UNIVERSITY OF LIVERPOOL
DEPARTMENT OF ELECTRICAL ENGINEERING & ELECTRONICS

Final Report for Project

Aircraft Tracking Algorithm using Automatic Dependent Surveillance Broadcast (ADS-B) data

Author: Dijun Liao (201376109)

Project Supervisor: Dr Angel Garcia-Fernandez

Project Assessor: Dr. Valerio Selis

Declaration of academic integrity

The standard University of Liverpool statement of academic integrity [6] should go here as follows:

I confirm that I have read and understood the University's Academic Integrity Policy.

I confirm that I have acted honestly, ethically and professionally in conduct leading to assessment for the programme of study.

I confirm that I have not copied material from another source nor committed plagiarism nor fabricated, falsified or embellished data when completing the attached piece of work. I confirm that I have not copied material from another source, nor colluded with any other student in the preparation and production of this work.

0 Abstract

The ADS-B is a new kinds of aircrafts surveillance that based on the satellite aircraft navigation system and ground station sensors and radars. These data are totally open sources and broadcasted so that everyone can use it to find the location of an aircraft and tracking it. In this report, it will introduce a new kind of aircraft tracking that based on Time Difference of Arriving (TDoA), k-Nearest Neighbour(k-NN), Kalman Filter (KF), Extended Kalman Filter (EKF) and RTS Smoother. This method consisted by three parts, the first is to use the TDoA and k-NN method to estimate the initial position. Then KF will use the measurement GPS data to estimate the position of the whole route; meanwhile, the EKF will use RSSI data to estimate the position in another way. And finally, the RTS smoother will use to soptimize the estimation of KF and EKF. By using this method, it can reduce the time delay error of estimate the position by using time and since the KF, EKF and RTS will included some noise and velocity of an aircraft, so it can estimate and predict the position of an aircraft included the process and measurement data. Though it can reduce the data, it still can improvement like the coordinate can use WGS 84 and use neural network to estimation parameters in KF, EKF and RTS. By developed this method, it can let the air traffic control faster and efference. Furthermore, this method is the fundamental of aircraft auto-pilot.

Due to the limitations of computer and database, the TDoA and k-NN joint location method will just have theoretical analysis and modelling whilst the KF and EKF has been constructed and tested on MATLAB.

Contents

1 Project Specification and Gantt Chart

1.1 Project Specification

As the number of commercial aircraft has increasing, the requirement to construct a new aircraft surveillance has been increased too. So, ADS-B, as a next-generation method to tracking aircraft has been included into air traffic method [1]. The most important things for an ADS-B system is to implement a tracking algorithm. So, this report will talk about an aircraft tracking method that based on Kalman Filter (KF) and GPS data. Extend Kalman Filter (EKF) and Received Signal Strength Indication (RSSI) data, Time Difference of Arriving (TDoA) and k-Nearest Neighbours (k-NN).

For further specification of this project, it has been shown on the appendix (appendix 1 is the copy of the original specification report, appendix 2 is the revision of the specification and Gantt Chart).

1.2 Gantt Chart

The following is the final version of Gantt Chart of this project (for original one and full revision, it has been shown in the appendix 1 and 2):

Figure 1.1 Final Gantt Chart of this project

2 Introduction

2.1 Automatic Depended Surveillance Broadcast

Automatic Depended Surveillance Broadcast (ADS-B) is a new surveillance technology that based on the aircraft navigation system and broadcast its position, velocity and other essential information automatically [2] [1]. Since it is a communal and automatic surveillance broadcast network, so it didn't extra input from external or pilots. Besides, any other aircraft or ground station can receive that without sending an inquiry signal first [1]. That means the aircraft can do some avoid action and the air traffic controller can deployment these aircraft more effectively.

Though ADS-B System is an integrated system which has combinate by many different sensors, the most important component if this system is the satellites (Space-based ADS-B). Which it is the most common navigation systems around the world nowadays [3] [4]. That means all the aircraft can under a public surveillance system so that it can improve the efficiency and safety [1] [5]. Furthermore, ADS-B system has two difference subsystems, the ADS-B In and ADS-B Out. Those aircrafts which install ADS-B Out subsystem will not only present aircraft information but also submit other useful information such as route control information and notice to airman [6]. That means for each individuality in an ADS-B system, they can know everything including weather, notice and other data in their route or airspace [1] [7]. Under this circumstances, pilots or air controller will faster and more effectively, that means the air transport will more safety and efficiency. Additionally, it will reduce the costs and the crash rate of air transport [5] [8].

But on the other hand, according to Brian Prince's article in Security week, ADS-B system may attacked by spoofed ADS-B data since all the ADS-B data are neither encrypted and authenticated [9]. Another researcher group from EURECOM also proof that for ADS-B data, it is easy to forge since it lack some basic security mechanisms such as message authentication or encryption [10]. Furthermore, since the data will all transmitted by electromagnetic, so the aircraft output port and ground station port are two high-risk that needs to protect. According to a paper from US Air-force Institute of Technology, the intercept and disrupts are two common and easy ways to attack the ADS-B system. They are all based on disturb or hijack communication channel from emitter or receiver [11]. Nowadays it already has some solving method to protect the ADS-B data. The EURECOM researchers' study shows that the spoofed message can rejected by using authenticated equipment and adding a unique digital signature on the massage (such as Simple Public Key Infrastructure (SPKI) certificates or encryption) [10] [12] [13].

For all ADS-B data, in order to increasing its accuracy and reduce the error, some aircraft tracking algorithm are utilised. This report will focus on implement an aircraft tracking algorithm that based on Kalman Filter, Extend Kalman Filter and other basic location algorithm.

2.2 Aircraft Tracking Algorithm

Aircraft tracking algorithm is the most important part of ADS-B system and also it is the core of this report. For a normal aircraft tracking algorithm, it can be divided into following parts:

Aircraft localization

In order to locate an aircraft, there is only one way to do that, that is the multiliterate. It is to use the distance between the receiver and transmitter to calculate the exact location [14]. As it is universal acknowledge that if one distance is got, then a circuit can got around the point. In this method, the airplane must on some position on the circuit. For a 2-dimension value, since only three circuits can determine a position on a plane. So, for a two-dimension value, it needs 3 receives to determine a point. Additional, for a three-dimension value, it is similar that 4 circuit can determine a position. The following picture is a schematic diagram of quadrilateral positioning.

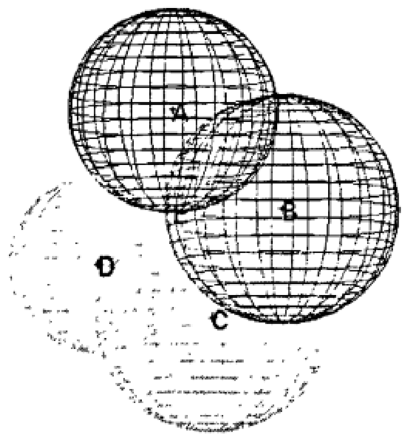


Figure 2.1 Schematic diagram of quadrilateral positioning [15]
A, B, C, D are representing four difference sensors in a 3D space. The intersection of these four spheres is the position of aircraft.

This is a common localization method, in real daily using, all the distance was based on the time difference of echo wave (that is the Time of Arriving (ToA) method). Since it will be influenced by many conditions, such as the corresponding time of system [16]. So, in this situation, another method called time difference of arriving (TDoA) used here to reduce the effect of server time delay and corresponding delay. In ADS-B system, the number of satellites and ground stations usually more than 4, that means the data is redundancy here. The first 4 data can use to estimate the position (for an airplane that its altitude has already knows, then it just need first 3). Then, the rest of them can use to improve the accuracy of it by using least squares, k-nearest neighbours or other method.

