1}{n}\sum_{k=1}^{n} |f(x_k) - y_k|^2}}$$

Since $\frac{1}{n}$ is a constant variable which is decide by the number of data point, we do not need to consider about it. Also, the root operation cannot impact on minimizing the errors, so we could ignore it either. To sum up, the real part in above equation is required to consider is shown in under.

$$\boldsymbol{E = \sum_{k=1}^{n} |f(x_k) - y_k|^2}$$

Therefore, the smallest $\boldsymbol{E}$ should be acquired. Then, we replace $f(x_k)$ to $a_n x^n + a_{n-1}\, x^{n-1} \ + \cdots + a_2 x^2 + a_1 x + a_0$, so we get:

$$\boldsymbol{E = \sum_{k=1}^{n} |\ a_n x_k{}^n + a_{n-1}\, x_k{}^{n-1} \ + \cdots + a_2 x_k{}^2 + a_1 x_k + a_0 - y_k|^2}$$

If we want to minimize E, the calculus approach should be employed to analyse it. In calculus given a function Y = f (X), if the derivative of a real variable X is equal to 0, it means the Y values reach its smallest or maximum value. However, there is no maximum value of errors in curve fitting algorithms, because the error could be an infinite value if the data point is infinite far apart from $f(x) = a_n x^n + a_{n-1}\, x^{n-1} \ + \cdots + a_2 x^2 + a_1 x + a_0$.Thereby, if we make the derivative of equation $|\ a_n x_k{}^n + a_{n-1}\, x_k{}^{n-1} \ + \cdots + a_2 x_k{}^2 + a_1 x_k + a_0 - y_k|^2$ become 0, it can only give a smallest value that also is the minimum error. In this case, $x_k{}^n, x_k{}^{n-1}$ ... $x_k{}^2, x_k\ and\ y_k$ is the constant value, which is the real data points we gotten. Hence, we can only manipulate a_n, a_{n-1} ... a_2, a_1, a_0 for promoting its derivative to become 0. To set the derivative turn to 0, the under differential equations are constructed.

$$\boldsymbol{\frac{dE}{da_n} = 0 \qquad \frac{dE}{da_{n-1}} = 0 \qquad \frac{dE}{da_{n-2}} = 0}$$

...

$$\boldsymbol{\frac{dE}{da_2} = 0 \qquad \frac{dE}{da_1} = 0 \qquad \frac{dE}{da_0} = 0}$$

Next, we need to solve the forgoing differential equation, which demonstrate under.

$$\boldsymbol{\frac{dE}{da_n} = 0 : \sum_{k=1}^{n} 2(a_n x_k{}^n + a_{n-1}\, x_k{}^{n-1} \ + \cdots + a_2 x_k{}^2 + a_1 x_k + a_0 - y_k) * x_k = 0}$$

$$\boldsymbol{\frac{dE}{da_{n-1}} = 0 : \sum_{k=1}^{n} 2(a_n x_k{}^n + a_{n-1}\, x_k{}^{n-1} \ + \cdots + a_2 x_k{}^2 + a_1 x_k + a_0 - y_k) * x_k = 0}$$

$$\frac{dE}{da_{n-2}} = 0 : \sum_{k=1}^{n} 2(a_n x_k{}^n + a_{n-1}\, x_k{}^{n-1} + \cdots + a_2 x_k{}^2 + a_1 x_k + a_0 - y_k) * x_k = 0$$

…

$$\frac{dE}{da_2} = 0 : \sum_{k=1}^{n} 2(a_n x_k{}^n + a_{n-1}\, x_k{}^{n-1} + \cdots + a_2 x_k{}^2 + a_1 x_k + a_0 - y_k) * x_k = 0$$

$$\frac{dE}{da_1} = 0 : \sum_{k=1}^{n} 2(a_n x_k{}^n + a_{n-1}\, x_k{}^{n-1} + \cdots + a_2 x_k{}^2 + a_1 x_k + a_0 - y_k) * x_k = 0$$

$$\frac{dE}{da_0} = 0 : \sum_{k=1}^{n} 2(a_n x_k{}^n + a_{n-1}\, x_k{}^{n-1} + \cdots + a_2 x_k{}^2 + a_1 x_k + a_0 - y_k) * 1 = 0$$

Then, we try to simplify above equations and gain a multivariate system of equations.

$$\sum_{k=1}^{n} (a_n x_k{}^{n+1} + a_{n-1}\, x_k{}^n + \cdots + a_2 x_k{}^3 + a_1 x_k{}^2 + a_0 x_k - y_k x_k) = 0$$

$$\sum_{k=1}^{n} (a_n x_k{}^{n+1} + a_{n-1}\, x_k{}^n + \cdots + a_2 x_k{}^3 + a_1 x_k{}^2 + a_0 x_k - y_k x_k) = 0$$

…

$$\sum_{k=1}^{n} (a_n x_k{}^{n+1} + a_{n-1}\, x_k{}^n + \cdots + a_2 x_k{}^3 + a_1 x_k{}^2 + a_0 x_k - y_k x_k) = 0$$

$$\sum_{k=1}^{n} (a_n x_k{}^n + a_{n-1}\, x_k{}^{n-1} + \cdots + a_2 x_k{}^2 + a_1 x_k + a_0 - y_k) = 0$$

According to above equations, we can build N dimension matrix equations to solve the value of $a_n, a_{n-1} \dots\ a_2, a_1, a_0$, which show in below.

$$\begin{pmatrix} \sum_{k=1}^{n} x_k{}^{n+1} & \sum_{k=1}^{n} x_k{}^{n} & \dots \sum_{k=1}^{n} x_k{}^{2} & \sum_{k=1}^{n} x_k \\ \sum_{k=1}^{n} x_k{}^{n+1} & \sum_{k=1}^{n} x_k{}^{n} & \dots \sum_{k=1}^{n} x_k{}^{2} & \sum_{k=1}^{n} x_k \\ \vdots & & \vdots & \vdots \\ \sum_{k=1}^{n} x_k{}^{n+1} & \sum_{k=1}^{n} x_k{}^{n} & \dots \sum_{k=1}^{n} x_k{}^{2} & \sum_{k=1}^{n} x_k \\ \sum_{k=1}^{n} x_k{}^{n} & \sum_{k=1}^{n} x_k{}^{n-1} & \cdots \sum_{k=1}^{n} x_k & \sum_{k=1}^{n} a_0 \end{pmatrix} \begin{pmatrix} a_n \\ a_{n-1} \\ \vdots \\ a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^{n} y_k x_k \\ \sum_{k=1}^{n} y_k x_k \\ \vdots \\ \sum_{k=1}^{n} y_k x_k \\ \sum_{k=1}^{n} y_k \end{pmatrix}$$

In conclusion, MATLAB do all above work for us, so we can utilize simple command line to establish the polynomial model for the target road. Nevertheless, it still important to know the principle of polynomial curve fitting algorithms. The least-squares method not only has used in polynomial curve fitting but also exerted in other curve fitting algorithms. In particular, the mechanism of making derivative be equal to 0 has become the major technique to search the best fitting function.

Research Question 2

There are 36 polynomial functions built by MATLAB. Each polynomial equation represent a specific trajectory in different range of X value. The route length depicted by these equation is metric. The under 36 functions are the mathematical of whole target road.

$\{0 \le x \le 10\}$

$y_1 = -0.01911x^4 + 0.3131x^3 - 0.6219x^2 - 0.4488\,x - 3.175$

When the x is from 0 to 10, the first part of road trajectory is portrayed by y_1 function. Moreover, the below function $y_2, y_3, y_4 \ldots y_{36}$ depict the road orbits in different range of x value. The X axis represent the West- East direction, and the Y axis represent North-South direction.

$\{10 \le x \le 66\}$

$y_2 = 6.471 * 10^{-7}x^5 - 0.0001296x^4 + 0.01006\,x^3 - 0.3746\,x^2 + 6.356x + 19.12$

$\{66 \le x \le 351\}$

$y_3 = -3.492 * 10^{-5}\,x^3 + 0.02368x^2 + 3.989x - 316.5$

$\{351 \le x \le 601\}$

$y_4 = 1.012 * 10^{-5}x^3 - 0.01352x^2 + 6.129x + 1545$

$\{601 \le x \le 1384.5\}$

$y_5 = -7.674 * 10^{-10}\,x^4 + 4.356 * 10^{-6}x^3 - 0.008831x^2 + 7.741x + 228.3$

$\{1384.5 \ge x \ge 550\}$

$$y_6 = -5.095e * 10^{-12}x^5 + 2.542 * 10^{-8}x^4 - 4.851 * 10^{-5}\,x^3 + 0.04349x^2 - 17.63x + 5039$$

$\{115 \le x \le 220\}$

$y_7 = 0.0001675\,x^3 - 0.06467x^2 + 8.158\,x - 48.18$

$\{220 \le x \le 286\}$

$y_8 = -1.002 * 10^{-5}\,x^4 + 0.009908\,x^3 - 3.674\,x^2 + 605.5x - 3.701 * 10^4$

$\{286 \ge x \ge 250\}$

$y_9 = -0.0003172x^4 + 0.3397\,x^3 - 136.2\,x^2 + 2.427 * 10^4x - 1.62e * 10^6$

$\{250 \ge x \ge 150\}$

$$y_{10} = 7.392 * 10^{-9}x^6 - 8.918 * 10^{-6}x^5 + 0.004464\, x^4 - 1.187x^3 + 176.8\, x^2 - 1.399 * 10^4\, x + 4.596 * 10^5$$

$\{150 \leq x \leq 162.5\}$

$$y_{11} = 0.04295x^3 - 20.22x^2 + 3181x - 1.671 * 10^5$$

$\{120 \geq x \geq 100\}$

$$y_{12} = 0.01854x^3 - 6.385x^2 + 733.1x - 2.777 * 10^4$$

$\{100 \geq x \geq 75\}$

$$y_{13} = 8.683\, x - 647.1$$

$\{75 \leq x \leq 919.5\}$

$$y_{14} = 2.071 * 10^{-5}x^2 - 0.2572x - 32.9$$

$\{919.5 \leq x \leq 1400\}$

$$y_{15} = 2.071x - 2170$$

$\{1400 \leq x \leq 1474.5\}$

$$y_{16} = 7.998 * 10^{-10}x^7 - 8.027 * 10^{-6}x^6 + 0.03452\, x^5 - 82.48x^4 + 1.182 * 10^5x^3 - 1.017 * 10^8x^2 + 4.858 * 10^{10}x - 9.946 * 10^{12}$$

$\{1474.5 \geq x \geq 1470.5\}$

$$y_{17} = 4.309x^2 - 1.274 * 10^4x + 9.411 * 10^6$$

$\{1470.5 \leq x \leq 1534\}$

$$y_{18} = 4.949 * 10^{-6}\, x^5 - 0.03717x^4 + 111.7\, x^3 - 1.678 * 10^5\, x^2 + 1.26 * 10^8x - 3.786 * 10^{10}$$

$\{1534 \geq x \geq 1518\}$

$$y_{19} = -15.75\, x + 2.559 * 10^4$$

$\{1518 \leq x \leq 1551\}$

$$y_{20} = 4.228 * 10^{-5}x^5 - 0.3246x^4 + 996.8\, x^3 - 1.531 * 10^6x^2 + 1.175 * 10^9x - 3.608 * 10^{11}$$

$\{1551 \geq x \geq 1400\}$

$$y_{21} = -4.773x + 9399$$

$\{300 \leq x \leq 399\}$

$$y_{22} = 3.056 * 10^{-6}x^4 - 0.004498x^3 + 2.466\, x^2 - 597.6x + 5.601 * 10^4$$

$\{399 \leq x \leq 759\}$

$$y_{23} = -4.135 * 10^{-11}x^5 + 1.248 * 10^{-7}x^4 - 0.0001495\, x^3 + 0.08875x^2 - 26.29\, x + 5040$$