Aircraft localization by using GNSS data

The GNSS data is provided by the airborne measurement system, the location data is from a GNSS (usually the GPS or Aireon data) that is the Space-based ADS-B [17]. It will receive the echo wave of the aircraft and then calculate the geometry distance between the stellate and aircraft by using the time difference between the wave arriving and transmitting [18].

The advantage of this method is that it has a high accuracy. Nowadays the GNSS all have a high-resolution ratio (about 5 meters for GPS, 10 meters for Beidou and 1 meter for Galilean in open area on civil field) [19] [20] [21] [22]. So, it is very suitable to use the GNSS to locate aircraft here.

But the GNSS still have some shortages, since it is based on the navigation system, so once the airplane has met a situation such as the radio failure, it cannot receive the echo wave and broadcast the aircraft information using airborne equipment. Another shortage is in some specific environment (cumulonimbus, another airplane nearby, satellite communication failure), the accuracy will be reduced since there are some noise of obstacle near the airplane [23]. Under this circumstance, the ground-based ADS-B system, that is use radar and sensor data is a very good supplement of the space-based ADS-B.

Aircraft localization by using radar and sensor data

This is a kind of original tracking method that developed by using radar, lidar, sensor and some other ground station equipment. Same to the GNSS method, some of them use TDoA method to estimate the airplane. But since the ground radar always limited by the atmospheric refractive index and earth curvature, so it didn't have a long detect distance [24]. In this situation, it is very imprecise to use TDoA method since this method is based on time and speed of light [25]. Another method called received signal strength indication (RSSI) has been introduced for ground-based ADS-B.

The strength of this method is that it used signal power indication instead of using time. By using this method, the receiver can measure the power of receiving signal instead of time difference, so that it can reduce the error since the order of magnitude of time are always on nanosecond and the power indicator will much better [26].

On the other head, though the RSSI method can fix the time delay, it still has some defect. The Electromagnetic power loss will not always have the logarithmic relationship with distance [27]. Furthermore, same as the TDoA method, it needs 4 sensors to locate a three-dimension objective. The rest data of sensors can use as the revision data here.

In the database that used in this project, sometime it just have 2 sensors, so in this situation, it needs an advanced method to revise the error of data, so in this situation, since Kalman Filter and Extended Kalman Filter have able to use a former value and serval measurement data to estimate the current value [28]. It is a good method to correct and predict.

Correct and Prediction

As it just shows in the above, Kalman Filter (KF) and Extended Kalman Filter (EKF) are two possible method to correct and predict. By referencing Simo's book, it is clear to see that both the KF and EKF have two steps, predicting and updating [29]. The predicting step is to use the former data to predict the current data (in this project, the data is the position and velocity of aircraft) [30]. Then the update step is to use other measurement value, the covariance of the real state matrix and other parameters to estimate the current value. It is based on linear algebra and hidden Markov Model and can be represented by the following figures [31].

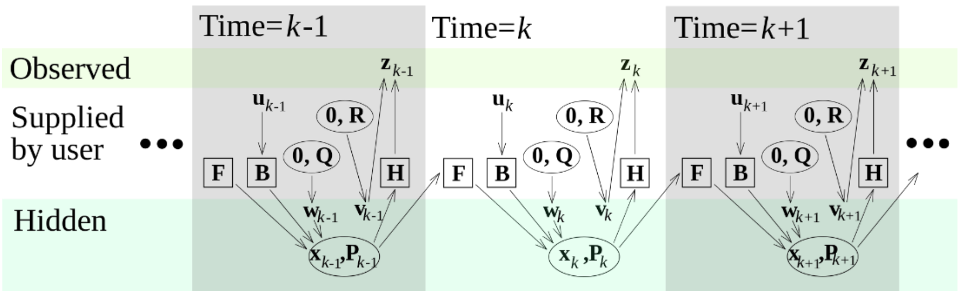


Figure 2.2 Markov model of KF [32]
This is the basic model of Kalman Filter, the parameters that rounded by square are the matrices. Then the parameters that rounded by ellipse are the covariance of multivariate normal distribution.

Since the KF included all the noise (including system error and measurement error), and have a covariance to fix that, so theoretically, if a KF is used in the tracking system, then it can get a more exact position that needed since it will consider all the errors by using the covariance of a multivariate normal distribution.

Additionally, it still has some shortages that it cannot be ignored. As the above figure shows, there are many difference parameters, the state-translation matrix – F; the observation matrix – H; the covariance of the process noise – Q; the covariance of observation noise. Each parameter matrix has difference components [29]. In a Kalman Filter, there are many difference parameters that needs confirm.

A possible method to confirm the right parameters is to use the neural network. For difference sensor, it has difference errors, under normal circumstance, it will provide by the manufacturer. But as it is known, the manufacture's data usually is a range. So, the neural network can fixed parameters by identified many aircraft, keep training and compare them with the correct one [33] [34]. It is too hard to construct and training a neural network since it didn't have enough data here. So, in this essay, it will no longer discuss about this.

EKF is similar to the KF, the only difference is that KF is based on linear Bayesian Statistics, so it used on a linear system, for example, it is suitable for the GPS data. And EKF is used on a non-linear system, so it is suitable for the RSSI data in this project. Besides, the EKF will use a Jacobian matrix to instead of the linear observation matrix and covariance of the process noise [35] [36].

For further smoothing and correcting, a RTS Smoother can used here, difference to the KF, the RTS Smoother is a kind of backward recursion method, that it can use the last value got by Kalman Filter to estimate the former one [37].

Optimization

As it is just shown on above, there are many difference methods to optimize the position data that it has just got. On GPS, it will use the least square method to estimate the final location. For RSSI, same method is used in that. But since the artificial intelligence is developed faster and faster, then some method used in neural network can be also used here, for example: k – Nearest Neighbour (k-NN) method. This is one of the simplest neural network methods.

The advantage of this part is that it can estimate and tracking the more correct, since not only the Least Square and the k-NN method, they are all the based on all the sensors that used to measurement. So, the result can reflect the location more realistic [38]. But it is also having some shortages, if the measurement sensor or training data is not enough, then it will not have a very good correcting effect. On the other hand, if it has too much data needs deal, then it is a big challenge for the robust and preference of a server [39].

Technically, KF and EKF are also two kinds of optimization method, but in this project, since the KF and EKF are used for correcting and predicting, so here it will not classify them as the optimization part of this project.

3 Industrial relevance, Real-world Applicability and Scientific/societal Impact

3.1 Industrial Relevance and Real-world Applicability

There are many difference usages for this air-tracking algorithm, but most of that are on aviation and air navigation field. As the above shows, all these algorithms are based on ADS-B data, so the most application in real-world is on air traffic since the KF and EKF can estimate the further route. So that means the air controller can make their decision more effective. That means it can reduce the time interval between two aircraft landing, take off or passing since all the routes and data are visible and predicable [40].

Additionally, in some area like the aircraft is fly over the ocean or someplace that the radar and satellite cannot covered. Here, the tracking algorithm has played an important role on this area. If the sensors or satellites didn’t have enough quantities to identify the position dependently. So, the KF and EKF can use the former one to estimate the next one. That is in this area, it can use the final determined position as the initial position, then using the KF, EKF method and measurement data to estimate the current position. Furthermore, by considered the first determined position after these unknown areas, then an RTS Smoother can be applicated so that it can get a more accuracy estimation.

In the real world today, more and more aircrafts are flying over two main difference ocean – the Atlantic Ocean and Pacific Ocean. As it shows on the following figure are this method used in a famous air-tracking software Flightradar24. This means the air-traffic on these skies are more and more crowded. Since all these two areas are mostly no man’s land and there are no sensors. So, all positions of aircraft will only be located by satellites. The satellite location usually has some delay and errors. Under this circumstance, the tracking route implemented by KF and EKF (usually KF since the satellite data will provide position coordinate directly) will always reduce these significantly which means it can plan the route more effective [41]. Furthermore, since all the data can be visual able and KF, EKF has been used. So, in further, it can reduce the air crash loss rate, since once the aircraft didn't broadcast, then the air-controller can find, report and take some relevant measures early.

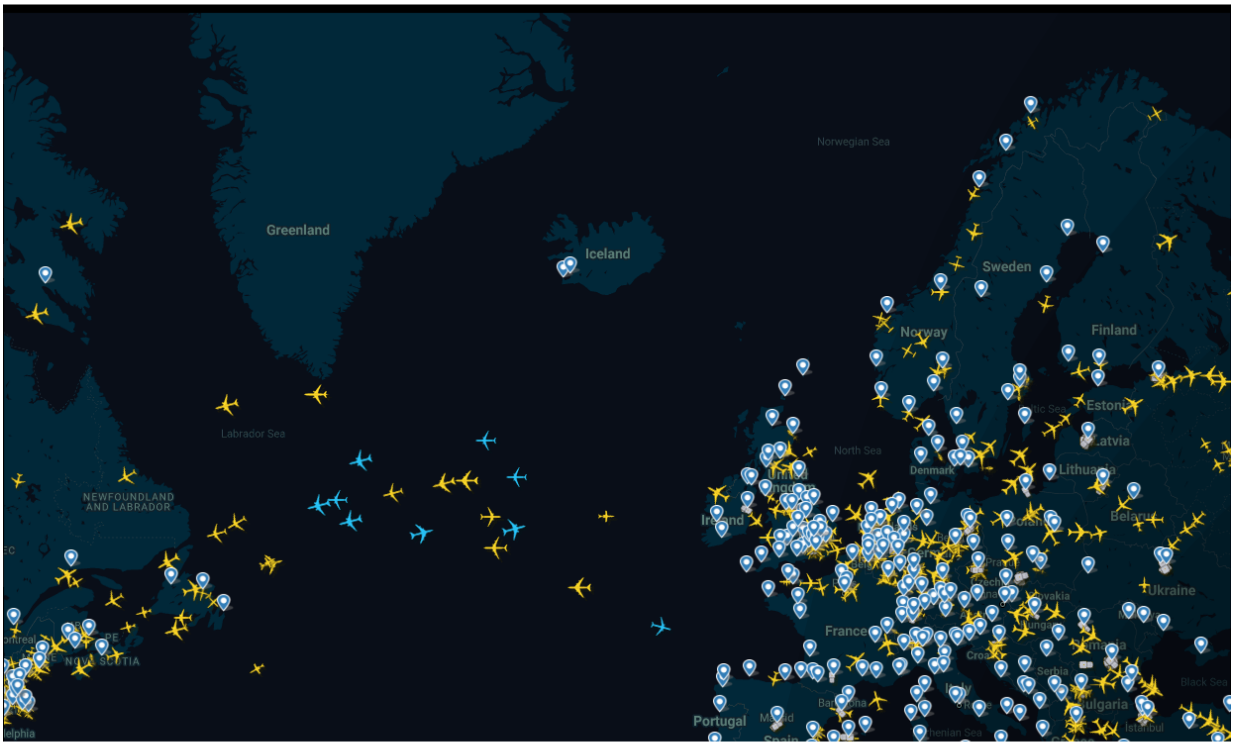


Figure 3.1 Flightradar24 Screenshot
The blue aircraft shows on the screen are the location of aircraft which depended on space-based ADS-B and advanced optimize algorithm. The yellow one over the ocean are used former ground-based ADS-B and inertial navigation.

3.2 Scientific/social Impact

As a useful supply of ADS-B, since ADS-B has been utilized more and more in today's air-traffic, for example, the US government requires all commercial aircraft and some aircraft which needs to flying in an area that requires a transponder before 2020 [7] [42]. In additional, Canada and Greenland's aviation administration department also start using the ADS-B to surveillance some remote area around Northern Canada and Greenland since 2009 [43] [44]. This makes airways can arrange more and more aircraft and the piolet can choose nearest route so that it can increase the income of the airways can reduce the energy consumption [43].

There are thousands of aircraft flying on the sky every day. So, if every aircraft is equipment ADS-B component, then it will take a considerable economy benefit. But as the above says, the qualities of an ADS-B system are highly depending on the accuracy of aircraft tracking algorithm. Which means a good aircraft tracking algorithm is important for an ADS-B system. So, in this project, a new kind of algorithm that not only suitable for space-based ADS-B, but also suitable for the ground-based ADS-B; not only suitable for pure GPS but also suitable for RSSI and TDoA data will be developed. That it will decrease the errors of the original ADS-B data and which is necessary for an ADS-B system.

Since this is an aircraft tracking method. So, for further research and scientific, it cannot be used on aircraft tracking. As it shows, the ADS-B is a totally public and broadcast system. That means in future, it can develop to an automatic piolet system since nowadays the ADS-B can replace some radar and air-controller. Meanwhile, this method can also transplant to another field like the robot. In Simultaneous localization and mapping (SLAM) system of a robot, this method can used to located the robot's position and predict another robot's location [45]. So that the robot can know its position itself. Furthermore, it also can used in automatic driving system and automatic vehicle. This may cause an important impact of society and science. And all of this are based on a good aircraft tracking algorithm.

4 Theory

4.1 Kalman Filter (KF) and RTS Smoother

Kalman Filter is a linear gaussian recursive filter that based on Bayesian Statistics. The core of Kalman Filter is the multivariable normal distribution. That it is considered x – the real location and y – the measurement data as two normal distribution. Besides, all the noise will also be considered as a normal distribution. It has been used in many difference situations, the automatic drive, artificial intelligent, neuroscience and microeconomic. The following is the model of Kalman Filter:

$$\boldsymbol{x_k = A_{k-1}x_{k-1} + q_{k-1}} \tag{4.1.1}$$
$$\boldsymbol{y_k = H_k x_k + r_k} \tag{4.1.2}$$

Notice that here the $\boldsymbol{x_k \sim N(m_0, P_0)}$ and $\boldsymbol{y_k}$ are the state and measurement, $\boldsymbol{q_{k-1} \sim N(0, Q_{k-1})}$ is the process noise; $\boldsymbol{r_k \sim N(0, R_k)}$ is the measurement noise; is the $\boldsymbol{A_{k-1}}$ transition matrix and $\boldsymbol{H_k}$ is the measurement model matrix of the dynamic model.

In order to find the parameters of the Kalman Filter, it can use the following prediction step and update step to calculate that [29]:

Prediction Step:

$$\boldsymbol{m_k^- = A_{k-1}m_{k-1}} \tag{4.1.3}$$
$$\boldsymbol{P_k^- = A_{k-1}P_{k-1}A_{k-1}^T + Q_{k-1}} \tag{4.1.4}$$

Update Step:

$$\boldsymbol{v_k = y_k - H_k m_k^-} \tag{4.1.5}$$
$$\boldsymbol{S_k = H_k P_k^- H_k^T + R_k} \tag{4.1.6}$$
$$\boldsymbol{K_k = P_k^- H_k^T S_k^{-1}} \tag{4.1.7}$$
$$\boldsymbol{m_k = m_k^- + K_k v_k} \tag{4.1.8}$$
$$\boldsymbol{P_k = P_k^- - K_k S_k K_k^T} \tag{4.1.9}$$

For Kalman Filter, it will recursed from $\boldsymbol{m_0}$ and $\boldsymbol{P_0}$, which it is the initial state of a Kalman Filter. Here, the $\boldsymbol{v_k}$ is the Innovation vector, $\boldsymbol{S_k}$ is the Innovation covariance and $\boldsymbol{K_k}$ is Gain matrix. Noticed that here other parameter can write as the following.