$\{759 \leq x \leq 849.5\}$

$$y_{24} = -0.009344x^2 + 13.77x - 3228$$

$$\{849.5 \leq x \leq 1107\}$$

$$y_{25} = -1.108 * 10^{-13}x^7 + 7.545 * 10^{-10}x^6 - 2.2 * 10^{-6}x^5 + 0.003558\, x^4 - 3.45x^3 + 2005\, x^2 - 6.464 * 10^5x + 8.923 * 10^7$$

$$\{1102 \leq x \leq 1108\}$$

$$y_{26} = 14.79x - 1.471 * 10^4$$

$$\{1102 \leq x \leq 1480\}$$

$$y_{27} = -1.7 * 10^{-10}x^5 + 1.113 * 10^{-6}x^4 - 0.002911x^3 + 3.797x^2 - 2471\, x + 6.438 * 10^5$$

$$\{1479.9 \geq x \geq 1478.9\}$$

$$y_{28} = 22.81\, x^3 - 1.012 * 10^5x^2 + 1.497 * 10^8x - 7.384 * 10^{10}$$

$$\{1478.9 \geq x \geq 1494.1\}$$

$$y_{29} = -0.0003317\, x^5 + 2.467x^4 - 7339\, x^3 + 1.092 * 10^7x^2 - 8.118 * 10^9x + 2.415 * 10^{12}$$

$$\{1494.1 \leq x \leq 1518\}$$

$$y_{30} = -0.0002118x^4 + 1.272\, x^3 - 2865x^2 + 2.868 * 10^6x - 1.076 * 10^9$$

$$\{1518 \geq x \geq 1506.5\}$$

$$y_{31} = 0.4313x^2 - 1302x + 9.835 * 10^5$$

$$\{1506.5 \geq x \geq 1479.8\}$$

$$y_{32} = 0.04349x^2 - 7.272x + 6992$$

$$\{1479.8 \geq x \geq 1473.1\}$$

$$y_{33} = -0.07291x^3 + 323.3\, x^2 - 4.777 * 10^5x + 2.353 * 10^8$$

$$\{1473.1 \leq x \leq 1477.4\}$$

$$y_{34} = -0.598x^2 + 1777\, x - 1.319 * 10^6$$

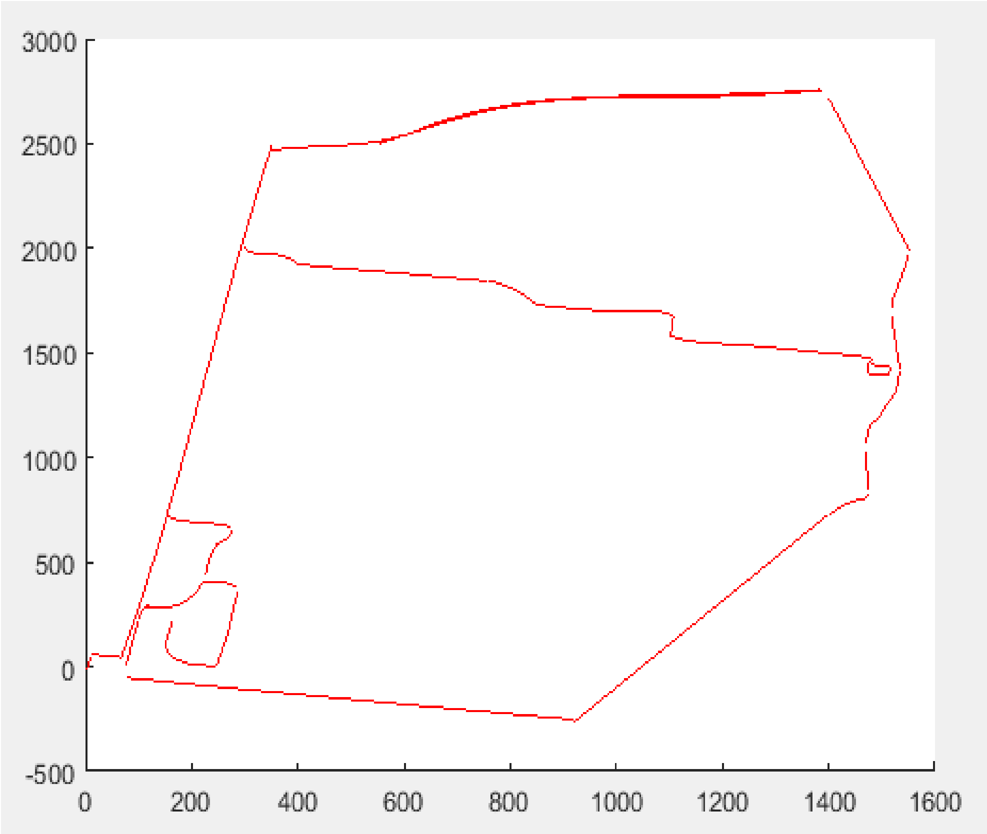
$$\{154 \leq x \leq 275\}$$

$$y_{35} = -4.636 * 10^{-8}x^5 + 4.917 * 10^{-5}x^4 - 0.02076x^3 + 4.364x^2 - 456.9x + 1.979 * 10^4$$

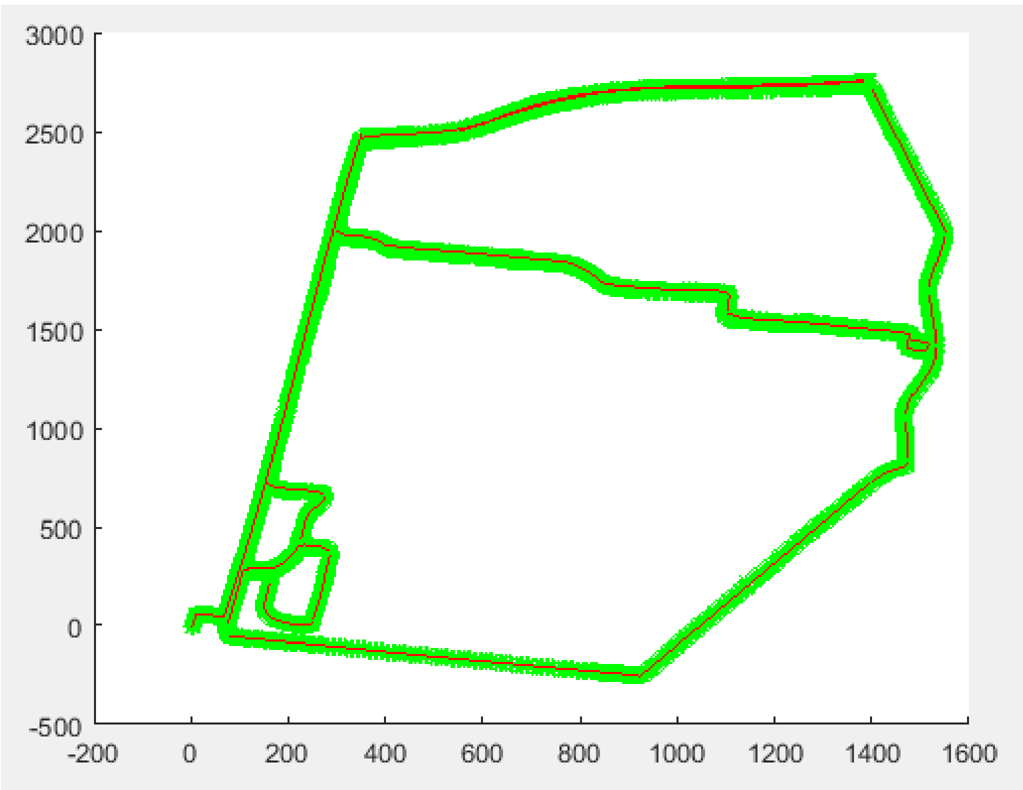
$$\{275 \geq x \geq 226\}$$

$$y_{36} = 4.316 * 10^{-5}x^4 - 0.03922\, x^3 + 13.12x^2 - 1899x + 9.992 * 10^4$$

Based on the above 36 polynomial function, the target road trajectory is able to be described in local coordinate system, which the orbit is shown in under figure.



Compared to the GPS tracks, this polynomial mathematical model fit pretty well with the target road, which is demonstrated in below figure.



The green line is GPS track and the red line is mathematical model of target road.

Research Question 3

After finishing the modelling process, we need to evaluate the performance of these polynomial approximating models. The main method to justify these models is to check the errors of each polynomial function. In MATLAB, the error information is saved in “rmse” field of “gof” struct. Through below MATLAB programming, the worst case can be obtained.

```
max = 0;
for n = 1:36
    if gof(n).rmse > max
        max = gof(n).rmse;
        number_max = n;
    else
        continue;
    end
end
disp('the worst fitresult');
disp(number_max);
disp('the worst error is:');
disp(max);
```

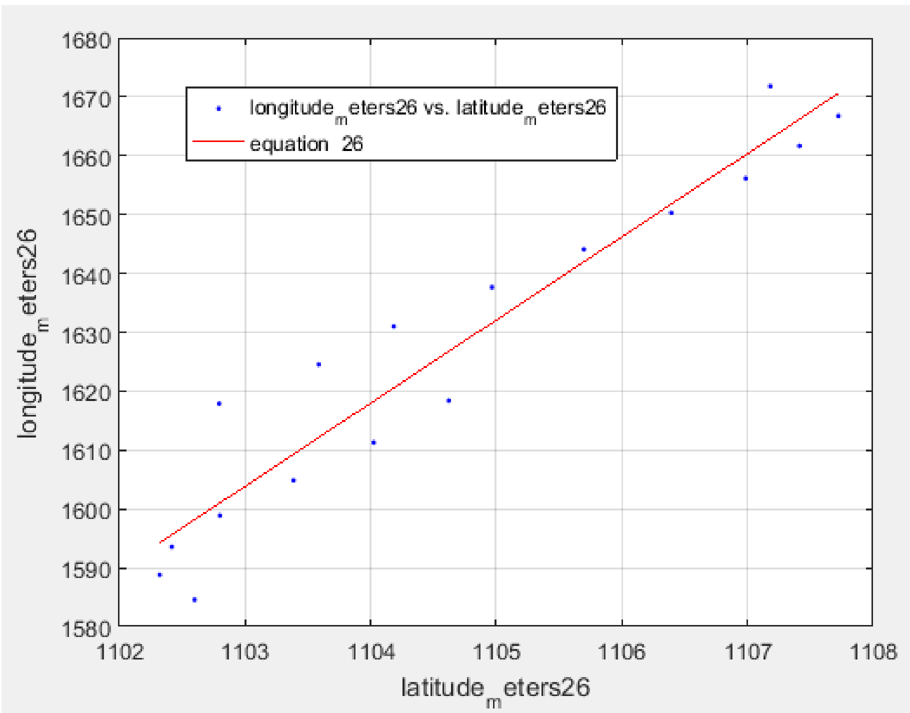
```
the worst fitresult
    26

the worst error is:
   10.3106
```

According the result, we can know the worst model is 26th polynomial, which is y_{26}. When x is from 1102 to 1108, y_{26} causes 10.3106 meters error compared with the GPS track. The below figure show the polynomial function (the blue data points is GPS data and the red line is polynomial model).

$$\{1102 \le x \le 1108\}$$

$$y_{26} = 14.79x - 1.471 * 10^4$$



Furthermore, the best case of these 36 polynomial model can be acquired by bellow MATLAB programming.

```
dim = 100;
for n = 1:36
     if gof(n).rmse < dim
         dim = gof(n).rmse;
         number_min = n;
     else
         continue;
     end
end
disp('the best fitresult');
disp(number_min);
disp('the best error is :');
disp (dim);
```

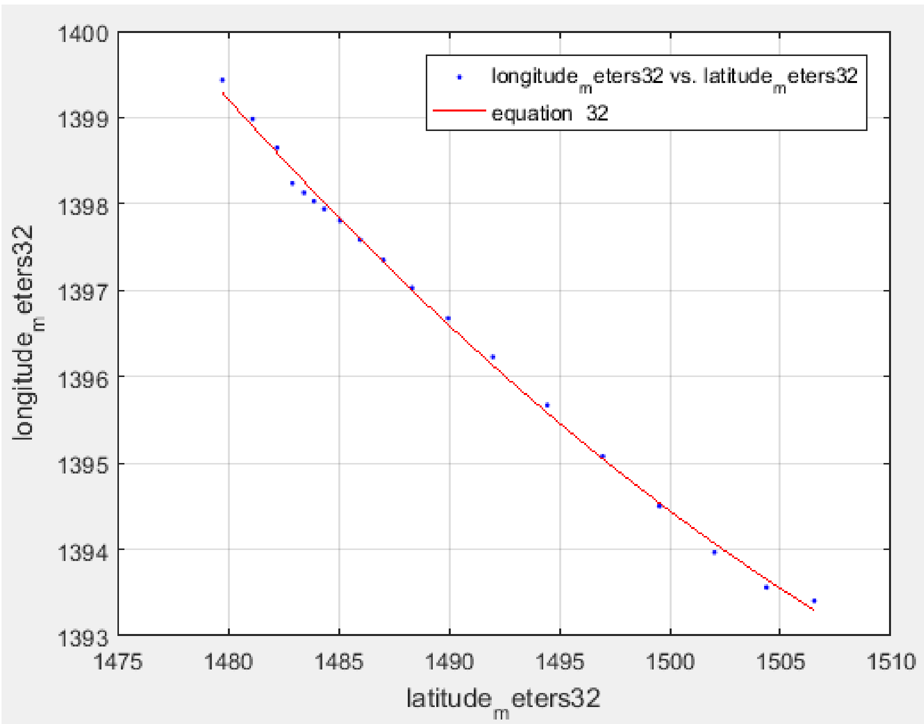
```
the best fitresult
    32

the best error is :
    0.0977
```