For all $\boldsymbol{x_k}$, in a 3D aircraft tracking method, it needs to write as a column vector: $\boldsymbol{x} = [x \quad \dot{x} \quad y \quad \dot{y} \quad z \quad \dot{z}]^T$, here it is consist by the location and the derivative of location, that it is the velocity. So, for other parameter, the following is their value. Assumed that in each step the parameter is same, and the time step is same at $\boldsymbol{T}$ too, then [46]:

$$\boldsymbol{A = I_2 \otimes \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}}, \quad \boldsymbol{Q = qI_2 \otimes \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix}}$$

$$\boldsymbol{H = I_2 \otimes [1 \quad 0]}, \quad \boldsymbol{R = I_2}$$

Here the ⊗ is Kronecker Product. For an tracking algorithm, the parameters can write as [29] [46]:

$$\boldsymbol{A} = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{Q} = \begin{bmatrix} \frac{q_1T^3}{3} & \frac{q_1T^2}{2} & 0 & 0 & 0 & 0 \\ \frac{q_1T^2}{2} & q_1T & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{q_2T^3}{3} & \frac{q_2T^2}{2} & 0 & 0 \\ 0 & 0 & \frac{q_2T^2}{2} & q_2T & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{q_3T^3}{3} & \frac{q_3T^2}{2} \\ 0 & 0 & 0 & 0 & \frac{q_3T^2}{2} & q_3T \end{bmatrix}$$

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad \boldsymbol{R} = \begin{bmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{bmatrix}$$

To further recursion, the RTS Smoother can use here, similar to Kalman Filter, it is based on Bayesian Statistics, but difference to that, it is a kind of backward recursion which use the next one to estimate the former one. Additionally, the RTS Smoother will use the value that based on Kalman Filter. The following is the backward recursion method of RTS Smoother [37] [47]:

$$\boldsymbol{m}_{k+1}^- = \boldsymbol{A}_k \boldsymbol{m}_k \tag{4.1.10}$$
$$\boldsymbol{P}_{k+1}^- = \boldsymbol{A}_k \boldsymbol{P}_k \boldsymbol{A}_k^T + \boldsymbol{Q}_k \tag{4.1.11}$$
$$\boldsymbol{G}_k = \boldsymbol{P}_k \boldsymbol{A}_k^T [\boldsymbol{P}_{k+1}^-]^{-1} \tag{4.1.12}$$
$$\boldsymbol{m}_k^S = \boldsymbol{m}_k + \boldsymbol{G}_k [\boldsymbol{m}_{k+1}^S - \boldsymbol{m}_{k+1}^-] \tag{4.1.13}$$
$$\boldsymbol{P}_k^S = \boldsymbol{P}_k + \boldsymbol{G}_k \left[\boldsymbol{P}_{k+1}^S - \boldsymbol{P}_{k+1}^-\right] \boldsymbol{G}_k^T \tag{4.1.14}$$

Cooperated with the GPS data, the Kalman Filter can get a more correct value of the position. For some place that it didn't have any position data, then it can also use to estimate that. RTS smoother, as a backward recursion method, it is use to optimize the value got by Kalman Filter. Here the $\boldsymbol{G}_k$ is the gain matrix in RTS Smoother.

4.2 Extend Kalman Filter (EKF) and Received Signal Strength Indication (RSSI)

For some non-linear gaussian system, EKF is a good method to solve this problem. In EKF, the transition matrix and observation matrix can represent as the differentiable function. And they can calculate by using Jacobian matrix, here the following is the expressions of Extended Kalman Filter:

$$\boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1}) + \boldsymbol{q}_{k-1} \tag{4.2.1}$$
$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{r}_k \tag{4.2.2}$$

Notice, here the $\boldsymbol{f}(\cdot)$ is the dynamic model function and $\boldsymbol{h}(\cdot)$ is the measurement model function. In this project, since the $\boldsymbol{x}_k$ are the position and give by the database or recuse from the former one. So, here the $\boldsymbol{x}_k$ didn't have any function, which means here it will use the $\boldsymbol{x}_k = \boldsymbol{x}_{k-1} + \boldsymbol{q}_{k-1}$. In this situation, in order to find each parameter in this method, the following is two steps that used here [35]:

Prediction Step:

$$\boldsymbol{m}_k^- = \boldsymbol{A}_{k-1} \boldsymbol{m}_{k-1} \tag{4.2.3}$$
$$\boldsymbol{P}_k^- = \boldsymbol{A}_{k-1} \boldsymbol{P}_{k-1} \boldsymbol{A}_{k-1}^T + \boldsymbol{Q}_{k-1} \tag{4.2.4}$$

Update Step:

$$\boldsymbol{v_k} = \boldsymbol{y_k} - \boldsymbol{h}(\boldsymbol{m_k^-}) \quad (4.2.5)$$
$$\boldsymbol{S_k} = \boldsymbol{H_x}(\boldsymbol{m_k^-})\boldsymbol{P_k^-}\boldsymbol{H_x^T}(\boldsymbol{m_k^-}) + \boldsymbol{R_k} \quad (4.2.6)$$
$$\boldsymbol{K_k} = \boldsymbol{P_k^-}\boldsymbol{H_x^T}(\boldsymbol{m_k^-})\boldsymbol{S_k^{-1}} \quad (4.2.7)$$
$$\boldsymbol{m_k} = \boldsymbol{m_k^-} + \boldsymbol{K_k}\boldsymbol{v_k} \quad (4.2.8)$$
$$\boldsymbol{P_k} = \boldsymbol{P_k^-} - \boldsymbol{K_k}\boldsymbol{S_k}\boldsymbol{K_k^T} \quad (4.2.9)$$

Here, the rest parameters are same as the former one, except the measurement model matrix $\boldsymbol{H}$. Here, it will be replaced by the Jacobian matrix of the measurement function, the $\boldsymbol{H_x}(\cdot)$. Since other parameters are same as that in KF, so it will no longer to show that here.

The principle of EKF is similar to KF that just show above. The only reason is that the measurement function will be replaced by the first order derivative. That it is the second term of a Taylor series. The rest of the Taylor series will not use since they are too small so that they can ignored.

Notice that in this method, the $\boldsymbol{x_k}$ is same as the original one, that it is use the position and the derivative of it that is the velocity as a column vector. But here the has a little bit change, since the EKF is a non-linear Gaussian model, so it is very suitable to use for RSSI method. That it is use RSSI data to estimate the position.

The RSSI method is to use the power strength indicate, that it is represent the receive power as a logarithmic function of distance, and by measurement the receive power indicate, it can calculated the distance, the expression can express as the following [48]:

$$h\left(x_k, m^j\right) = P_0 - 10\gamma \log_{10}(\sqrt{\|p_k - m^j\|^2}) \quad (4.2.10)$$

Here, the P_0 is the received power at a reference distance (usually at 1 meters), γ is the path loss exponent and m^j is the position of the jth beacon. To calculate the measurement function, it just needs to calculate the Jacobean matrix of $h\left(x_k, m^j\right)$. It will show on the following part. Notice that on above, the $\left\|p_k - m^j\right\|$ means it is the Euclidean norm of $p_k - m^j$. That it is the Euclidean distance of that.

Both the KF and EKF can simplified to use only the position as the column vector of x_k. The function will a little bit easy if we used that here.

4.3 Time of Arriving (ToA) and Time Difference of Arriving Algorithm (TDoA)

As the above shows, not only the time of arriving, but also the time difference of arriving method is based on the receiving and transiting time. But they still have some difference, the ToA is an algorithm that will only use the time of one sensor for one equation, not the time difference of two sensors which means it may cause higher error, so it will no longer to show the method here. The TDoA is a method that it will combine two sensors data in one equation, in this method, the result will not be influenced by the delay time. Since the server may have delay time and error. Here, the following is the method of TDoA:

Basic method can be developed by the radar distance measurement and distance formula in a Cartesian coordinate system. That the following is the function that it can be used:

Assumed that the Cartesian coordinate of aircraft is (x, y, z), and the Cartesian coordinate of a sensor is (x_s, y_s, z_s)

$$d = ct \quad (4.3.1)$$
$$d = \sqrt{(x - x_s)^2 + (y - y_s)^2 + (z - z_s)^2} \quad (4.3.2)$$

Simultaneous (4.3.1) and (4.3.2), the following function can get:

$$ct = \sqrt{(x - x_s)^2 + (y - y_s)^2 + (z - z_s)^2} \tag{4.3.3}$$

This is the function of ToA method, by simultaneous the above as an equation set by using 4 difference position, an exact position can get. Additional, if two sensors have been considered in one equation, that it is called the TDoA method, the following is the function of that, noticed that here the (x, y, z) is the coordinate of aircraft and (x_{s1}, y_{s1}, z_{s1}), (x_{s2}, y_{s2}, z_{s2}) are the coordinate of sensor. The following is the method that it used here [49] [50]:

$$c(t_1 - t_2) = \sqrt{(x - x_{s1})^2 + (y - y_{s1})^2 + (z - z_{s1})^2} - \sqrt{(x - x_{s2})^2 + (y - y_{s2})^2 + (z - z_{s2})^2} \tag{4.3.4}$$

By combined 4 difference sensors to get 3 equation, then it can get the sure position. Since it combined 2 sensors in 1 equation. Then if these 2 sensors are same type or they have similar server delay time and error. By using this method, it can reduce the influence of these delay and error [51]. For more than 4 sensors, it needs further optimize, a good method that it can be use is the k-NN algorithm, it will show on the following.

4.4 k-Nearest neighbors Algorithm (k-NN)

K-NN is the simplest and the laziest neural network. The state of it will only determine by the nearest neighbours [50]. That it is means in normal k-NN method, it will just consider some of the neighbours and then use them to recursion and adaption. But in this project, since the maximum sensor are only 7, so it will consider all the sensors data here. That it will not have the classify step, just have the recursion step. And the weight for the neural network, for some sensors that it is near the position, it will have the higher weight, that it will infect more on determine the position [52] [53].

For the initial value of position, it will use the geometric centre of polygon that consisted by the other location point that determined by difference redundancy sensors. By give difference weight to difference location point (the shorter distance, the higher weight). Since 4 sensors will determine a position, then, it can get that for an aircraft that have 7 sensors measurement data, it will have $C_7^4 = 35$ difference location point. The following is a figure that used to explain this.

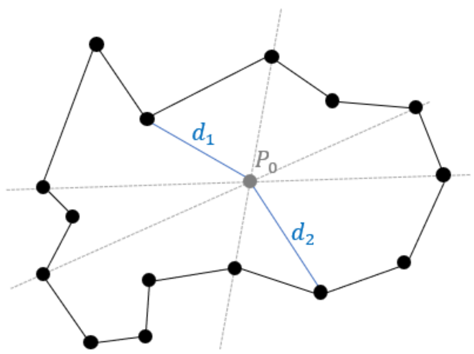


Figure 4.1 the method explains

In this picture, the P_0 is the initial value (that it is the geometric centre of the polygon), here since $d_1 < d_2$, then for the k-NN which will build, the weight of these point w_1 and w_2 will have the following relations: $w_1 < w_2$

Here, the weight is determined by the Euler distance between two point, that it is $w_1 = \frac{1}{d_1}$. And the neuron receives signal of this kind of method is a distance that it is determined by errors of the system and measurement.

5 Design

5.1 Overall method

The total aircraft tracking method are consisted by two part, the first is the location determination part, which is use space based and ground-based ADS-B data to determine the position data. This will usually use time based multiliterate, which is the TDoA method as it just mentioned. This is the basic location and optimize part. In this method, it will only use it to estimate the initial position for an unknown aircraft.

Next it is a correcting and predicting method. Here it has been divided into two difference method, the first is use the Kalman Filter that will use for the GPS data. The GPS data will provide the location method directly. The second is the Extended Kalman Filter, it will use the RSSI data since this will just provide a indicate of power. By using EKF, it can use the RSSI data that the sensors provided directly to estimate the position. As the above mentioned, the GPS data is for space-based ADS-B and the RSSI is for ground-based ADS-B. this is a further optimized part. But in this method, it only needs to determine the first location by using the basic location and optimize part that the above shows, and then by using KF and EKF method, the following positions can be inferred.

Finally, part is using RTS Smoother to smooth the result and make it more accuracy. The RTS Smoother can smooth not only the result of KF but also the result of EKF since both the KF and EKF will finally get a value that it is related with the position. The RTS is a backward recursion method of KF. For EKF, since the result of that is also the position, which can use the RTS directly since it is just related with position and will not use other parameters. This is a further estimation, which can make the result better and conform to the fact.

The following is a schematic diagram that can explained the overall method here:

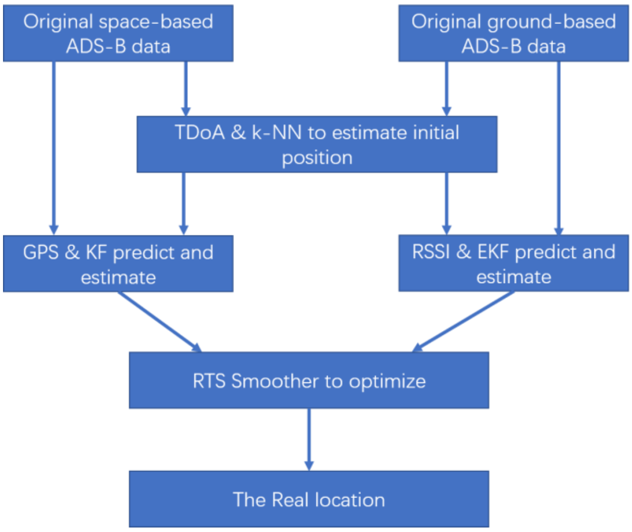


Figure 5.1 A simple schematic diagram of this method

As it can be seen, in each kind, it has two different method, the first one is to use the KF or EKF directly, and the other is to use TDoA and k-NN to estimate first. This is because in this method, the data it used is from real airplane data. that means in the database, sometime the location data will be provided directly, which means for these data, it can use as an initial data directly and the method will just need to use KF or EKF to estimate. But for other airplane that it didn't have any position data, then it will use the TDoA and k-NN method to estimate the initial location first, and then estimate by the KF or EKF (usually EKF, the KF will estimate from former location directly).