According to the above result, the best model is 32th polynomial equation y_{32}. When x is from 1479.8 to 1506.5, y_{32} causes 0.0977 meters error compared to the GPS track.

$$\{1506.5 \geq x \geq 1479.8\}$$

$$y_{32} = 0.04349x^2 - 7.272x + 6992$$



In all 36 polynomial function, there are seven types of polynomial equation are used to portray the orbit of the target road. The below table demonstrate the type of each polynomial function:

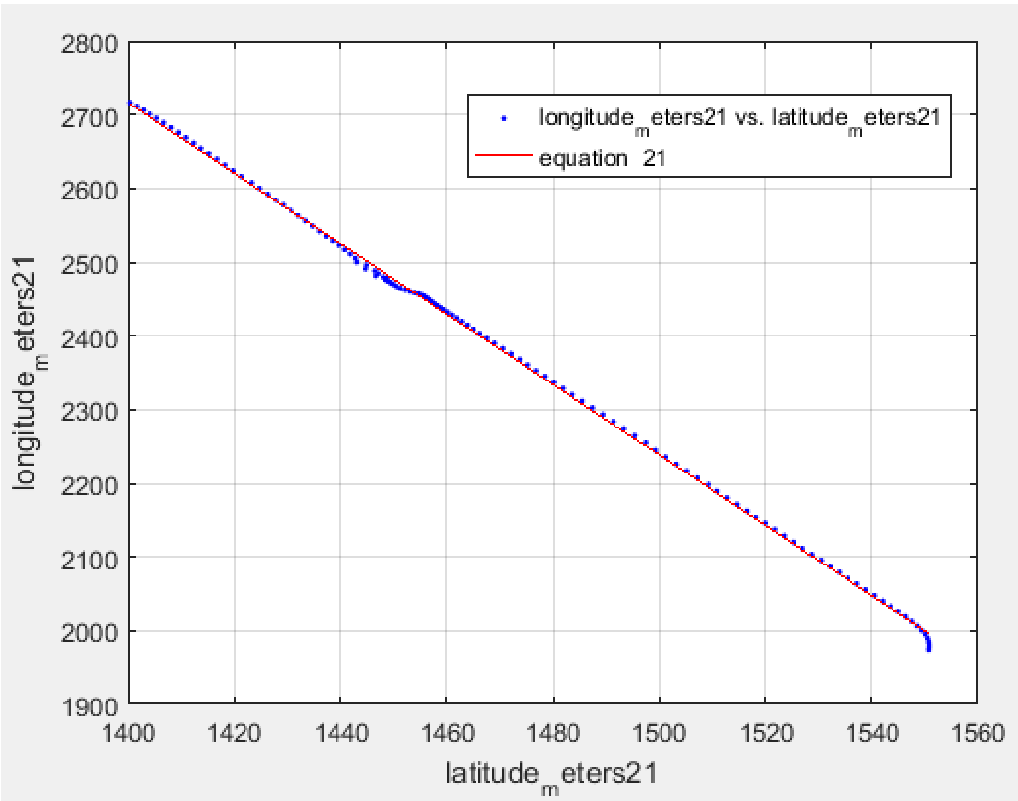
Polynomial type (degree)	Number of polynomial equation
Ploy 1	$y_{13}, y_{15}, y_{19}, y_{21}, y_{26}$
Ploy 2	$y_{14}, y_{17}, y_{24}, y_{31}, y_{32}, y_{34}$
Ploy 3	$y_3, y_4, y_7, y_{11}, y_{12}, y_{28}, y_{30}, y_{33}$
Ploy 4	$y_1, y_5, y_8, y_9, y_{22}$
Poly 5	$y_2, y_6, y_{18}, y_{20}, y_{23}, y_{27,} y_{29}, y_{35}, y_{36}$
Poly 6	y_{10}
Poly 7	y_{16}, y_{25}

We would exhibit seven examples to demonstrate each type of polynomial function and their accuracy.

Ploy 1 type

When X is from 1551 meters to 1400 meters, y_{21} causes 5.2493 meters error compared to the GPS track (the blue data points is GPS data and the red line is polynomial model).

$$\{1551 \geq x \geq 1400\}$$
$$y_{21} = -4.773x + 9399$$

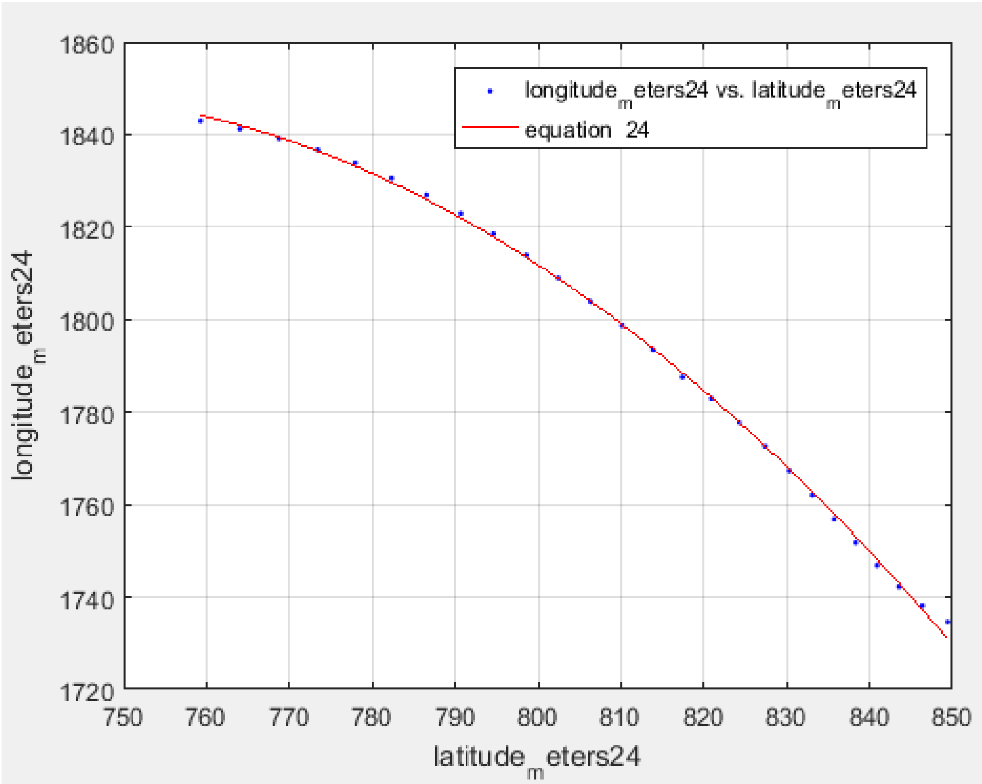


Ploy 2 type

When X is from 759meters to 849.5meters, y_{24} causes 1.0847 meters error compared to the GPS track (the blue data points is GPS data and the red line is polynomial model).

$$\{759 \leq x \leq 849.5\}$$

$$y_{24} = -0.009344x^2 + 13.77x - 3228$$

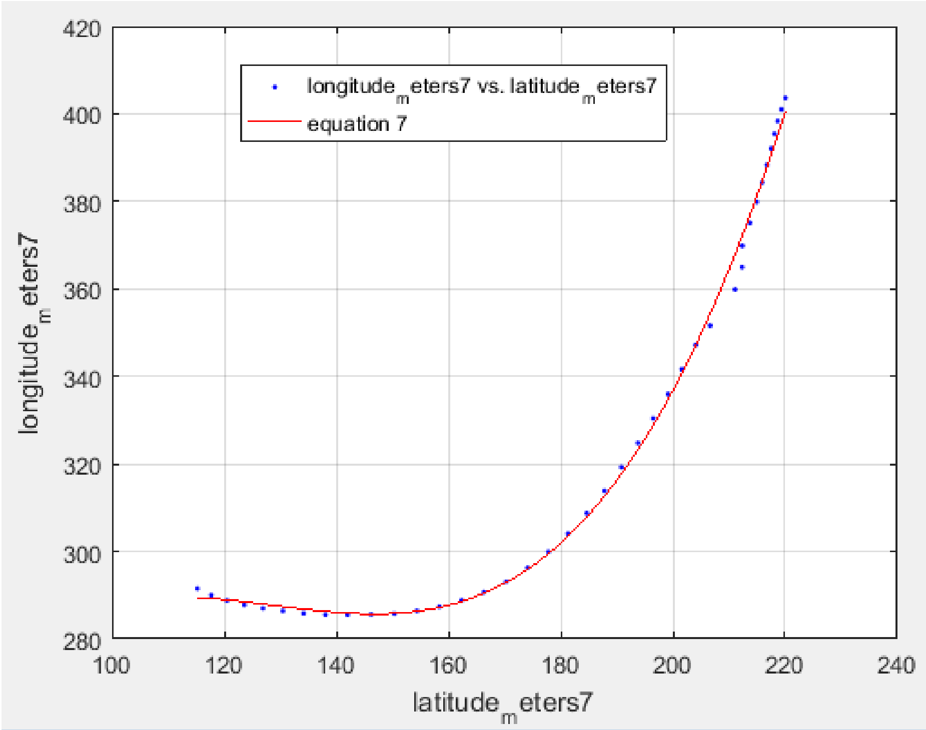


Ploy 3 type

When X is from 115meters to 220meters, y_7 causes 2.3211 meters error compared to the GPS track (the blue data points is GPS data and the red line is polynomial model).

$$\{115 \leq x \leq 220\}$$

$$y_7 = 0.0001675\,x^3 - 0.06467x^2 + 8.158\,x \ - 48.18$$

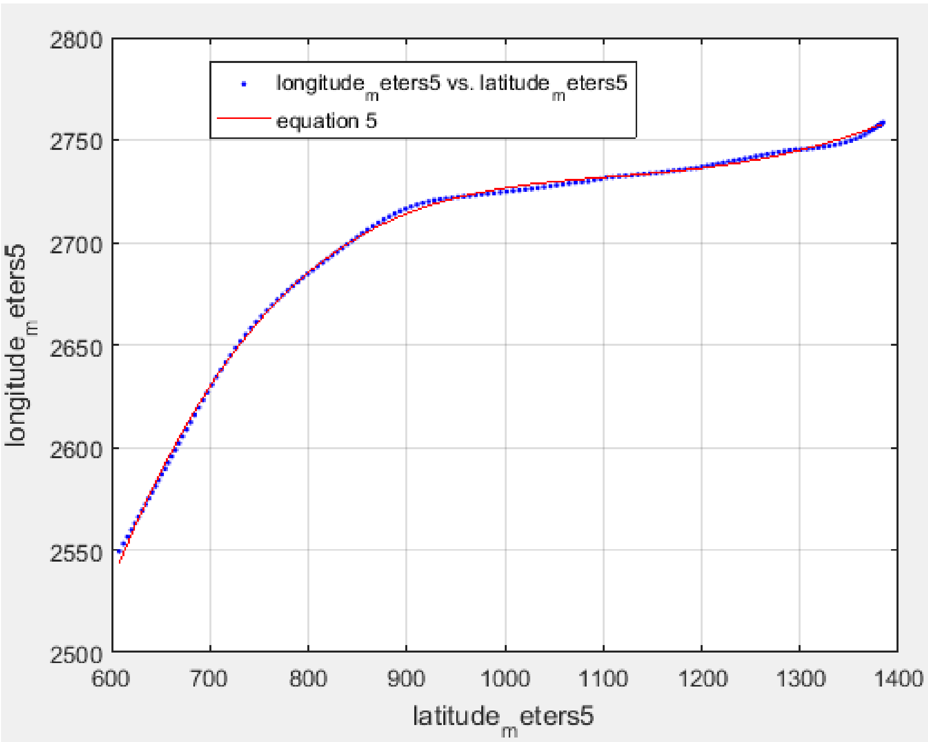


Ploy 4 type

When X is from 601 meters to 1384.5 meters, y_5 causes 1.174 meters error compared to the GPS track (the blue data points is GPS data and the red line is polynomial model).