5.2 KF and satellite tracking data

As the above shows, the KF is a kind of recursion filter that based on Bayesian Statistics. KF is used for linear Gaussian System. In air tracking data, the GPS data which is the coordinate of an aircraft in some timestep, can use KF to estimate, since the measurement position can determine the estimation position linearly. So, for this method, the KF will use to estimation the GPS data directly. For the KF, x is the real state, for this method, it is a 6-dimension vector, which consist by the 3 components of aircraft coordinate, and the resist 3 are the components of velocity. y is the measurement, the form of it is totally same as x but the only difference is all the data here are got by measurement not the real state. The original database didn't have the velocity information (in real world, the ADS-B will include this information, to estimate the velocity, an adaption function has been set up, it will take later).

The figure in right is the flow chart of this part. Notice that it just shows one circle of the whole loop here. First is the velocity calculation, this is based on the measurement data (here the measurement data is the data that the database gives to us). Then it is inserting the initial data and measurement data, here the initial data will use the first location information. Next since the Kalman Filter have a high influence by measurement data since it has been based in that. So, before that, it needs to add a pre-filter model to remove the wrong data. Here it set a 99% confidence interval. The original data is a discrete set, so here a polynomial function has been used to curve these discrete points and get a function. Then the 99% confidence interval are based on this adaption function. Then by using this method, the suitable data can get. Then these measurement data will use KF to get the real state estimation. After that, a plot module is used to let these data become visible and then output them.

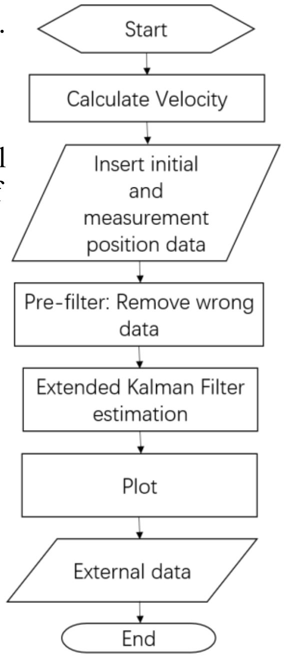
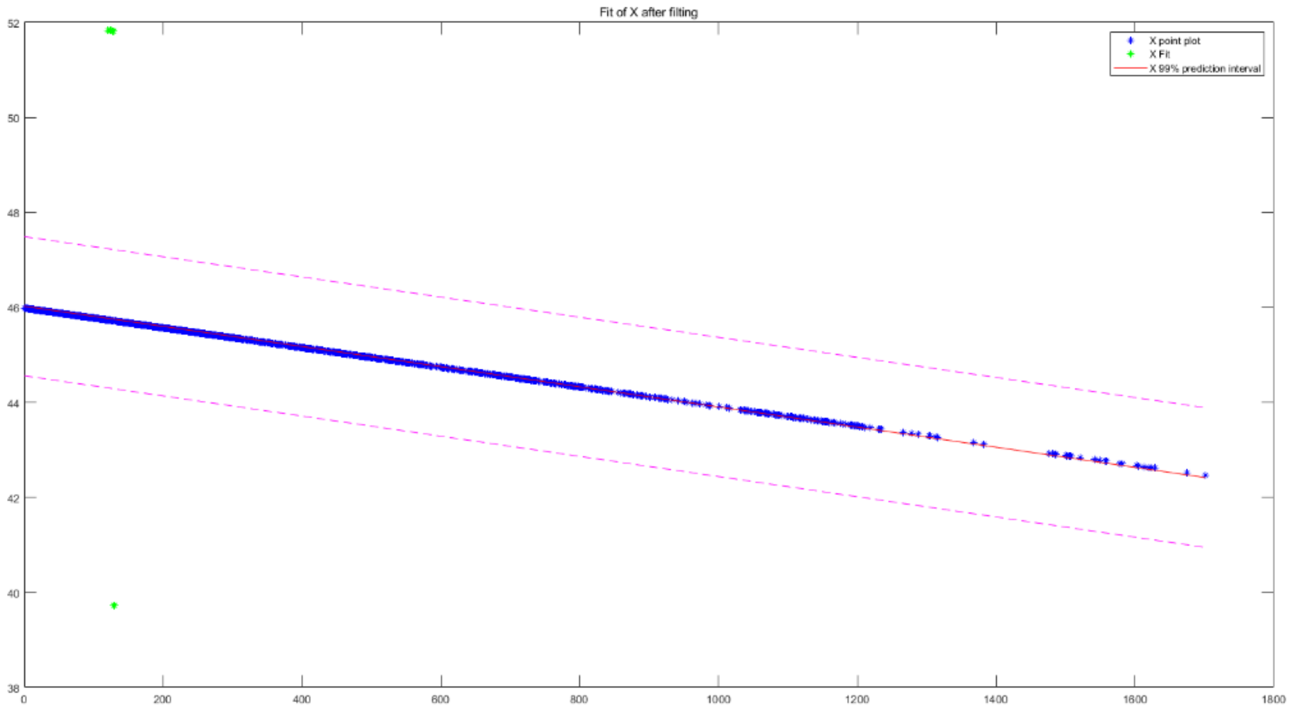


Figure 5.2 Flow chart of KF, GPS method

The following is the fit function plot output of this method:



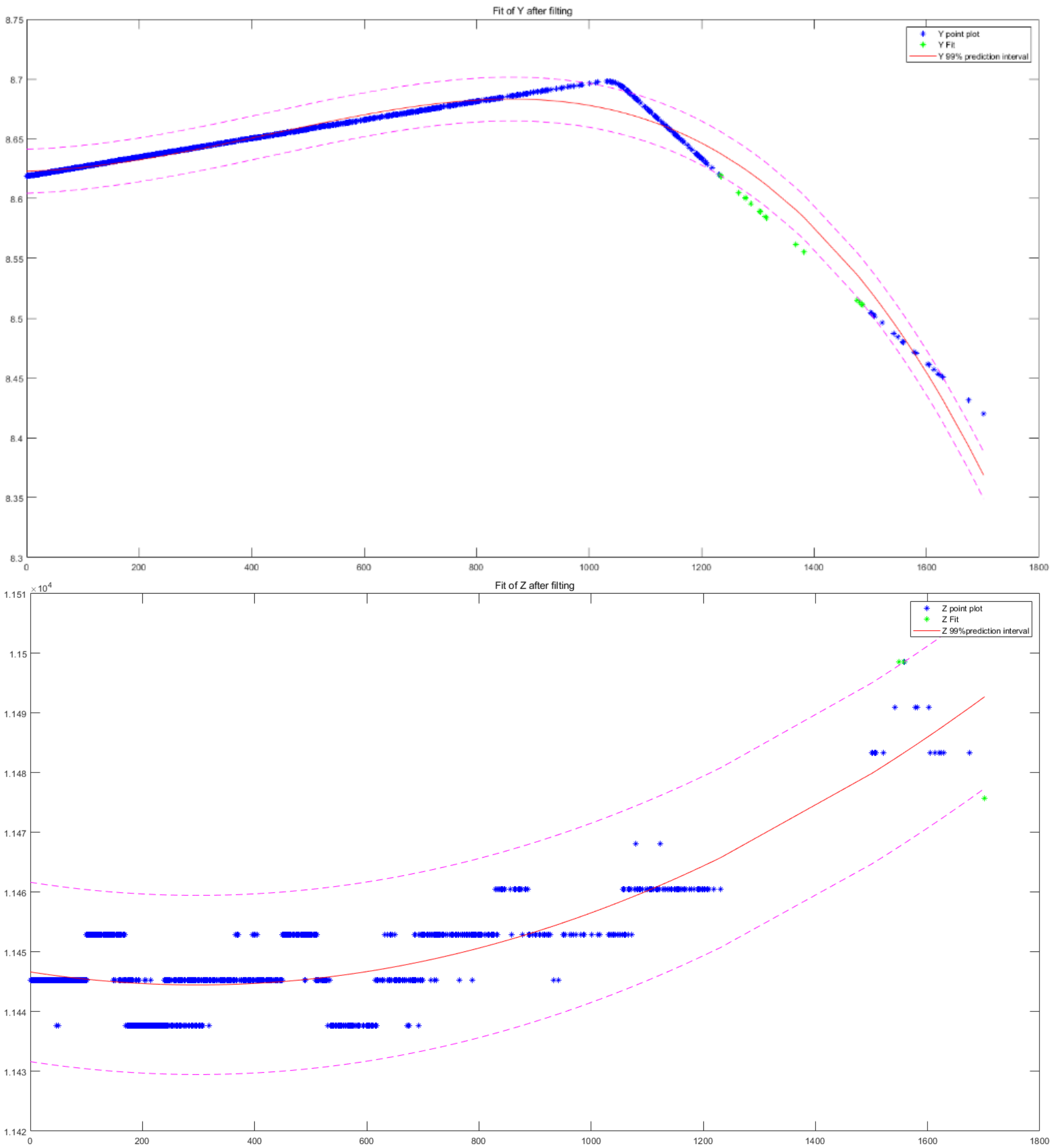


Figure 5.3 the fit function figure
The blue point is the point that required (didn't have any mistake), the green point is the point that didn't required. The central red line is the regression function figure and the two-pink line between that are two confidence intervals.

For the pre-filter, as it shows here, some point that have obviously error has been removed here. In this method, it didn't remove the error point once since MATLAB didn't have able to create a regression function of 4 different variables (the three coordinates of location and time). So, in this method, first it removed some points that the horizontal ordinate (x) did not meet the requirement, then it is the point that longitudinal coordinates (y) did not meet the requirement and finally those which the vertical coordinates did not meet the requirement.

5.3 EKF and RSSI data

The RSSI is a power strength indication, which is represent as dB form, that it usually uses for two parameters which they have an exponential relationship. In this situation, since the KF is used for linear Gaussian system, so it will no longer suitable to sue here. In order to solve this problem, a new kind of method that it is based on EKF has been used. Same as the KF, the x is the state that it is a 6-dimension vector which is consist of 3 different components of position of an aircraft and 3 different components of velocity. But the y is difference with that, here the y is the RSSI value of a sensor, so the measurement model matrix $\boldsymbol{H}$ is difference with the Kalman Filter. But for other parameter, since the position that used to estimate current state will directly from the last state, so the transition matrix $\boldsymbol{A}$ is same as that in KF.

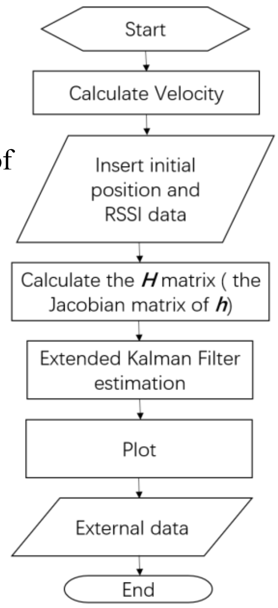


Figure 5.4 Flow chart of EKF, RSSI method

The figure on the right shows a flow chart of EKF in this method. Here, it only shows one loop circle of the complete integration method too. First it is the calculation of velocity, that it is same as that in KF. But in the real construct of EKF, since the final result that need to show visible on the screen is the position of an aircraft. So here the velocity and its related components have been deleted from the model. That means in real construct, it will no longer to calculate the velocity here. Then, the next step is same as that in KF. Which is insert the essential data, in here it is the initial position and RSSI data since here the measurement is the RSSI rather than the position data in KF. Next, since in EKF, the H has been changed to the total derivative of the exponential path loss model h, so it needs to calculate the H matrix here by using the Jacobean matrix of h. Finally, it is same to KF that it is using the EKF to estimate, plot the route, external and visualize the position data.

To get an accuracy result, two parameters in RSSI exponential path loss model must be got, the first is P_0, the received power at a reference distance (usually at 1 meters), this is usually got by the manufacturer's instructions. So, it has been designed already, and for each difference, it has a different P_0. The other one is the path loss exponent γ. In ordinary signal and electromagnetic wave transmission's SNR model, the power of it is proportional to the square of its amplitude, so it has a related coefficient of 2. So, in here, it will use $\gamma = 2$ as the path loss exponent here [54].

5.4 TDoA and k-NN joint location algorithm

This is a preliminary location algorithm by using ADS-B data (both Space-based and Ground-based). The TDoA method is an advanced multiliterate method that it is use the time difference of arriving of two difference sensors. Theoretically, for one position, it needs 4 difference sensors to locate. In ToA, this will just use on the initial value of that, since for other position, it can set to choose the value that it is near the initial value. Under this circumstance, it just needs 3 difference sensors to determined one position. But here since the time in the database have a higher delay, so it will use TDoA method. Which means for any position, it all need 4 sensors to determine the position. The determine matrix can write as the following:

$$\begin{bmatrix} d_1 - d_4 \\ d_2 - d_4 \\ d_3 - d_4 \end{bmatrix} = \begin{bmatrix} \|m^1 - m^4\| \\ \|m^2 - m^4\| \\ \|m^3 - m^4\| \end{bmatrix} = \begin{bmatrix} \sqrt{(x_1 - x_4)^2 + (y_1 - y_4)^2 + (z_1 - z_4)^2} \\ \sqrt{(x_2 - x_4)^2 + (y_2 - y_4)^2 + (z_2 - z_4)^2} \\ \sqrt{(x_3 - x_4)^2 + (y_3 - y_4)^2 + (z_3 - z_4)^2} \end{bmatrix} \tag{5.4.1}$$

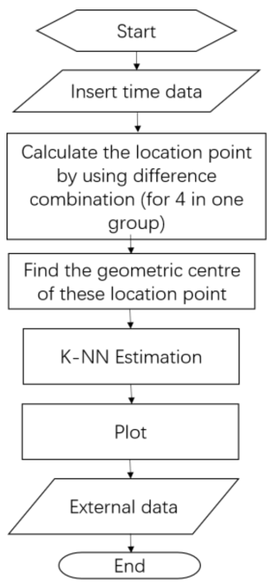


Figure 5.5 Flow chart of TDoA, k-NN method

As the flow chart on the right shows, here the method will start with insert the time data, that it is the original transmit time of difference sensors and the receiving time. By using these methods, it can get the transmit time on the road. Next, it will calculate the location point by using the method above. By determined the location points, it need to find the geometric centre of that, for a k points finite point set, the coordinate of the geometric centre (x_c, y_c, z_c) can be write as the following:

$$(x_c, y_c, z_c) = (\frac{x_1+x_2+\cdots+x_k}{k}, \frac{y_1+y_2+\cdots+y_k}{k}, \frac{z_1+z_2+\cdots+z_k}{k}) \tag{5.4.2}$$

Then, it can use a k-NN network to estimate that by combined the weight and unit distance shifting. The last two steps are same as the above that is show the result on the screen as a figure and then output that data.

Notice here the coordinate that using here is the geocentric coordinate system since it can represented and calculated by the distance easier than any other coordinate system, but in order to use the KF or EKF, it needs to transfer the geocentric coordinate system to a geographic coordinate system (usually a perfect circle tangent plane coordinate system). The calculate of that will shows on part 7.

For many of the aircraft in the database, it just has two or three sensors data, so here it will no longer to coding this method in a computer since only 2 or 3 sensors cannot let the program run. Here it will just have a written and theoretical analysis and construct of that.