$$\{601 \leq x \leq 1384.5\}$$

$$y_5 = -7.674 * 10^{-10}\, x^4 + 4.356 * 10^{-6} x^3 - 0.008831 x^2 + 7.741 x + 228.3$$

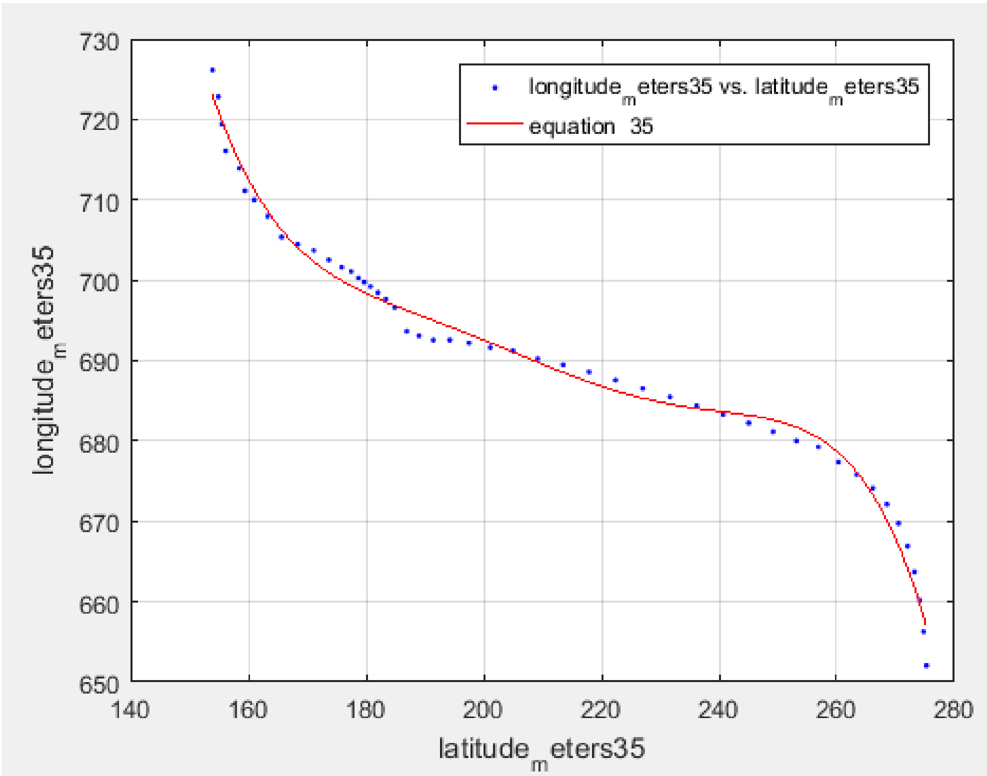


Ploy 5 type

When X is from 154meters to 275meters, y_{35} causes 1.7732 meters error compared to the GPS track (the blue data points is GPS data and the red line is polynomial model).

$$\{154 \le x \le 275\}$$

$$y_{35} = -4.636 * 10^{-8}x^5 + 4.917 * 10^{-5}x^4 - 0.02076x^3 + 4.364x^2 - 456.9x + 1.979 * 10^4$$

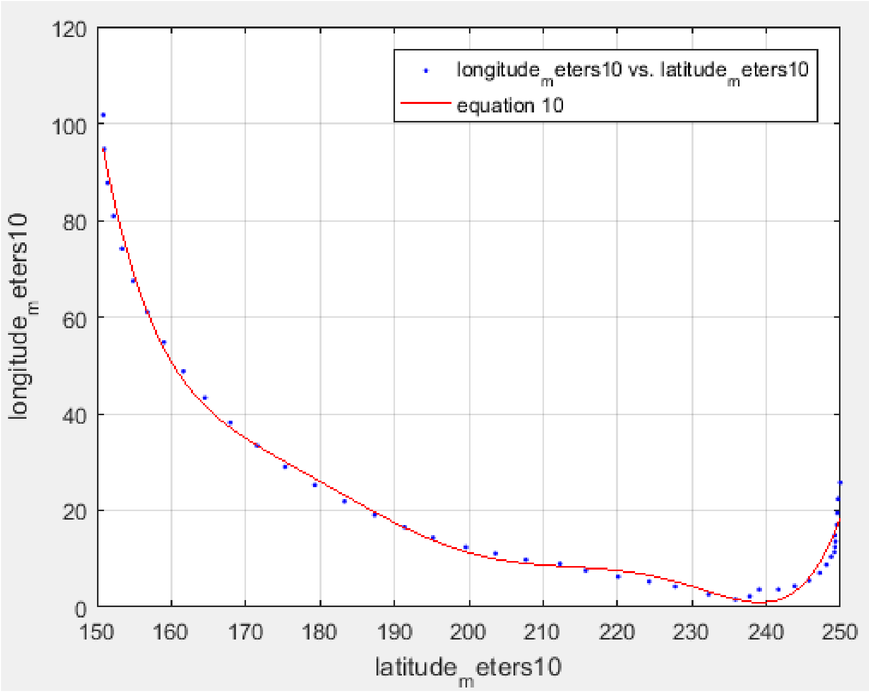


Ploy 6 type

When X is from 250meters to 150meters, y_{10} causes 2.6781 meters error compared to the GPS track (the blue data points is GPS data and the red line is polynomial model).

$$\{250 \ge x \ge 150\}$$

$$y_{10} = 7.392 * 10^{-9}x^6 - 8.918 * 10^{-6}x^5 + 0.004464\, x^4 - 1.187x^3 + 176.8\, x^2 - 1.399 * 10^4\, x + 4.596 * 10^5$$

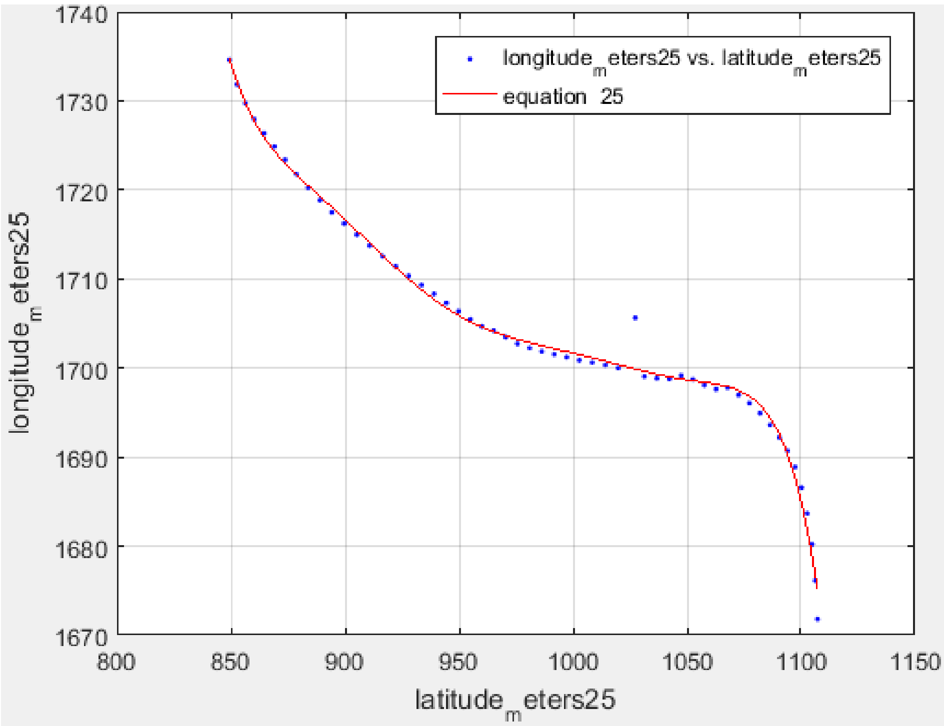


Ploy 7 type

When X is from 849.5meters to 1107meters, y_{25} causes 1.17 meters error compared to the GPS track (the blue data points is GPS data and the red line is polynomial model).

$$\{849.5 \leq x \leq 1107\}$$

$$y_{25} = -1.108 * 10^{-13}x^7 + 7.545 * 10^{-10}x^6 - 2.2 * 10^{-6}x^5 + 0.003558\, x^4 - 3.45x^3 + 2005\, x^2 - 6.464 * 10^5 x + 8.923 * 10^7$$



After gaining all 36 polynomial model performance information, I can evaluate the whole mathematical model accuracy. Because each polynomial function lead to a corresponding error, we could add all their errors together and divide by the number of polynomial models. As a result, the average error of 36 polynomial equations will be gotten to represent the whole mathematical model precision. The under code can calculate the average error of 36 polynomial models.

```
sum = 0;
for n=1:36
    sum = sum +gof(n).rmse;
end
average = sum/36;
disp('the average error is:');
disp(average);
```

```
the average error is:
    2.7498
```

To sum up, the whole mathematical model causes 2.7498 meters error compared with GPS trajectory. In addition, the below table will show the errors of other 27 polynomial models.

Polynomial equation	Errors (meters)	Type (degree)
y_1	1.0715	Ploy 4
y_2	0.2199	Ploy 5
y_3	7.1212	Ploy 3
y_4	1.0723	Ploy 3
y_6	2.3038	Ploy 5
y_8	0.7743	Ploy 4
y_9	6.5048	Ploy 4
y_{10}	2.6781	Ploy 6
y_{11}	1.3015	Ploy 3
y_{12}	1.8584	Ploy 3
y_{13}	1.6479	Ploy 1
y_{14}	0.8612	Ploy 2
y_{15}	4.4600	Ploy 1
y_{16}	2.2338	Ploy 7
y_{17}	4.1894	Ploy 2
y_{18}	8.7176	Ploy 5
y_{19}	4.1216	Ploy 1
y_{20}	2.1878	Ploy 5
y_{22}	2.9812	Ploy 4
y_{23}	1.8030	Ploy 5
y_{27}	2.3747	Ploy 5
y_{28}	1.1834	Ploy 3
y_{29}	0.5093	Ploy 5
y_{30}	0.1331	Ploy 4

y_{31}	7.9660	Ploy 2
y_{33}	0.5102	Ploy 3
y_{34}	1.7948	Ploy 2
y_{36}	0.9938	Ploy 4

According to above errors table, the errors that is higher than average error would classify to low-quality mathematical models, which contain $y_9, y_{15}, y_{17}, y_{18}, y_{19}, y_{22}, y_{31}$. The other polynomial models are shown in appendix.

Discussion

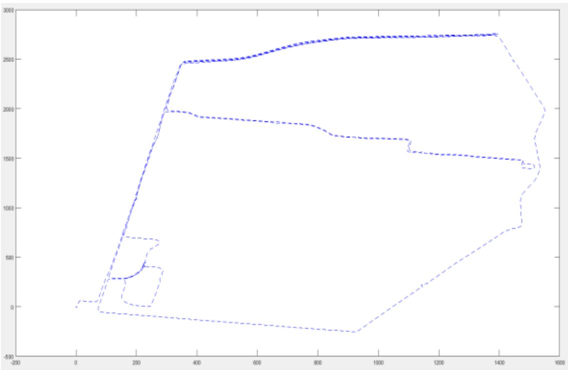
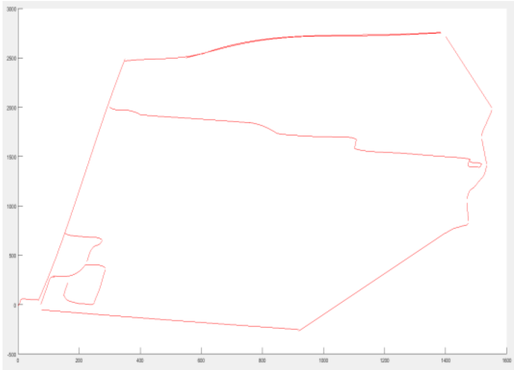
In the literature review and outcome part, we have already given the detail theory about how to use least-squares method to acquire the suitable polynomial curve fitting results. Then, we employ the MATLAB to successfully construct 36 polynomial equations. The whole mathematical model only generate 2.7498 meters error compared to the real GPS tracks. Based on this mathematical model, the least-squares method actually work pretty well. However, there are still some limitations in least-squares approximating. As it attempt to minimize the squares of error, the squares operation could amplify the actual errors if there are too many outliers in dataset. Therefore, the least-squares method require the approximating data have good quality. In this project, the GPS data record the real log vehicle driving path which is smooth, so there are not too many outliers in the experimental dataset. If the data quality is not really good, minimizing the average error (mention in literature review) may give us a good curve fitting result. The average error method is shown below:

$$\boldsymbol{E_{Average}(f(x)) = \frac{1}{n}\sum_{k=1}^{n}|f(x_k) - y_k|}$$

Furthermore, when do I should use the Maximum Error? The advantage of the Maximum Error approach is to have high computational efficiency, because it only need to minimize the maximum error. However, if there is any outliers, the approximating result could become terrible. Hence, we would better exploit Maximum Error method when the dataset have really high quality that do not have any outliers.

$$\boldsymbol{E_\infty (f(x)) = max\,|f(x_k) - y_k| \quad (1 < k < n)}$$

During the project, we utilize two ways to display the target road trajectory. The first approach is directly to employ the GPS tracks (the blue route in left hand figure). The second method is to use the polynomial functions (the red route in right hand figure).

Both techniques have ability to show the target road orbit, so what are the benefits to exploit the polynomial functions to describe the target road. Compared with GPS tracks, the polynomial models can save a great deal of computer memory space to store the target road trajectory. Via building the mathematical model, the computer can only store hundreds of polynomial parameters to depict the target road, including the polynomial coefficient $a_n, a_{n-1} \ldots a_1, a_0$ and the definition domain X of every equations. Nevertheless, if the GPS data are utilized to show the trajectory, we have to apply a huge amount of data points (latitude and longitude). In this project, both latitude and longitude matrix have 5056 double value, so we require to save the trajectory by storing 10112 double value. Through comparing hundreds of parameters and thousands of GPS data points, it is obvious that using polynomial functions to record the trajectory would save lots of memory space. Another advantage is that we can obtain any position in the target orbit by given any X value in definition domain. Conversely, if the given X value does not exist in GPS data point, the position can never be acquired. In the end of outcome, there is a table to show the errors of every polynomial models. To analyse how degree influence the accuracy, I determine to calculate the average error of each polynomial type. The result is exhibited in below table.

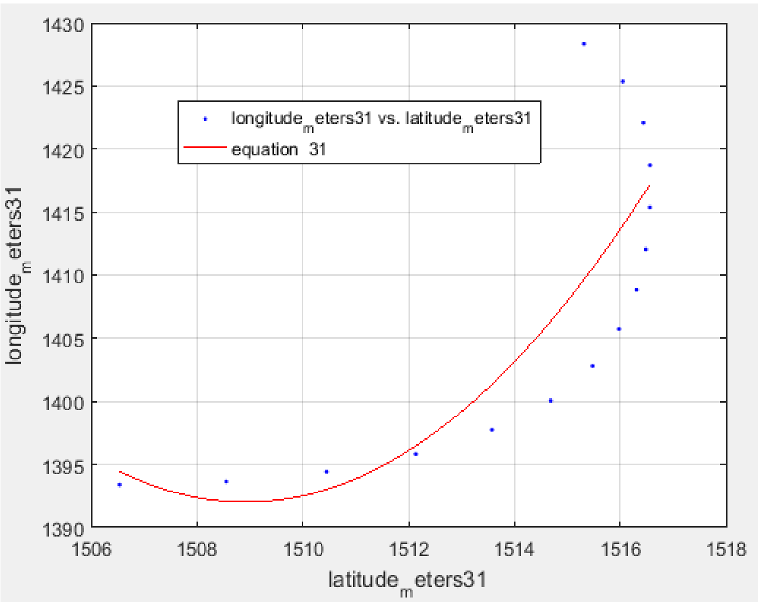
Polynomial type	Average error for each type
Poly1 $(y_{13}, y_{15}, y_{19}, y_{21}, y_{26})$	5.1582 meters
Poly2 $(y_{14}, y_{17}, y_{24}, y_{31}, y_{32}, y_{34})$	2.6656 meters
Poly3 $(y_3, y_4, y_7, y_{11}, y_{12}, y_{28}, y_{30}, y_{33})$	1.9376 meters
Poly4 $(y_1, y_5, y_8, y_9, y_{22})$	2.5151 meters

Poly5 $(y_2, y_6, y_{18}, y_{20}, y_{23}, y_{27,} y_{29}, y_{35}, y_{36})$	2.5612 meters
Poly6 (y_{10})	2.6781 meters
Poly7 (y_{16}, y_{25})	1.7019 meters

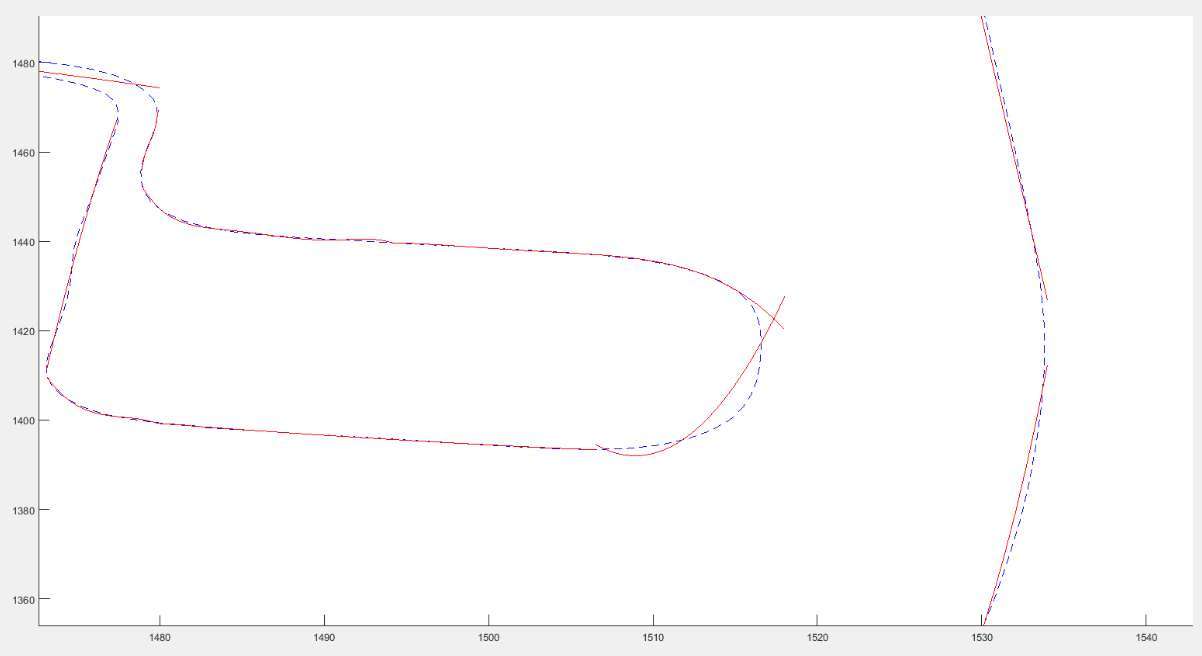
Based on the above table, the linear models (ploy1) have the biggest average errors, and the ploy7 type models have the smallest average meters. Even the errors of each type fluctuate a little when the polynomial degree increase from 1 to 7, we observe the accuracy is upward trend. Especially, when the polynomial degree raise from 1 to 3, the upward trend is significant. Nonetheless, after the degree increase to 4, the error do not reduce too much. In summarise, increasing the degree of polynomial could help this model to perform better.

According to the performance of each polynomial model, we offer a low-cost way to build mathematical model of road orbit. By using this method, many small organizations or individuals can map the road trajectory. Especially, the polynomial function is really expert in portraying the low-medium radian curve, which is suitable to describe the highway route. Furthermore, polynomial models are beneficial to be calculated and understand by computer, so it can assist the Autonomous vehicle or navigation system to recognize the road trajectory

Although the manipulation of polynomial degree can provide many flexible curve shape, there are still some limitations in polynomial curve fitting. The first drawback is that polynomial is not good at approximating a big radian curve. The under graph demonstrate an example. $\{1518 \geq x \geq 1506.5\}$ $y_{31} = 0.4313x^2 - 1302x + 9.835 * 10^5$



When x is from 1518 meters to 1506.6 meters, y_{31} causes 7.9660 meters error. No matter how the polynomial degree change, y_{31} error cannot be minimize more. The second problem is concatenation errors, which is shown in under figure (the blue line is GPS tracks and the red line is fitting result).



In the right hand of above picture, there are two red line which represent different polynomial equations. The two polynomials fit well to the GPS tracks (blue), but there is an empty space in the middle of two equations. These two polynomials do not connect tightly, which generate broken part. Moreover, when one polynomial model concatenate with another polynomial, it may lead to noisy cross point (show in central position of above picture). In addition, as the whole methodology is based on the GPS data, the data quality strictly depend on GPS receiver. If there are some accidental and systematic error in GPS receiver or satellites, the wrong mathematical model may be constructed because of poor quality data.

In the future work, since the polynomial cannot approximate big radian curve well, we need to employ other mathematical model to do the curve fitting, such as circumference function. Meanwhile, the road should be divide into proper segments, which try to minimize concatenation errors. Be doing above improvement, a more precision could be established. This time we build a 2D mathematical model for the target road via longitude and latitude GPS data. Next, we will attempt to construct 3D road surface model by add the altitude parameters. The 3D mathematical model will supply more detail road information like gradient attribute.

Conclusion

During this research, 36 polynomial functions are successfully constructed to portray the trajectory of the target road by curve fitting algorithm. In this article, three methods to get best fitting are introduced, including maximum error, average error and least-squares. For this project, the least-squares method is chosen to minimize the errors between GPS data points and mathematical model. In research question 1, the process of least-squares curve fitting is proved step by step. The major approach is to minimize the value of $\sum_{k=1}^{n} | \, a_n {x_k}^n + a_{n-1} \, {x_k}^{n-1} + \cdots + a_2 {x_k}^2 + a_1 x_k + a_0 - y_k|^2$. To minimize errors, a series of differential functions are established to make $| \, a_n {x_k}^n + a_{n-1} \, {x_k}^{n-1} + \cdots + a_2 {x_k}^2 + a_1 x_k + a_0 - y_k|^2$ become 0, which are $\frac{dE}{da_n} = 0 \; \ldots \frac{dE}{da_0} = 0$.

After solve the function, an N dimension matrix equations is obtained. The matrix is shown in below:

$$\begin{pmatrix} \sum_{k=1}^{n} {x_k}^{n+1} & \cdots & \sum_{k=1}^{n} x_k \\ \vdots & \vdots & \vdots \\ \sum_{k=1}^{n} {x_k}^{n} & \cdots & \sum_{k=1}^{n} a_0 \end{pmatrix} \begin{pmatrix} a_n \\ \vdots \\ a_0 \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^{n} y_k x_k \\ \vdots \\ \sum_{k=1}^{n} y_k \end{pmatrix}$$

In research question 2, the target road is separated to 36 segments based on the inflection points. The zoom out method is used to find these inflection points. Then, by exploiting the curve fitting tool in MATLAB, the polynomial degree is manipulated for established 36 polynomial models of corresponding segments. According to the error of each model, the best case leads to 0.0977meters error and the worst case causes 10.3106 meters error. Next, through calculating each error of all polynomial equations, the average error of whole mathematical model is 2.7498 meters. In 36 polynomial equation, there are sever types of polynomial (from poly1 to poly7). In discussion part, three significant benefits for polynomial curve fitting are summarised. Firstly, the polynomial model only utilize several parameters to store the orbit information of target road. Compared with GPS tracks, it can save lots of computer memory. The second advantage is that polynomials could have flexible curve shapes by manipulating its degree. The final point is that positions are able to gain for any given x in definition domain. Furthermore, based on the average errors of each polynomial type, there is a trend which is properly increasing polynomial degree may reduce its error. Through constructing the road mathematical model, it can assist the self-driving and navigation system to recognize the trajectory of target road. Moreover, the method which use the GPS data to establish polynomial model can help the small organization or individual to daw a map. However, there are still three limitations for polynomial curve fitting. The fist drawback is polynomial cannot represent a big radian curve well. The concatenation errors between different polynomial is the second problem. Final risk is the data quality is totally depend on the GPS receiver. A poor data probably generate a low quality mathematic model. In the future work, I will add circular model to depict a big radian curve. Also, based on the 2D mathematical model, we want to build 3D math model to record the road geometrical information.

Reflection

I do the best activity is to learn MATLAB programming language by myself. I search the MATLAB online course in YOUTUBE. With one week watching this course, I understand the syntax of MATLAB. Then I use the curve fitting tool in MATLAB to build my first polynomial model. The insufficient part is about literature review. As an only focus on studying MATLAB programming language, when I start to write report I feel really difficult for how to write literature review.

There are two big challenges during the project. The first problems is at the beginning of the project. When I first meet my academic supervisor, I cannot understand what I should do in project. I have no concept for curve fitting algorithm, so it is difficult to understand what teacher talk about. Since I do not know how to begin the project, I feel lots of pressure about it. Time goes by, after my supervisor explain me several times, I figure out the project is to build mathematical model of target road. The second challenge is how to plan your project. Since the project do not have any assignment after week 4, it make me easily forget to work on project. In early of the semester, when supervisor ask us for meeting, I am hurry to learn and do the work in project. To solve this problem, I utilize the middle break to concentrate on project for one week. And after middle break, I select two days every week for doing project.

My project is to build mathematical model for target road via polynomial curve fitting algorithms. Since the whole road cannot be describe by a single function, I need to divide the target road to different segments. Then build model for these segments and combine them together. The most difficult is how to divide the target road to different part. Through working for a period, I decide to separate them based on inflection points.

The programming skills require in the further development. Nowadays, computers expand to all kinds of areas. No matter what you do, you cannot leave computer. Thereby, the programming skills can help me deal different kinds of problems.

In the whole project, the most import thing is we should study by myself. The academic supervisor only give you a framework and basic theory about project. I need to do learn the detail knowledge in google and YouTube. The most import thing in the project is to study how to use MATLAB. When I begin to learn MATLAB, I found there are many similar concept in C#. Therefore, I just use one week to master the basic knowledge of MATLAB. To sum up, the ability for quick learning is the most critical thing I learned.

Because I do not catch the free time, it make me become very busy at the last week. Almost all subjective is due in last week, so it give me a lot of pressure. This event teach me that I should separate a big work to many small assignment and do it in daily life. Do not accumulate all the assignments in last week. This project type is a research project. It show me how to do a research. Not only the coding is important, bust also read and understand related literature review is critical.

This project I learn least-squares algorithms. The least-squares method is widely employed in different curve fitting algorithms and regression analysis. I can use it in data science area. Also, MATLAB is widely used in engineer and mathematical area. I can do some engineer analyse in the future. Also, the project teach me the research skills. It can help me to do research if I want to study for a doctorate.

Reference

Anderson, J. M. (2014). *Autonomous vehicle technology : a guide for policymakers*. Santa Monica, CA: Rand Corporation.

Chatterjee, S. (Singer-songwriter). (2011). Residuals. On.

Chen, D., & Chen, L. (2013). Practical Constraint K-Segment Principal Curve Algorithms for Generating Railway GPS Digital Map. *Mathematical Problems in Engineering, 2013*. doi: 10.1155/2013/258694

Di Mascio, P., Di Vito, M., Loprencipe, G., & Ragnoli, A. (2012). Procedure to Determine the Geometry of Road Alignment Using GPS Data. *Procedia - Social and Behavioral Sciences, 53*, 1202-1215. doi: 10.1016/j.sbspro.2012.09.969

Gilat, A. (2016). *MATLAB* (6th ed. ed.). New York: Wiley.

Hassan, S., Farhan, M., Mangayil, R., Huttunen, H., & Aho, T. (2013). Bioprocess data mining using regularized regression and random forests. *BMC Systems Biology, 7*(Suppl 1). doi: 10.1186/1752-0509-7-S1-S5

Ji, C., Yang, X., & Wang, W. (Singer-songwriters). (2015). A Novel Method for Image Recognition Based on Polynomial Curve Fitting. On: IEEE.

Jiménez, F., Aparicio, F., & Estrada, G. (2009). Measurement uncertainty determination and curve-fitting algorithms for development of accurate digital maps for advanced driver assistance systems. *Transportation Research Part C, 17*(3), 225-239. doi: 10.1016/j.trc.2008.10.004

Khartov, A. A., & Zani, M. (2018). Asymptotic analysis of average case approximation complexity of additive random fields. *Journal of Complexity*. doi: 10.1016/j.jco.2018.04.001

Markushevich, A. I. (2014). *Entire Functions*. Saint Louis: Elsevier Science.

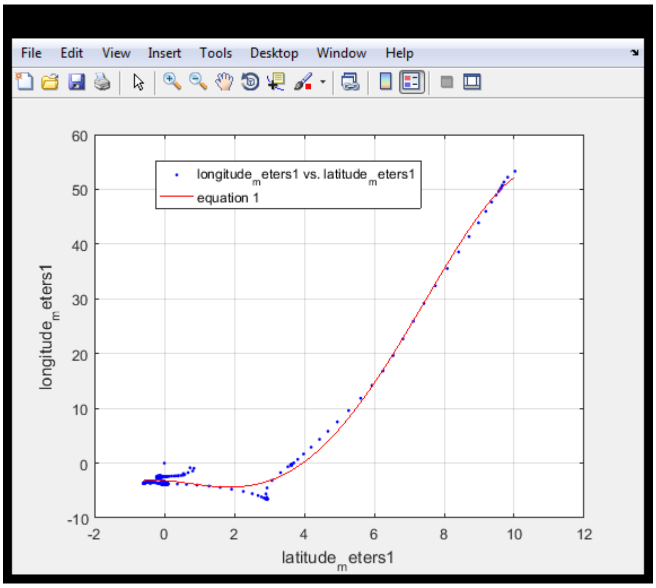
Peng, L., Sun, Y., Meng, Z., Wang, Y., & Xu, Y. (2013). A new method for determining the characteristics of solar cells. *Journal of Power Sources, 227*, 131-136. doi: 10.1016/j.jpowsour.2012.07.061

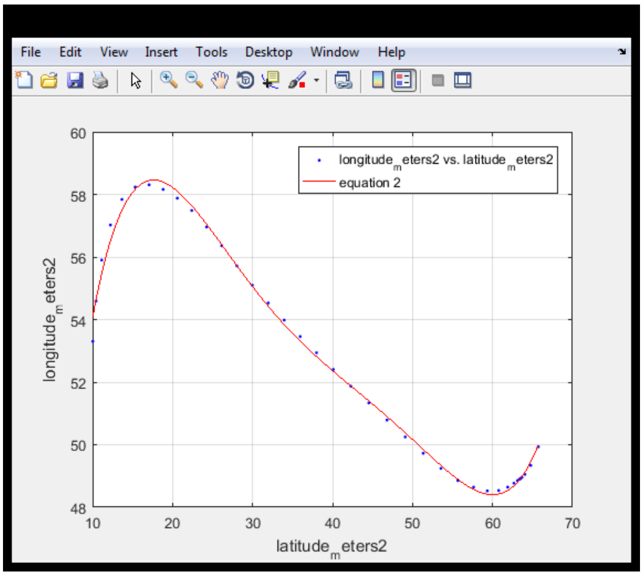
polynomial. On. (2018).

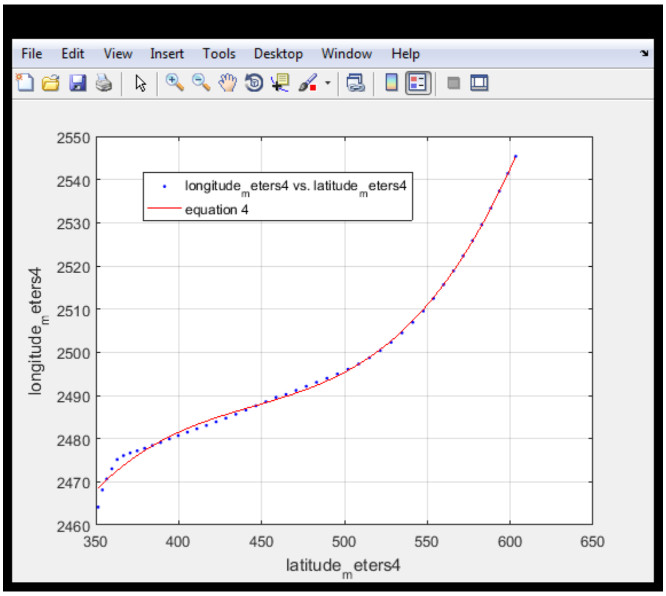
Xu, W., Chen, W., & Liang, Y. (2018). Feasibility study on the least square method for fitting non-Gaussian noise data. *Physica A: Statistical Mechanics and its Applications, 492*, 1917-1930. doi: 10.1016/j.physa.2017.11.108

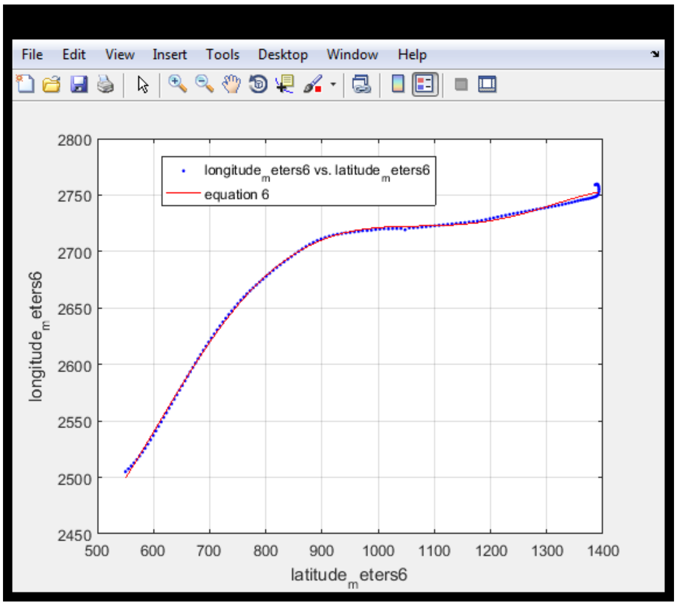
Yang, Y., & Gordon, S. P. (2014). Interpolation and Polynomial Curve Fitting. *Mathematics Teacher, 108*(2), 132-141. doi: 10.5951/mathteacher.108.2.0132

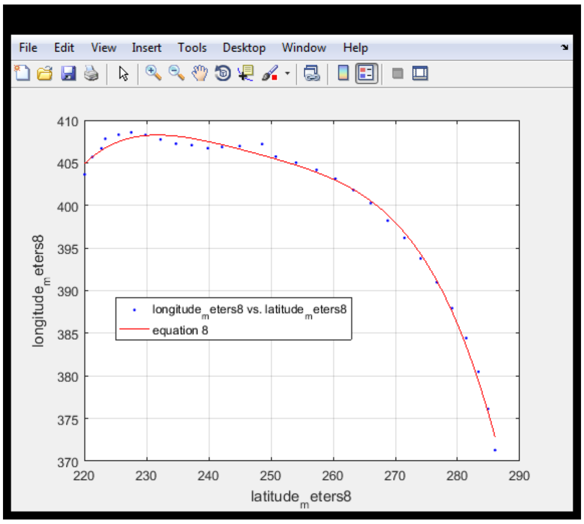
Appendix

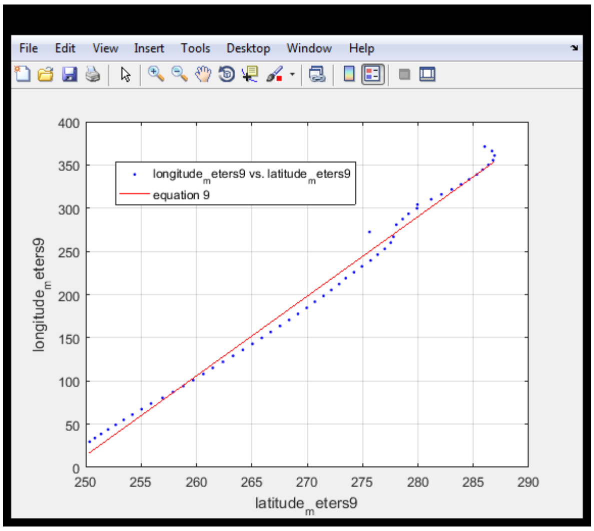


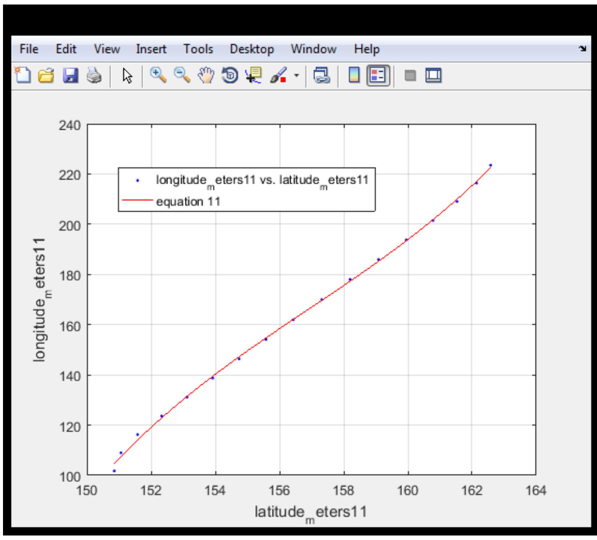


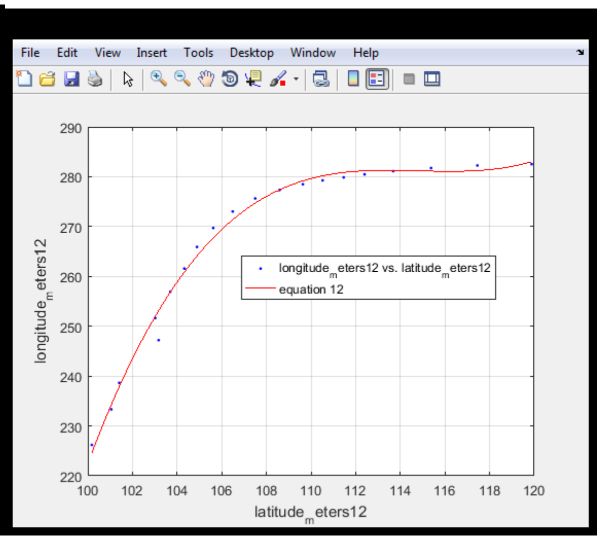


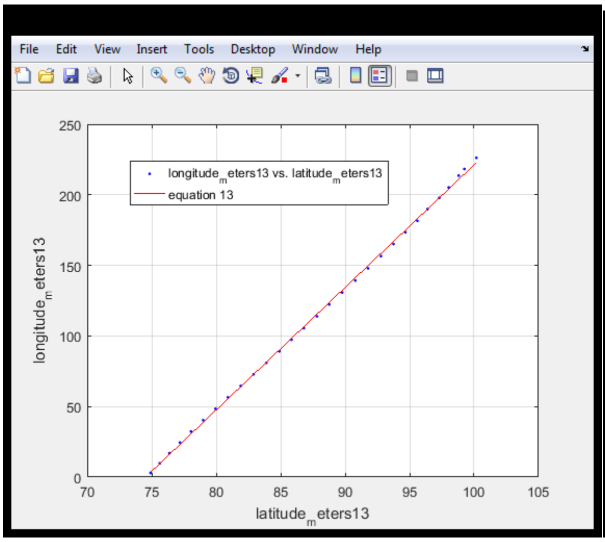


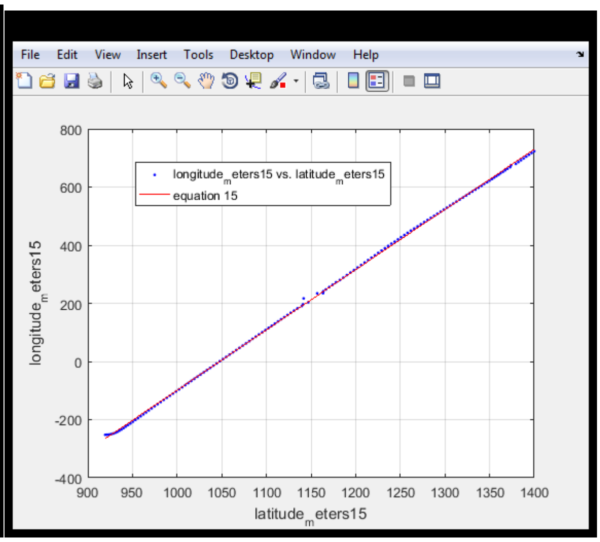

File Edit View Insert Tools Desktop Window Help
longitude_meters8 vs. latitude_meters8
equation 8
longitude_meters8
latitude_meters8
410
405
400
395
390
385
380
375
370
220
230
240
250
260
270
280
290

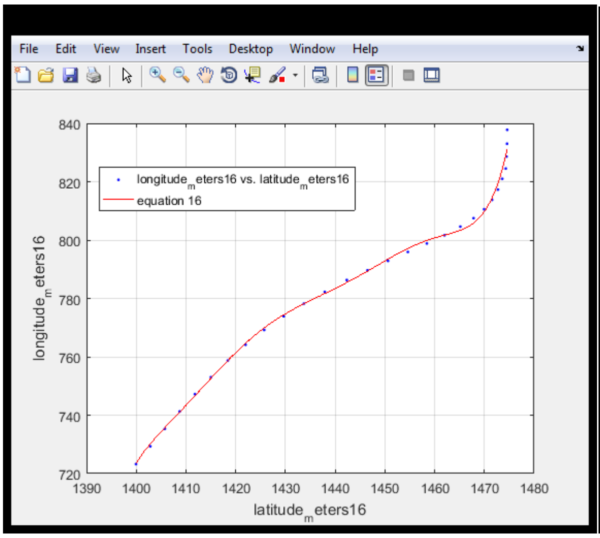

File Edit View Insert Tools Desktop Window Help
longitude_meters9 vs. latitude_meters9
equation 9
longitude_meters9
latitude_meters9
400
350
300
250
200
150
100
50
0
250
255
260
265
270
275
280
285
290

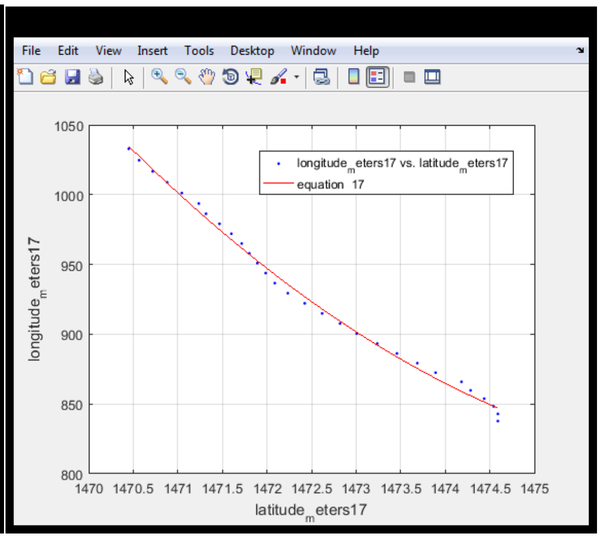

File Edit View Insert Tools Desktop Window Help
longitude_meters11 vs. latitude_meters11
equation 11
longitude_meters11
latitude_meters11
240
220
200
180
160
140
120
100
150
152
154
156
158
160
162
164

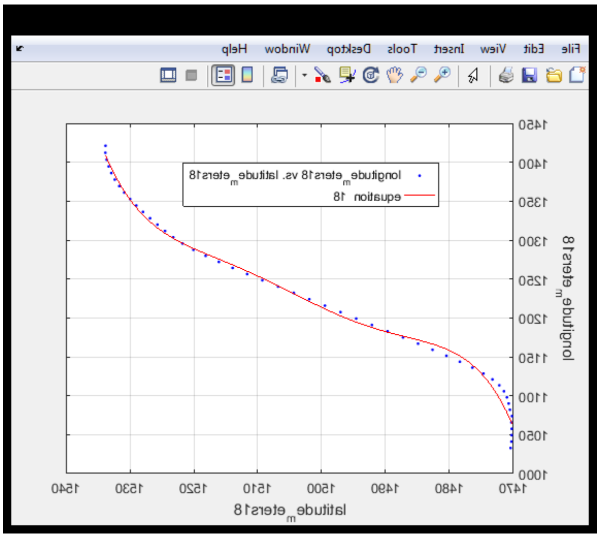

File Edit View Insert Tools Desktop Window Help
longitude_meters12 vs. latitude_meters12
equation 12
longitude_meters12
latitude_meters12
290
280
270
260
250
240
230
220
100
102
104
106
108
110
112
114
116
118
120

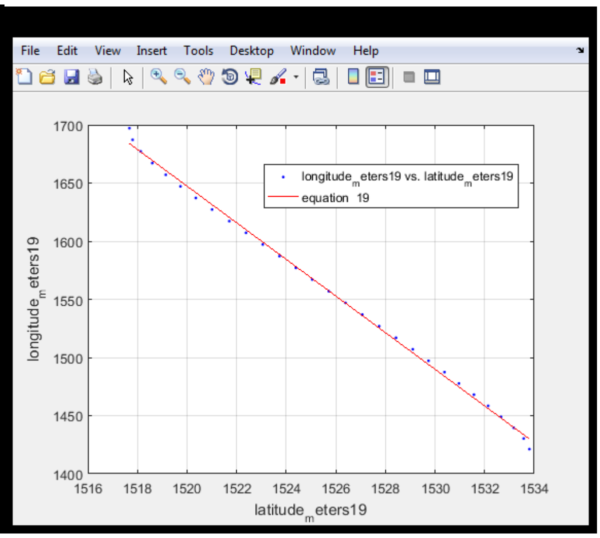

File Edit View Insert Tools Desktop Window Help
longitude_meters13 vs. latitude_meters13
equation 13
longitude_meters13
latitude_meters13
250
200
150
100
50
0
70
75
80
85
90
95
100
105

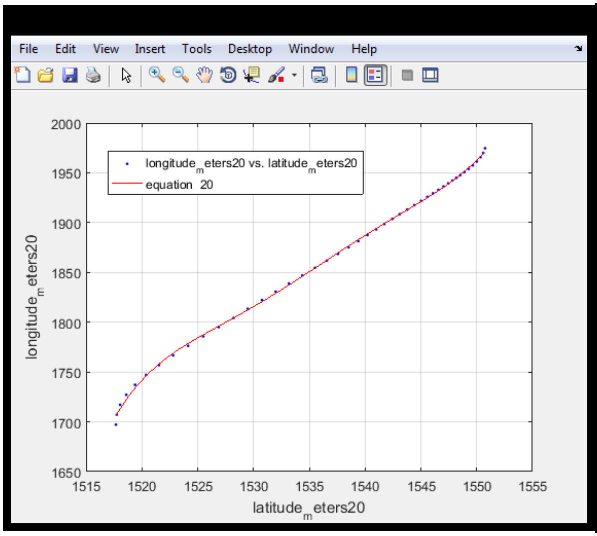

File Edit View Insert Tools Desktop Window Help
longitude_meters15 vs. latitude_meters15
equation 15
longitude_meters15
latitude_meters15
800
600
400
200
0
-200
-400
900
950
1000
1050
1100
1150
1200
1250
1300
1350
1400

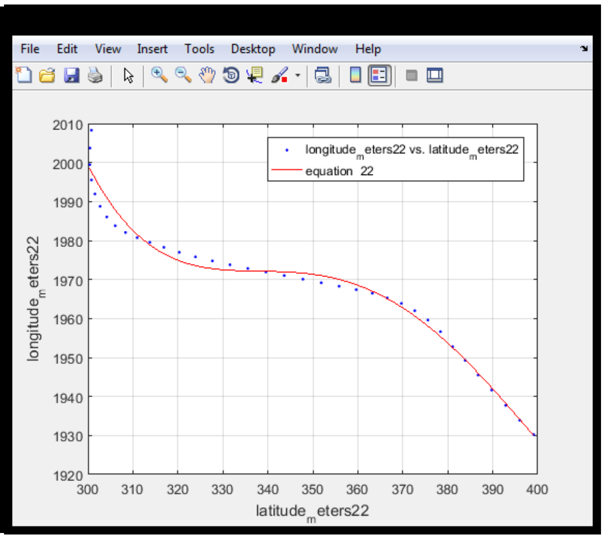

File Edit View Insert Tools Desktop Window Help
longitude_m eters16 vs. latitude_m eters16
equation 16
longitude_m eters16
latitude_m eters16

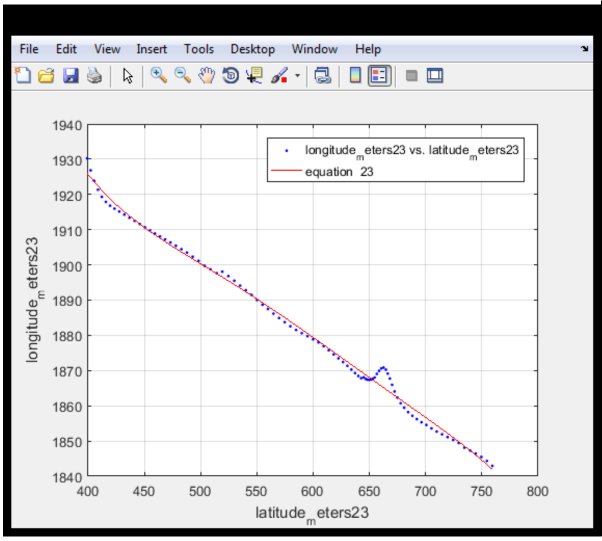

File Edit View Insert Tools Desktop Window Help
longitude_m eters17 vs. latitude_m eters17
equation 17
longitude_m eters17
latitude_m eters17

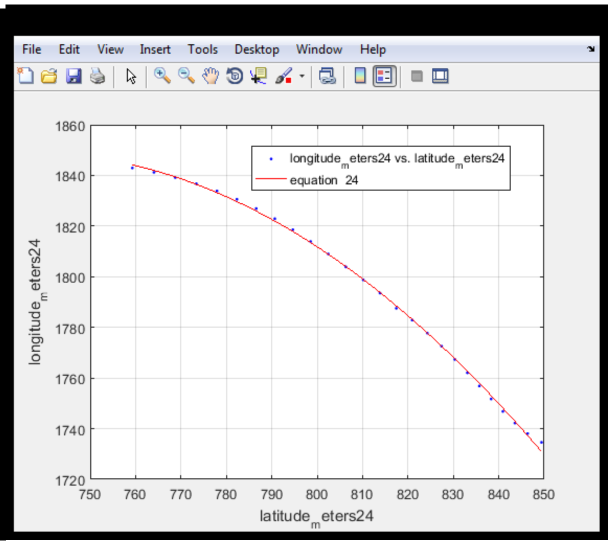

File Edit View Insert Tools Desktop Window Help
longitude_m eters18 vs. latitude_m eters18
equation 18
longitude_m eters18
latitude_m eters18

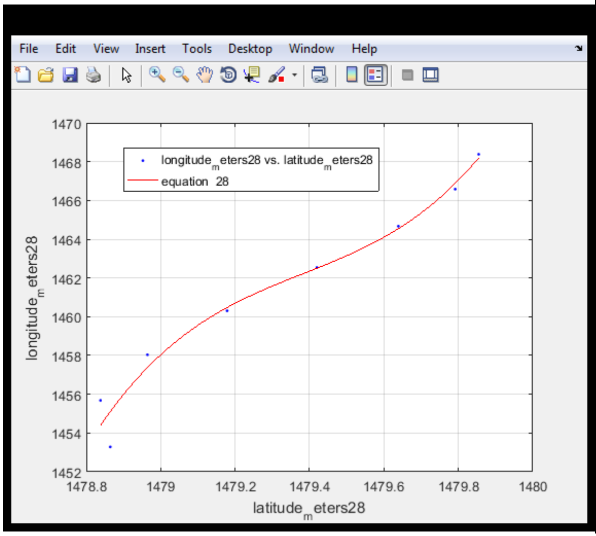

File Edit View Insert Tools Desktop Window Help
longitude_m eters19 vs. latitude_m eters19
equation 19
longitude_m eters19
latitude_m eters19

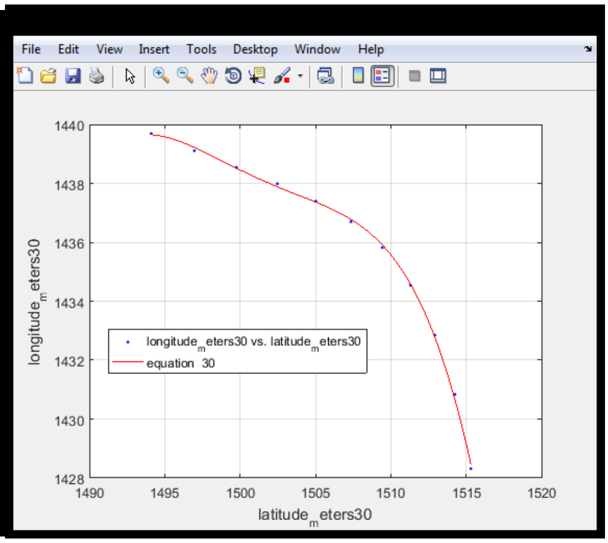

File Edit View Insert Tools Desktop Window Help
longitude_m eters20 vs. latitude_m eters20
equation 20
longitude_m eters20
latitude_m eters20

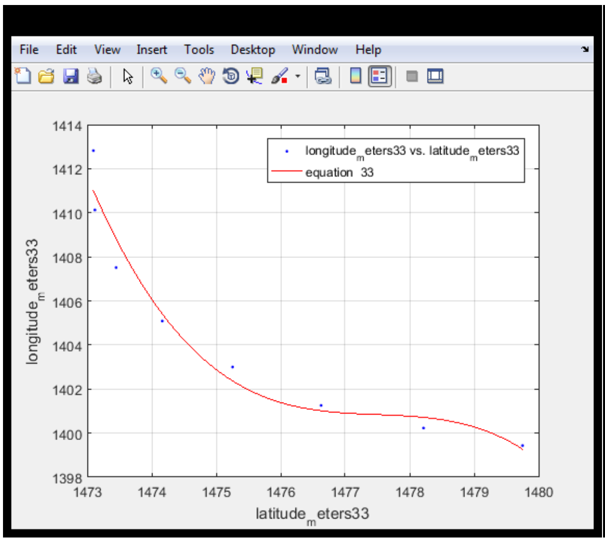

File Edit View Insert Tools Desktop Window Help
longitude_m eters22 vs. latitude_m eters22
equation 22
longitude_m eters22
latitude_m eters22

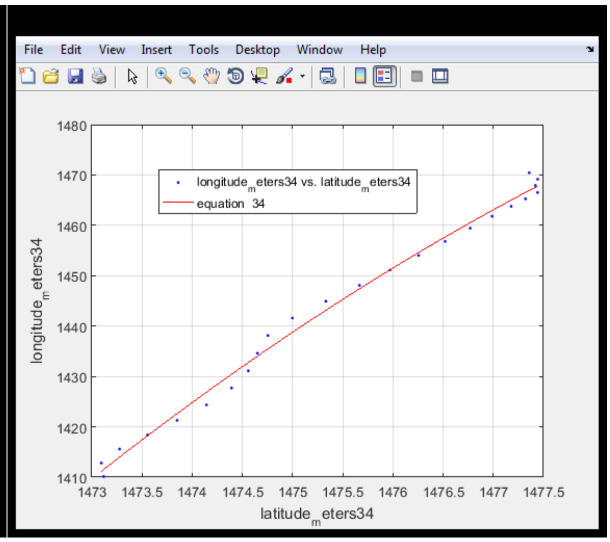

longitude_meters23 vs. latitude_meters23
equation 23
longitude_meters23
latitude_meters23

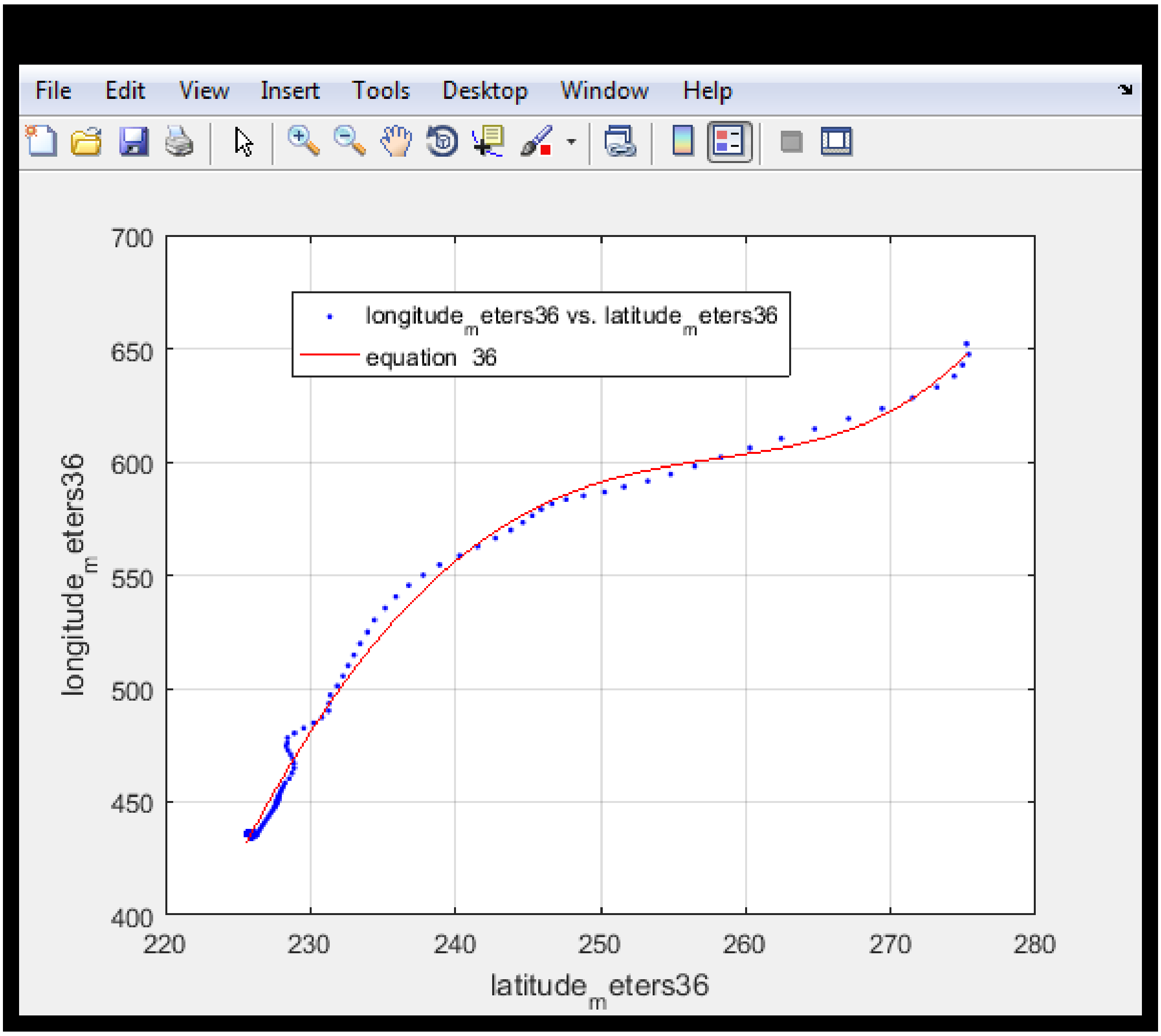

longitude_meters24 vs. latitude_meters24
equation 24
longitude_meters24
latitude_meters24


longitude_meters28 vs. latitude_meters28
equation 28
longitude_meters28
latitude_meters28


longitude_meters30 vs. latitude_meters30
equation 30
longitude_meters30
latitude_meters30


longitude_meters33 vs. latitude_meters33
equation 33
longitude_meters33
latitude_meters33


longitude_meters34 vs. latitude_meters34
equation 34
longitude_meters34
latitude_meters34


File Edit View Insert Tools Desktop Window Help
700
650
600
550
500
450
400
220
230
240
250
260
270
280
longitude_meters36 vs. latitude_meters36
equation 36
longitude_meters36
latitude_meters36