5.5 RTS Smoother optimization

Finally, it is a kind of backward recursion that it will use the last state and current state that estimated to optimize the current state. That means difference to KF and EKF, the KF and EKF is estimate from the front to the back, but RTS Smoother is from the back to the front, so that here the initial value may change to a more correct value. Same as the KF, it is used in a linear Gaussian system. For the reason it can use in an EKF is that after estimation, the result of an EKF are all positions too, that it is same as a result of KF. So, they can use the same method, that it is the RTS Smoother to estimate.

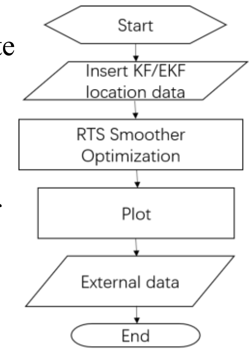


Figure 5.6 Flow chart of RTS Smoother method

The right side is a flow chart of RTS Smoother methods. Same as the above, here it will use the location data that estimated by KF and EKF, the velocity which has been got on the first to re-estimate and optimize. Getting the result and show that on a plot figure and finally output data.

Normally, the result of adding RTS Smoother will better and more accuracy than the result that only use KF/EKF. But sometime, due to the velocity error, the results will worse than that, here it can add a post-filter which have the same construct with the pre-filter in KF. that it is to fliting some excessive error value and let the final result more logical and suitable for the real-world route.

6 Experimental method

6.1 Aircraft Data Source and tools

As It shows on the above, all the aircraft data are real, they are from the Opensky database and was fly around the whole Europe. That means all the method that developed here will applicate and test directly on the real-world condition. All the errors and the mistakes that took in real world will also influence the estimation of this method too. To summary, the total experimental method to test the algorithm are all based on the natural and real condition.

The algorithm compile environment is MATLAB. As it known, the MATLAB language is similar to lots of other languages. It has a high robust and performance That means it can transplant and translate to other language fast and correct. And to test it, in MATLAB in can reduce the influence of language defect to the minimum.

So, for the tools and source method here, it is a software real-data-based method.

6.2 Other experimental needed data

But in the database, it has still lost some necessary information that used in this experiment. First, since the KF and EKF are all based on the normal distribution. That means all the covariance components are unknown at first. So here it is evaluated by the error that provided by the Opensky database, the aircraft velocity in real world and the step time [55]. The result of estimation will show on the code of the method.

For other parameter that didn't know, it will use many difference values to try or determine that by using general knowledge. For example, the P_0 of RSSI data, since it cannot find in any kinds of network, and for difference type, it has difference value. So, in here it just uses a common value and the value is get by trying.

As it shows above, for some location, there cannot be using software to coding since in the database, it didn't give them enough data to calculate, for EKF and RSSI method since it can adapted the filter itself, so to some missing value on building, it will simply adding a unit vector on that, and then estimate that. but for TDoA and k-NN, since the value are based on the coordinate, once it didn't have enough data. It cannot be calculated by adding unit vector, so that means it cannot run if it has been coding. So, in this situation, it can just have a theoretical analysis and mathematical modelling for that.

So, the method here to construct the difference model is that using software and theoretical analysis. And for parameter confirm some of that are calculated, but the rest of that are confirmed by trying.

6.3 The experiment method

This method is created on MATLAB, that means many works of that are finished by computer, but for the different parts or model build, it really has some method, the following is the methodology for experiment to developed this algorithm.

1. Study the KF, review the basic knowledge of that (MATLAB, Linear Algebra)

2. Construct, optimize and adjust a KF and RTS Smoother, adding some necessary sub-method on that, plot the estimate result and measurement

Algorithm 6.1 Prepare ---- insert, calculate velocity and pre-filter

Input: the original data location matrix x_{in}
Output: the state matrix included velocity and without mistake value x_s
Calculate the polynormal regression function of x_{in}, $x(t)$
Calculate the derivative of $x'(t)$
Get the matrix included all velocity x'_S
for *k1 = 1* to S **do**
    if the horizontal ordinate x of x'_{k1} is more than 99% confidence interval of x of $x(t)$
        Delete point x'_{k1}
    end if
    Get a new point set x'_{S1} that remove x errors
end for
for *k2 = 1* to S **do**
    if the longitudinal ordinate y of x'_{k2} is more than 99% confidence interval of y of $x(t)$
        Delete point x'_{k2}
    end if

Get a new point set x'_{S2} that remove *x, y* errors
end for
for *k3 = 1* to S **do**
if the horizontal ordinate *z* of x'_{k3} is more than 99% confidence interval of *z* of $x(t)$
Delete point x'_{k3}
end if
Get the final point set x_s
end for

Algorithm 6.2 Kalman Filter

Input: Approximated prior mean of x_s, m_s; the prior covariance matrix of x_s, P_s; the measurement y_s
Output: Approximated mean of x_s, m_s; the covariance matrix of x_s, P_s
Set $m_{s-1} = m_0$, $P_{s-1} = P_0$
for *k = 1* to S **do**
Calculate $m_k^-, P_k^-, v_k, S_k, K_k, m_k, P_k$ by using equation (4.1.3) - (4.1.9)
Save m_k, P_k in array m_s, P_s
$m_k = m_{s-1}$; $P_k = P_{s-1}$
end for
Get a location point set that estimated by KF $x_s = m_s$

Algorithm 6.3 RTS Smoother

Input: Approximated prior mean of x_{ss}, m_{ss}; the prior covariance matrix of x_{ss}, P_{ss} ; the state after KF x_s
Output: Approximated mean of x_s, m_s; the covariance matrix of x_s, P_s
Set $m_{s+1}^s = m_s$, $P_{s+1}^s = P_s$
for *k = 1* to S **do**
Calculate $m_{k+1}^-, P_{k+1}^-, G_k, m_k^s, P_k^s$ by using equation (4.1.10) - (4.1.14)
Save m_k^s, P_k^s in array m_{ss}, P_{ss}
$m_k^s = m_{s+1}$; $P_k^s = P_{s+1}$
end for
Get a location point set that estimated by KF $x_{ss} = m_{ss}$

Algorithm 6.4 Post dealing --- post-filter and export

Input: the data location matrix after RTS Smoother x_{ss}
Output: the state matrix location without mistake value x_{out}
for 1 to S **do**
Calculate the polynormal regression function of x_{ss}, $x_{ss}(t)$
end for
for 1 to S **do**
if the horizontal ordinate *x* of x_{ss} is more than 99% confidence interval of *x* of $x_{ss}(t)$
Delete point x_{ss}
end if
Get a new point set x_{ss} that remove *x* errors
end for
for 1 to S **do**
if the longitudinal ordinate *y* of x_{ss} is in 99% confidence interval of *y* of $x_{ss}(t)$
Keep point x_{ss}
else
Delete point x_{ss}
end if
Get a new point set x_{ss} that remove *x, y* errors
end for
for 1 to S **do**

if the horizontal ordinate z of x_{ss} is more than 99% confidence interval of z of $x_{ss}(t)$
    Delete point x_{ss}
end if
Get the final point set x_{out}
end for
Plot and output the matrix x_{out}

3. Construct optimize and adjust an EKF by using RSSI data, adding some necessary sub-method on that, plot the estimate result and measurement

Algorithm 6.5 Extended Kalman Filter

Input: Approximated prior mean of x_s, m_s; the prior covariance matrix of x_s, P_s; the measurement of RSSI y_s; Sensor Location x_{sen}
Output: Approximated mean of x_s, m_s; the covariance matrix of x_s, P_s
Set $m_{s-1} = m_0$, $P_{s-1} = P_0$
for k = *1* to S **do**
    Calculate m_k^-, P_k^- by using equation (4.2.3), (4.2.4)
    for *j=1* to m **do**
        Calculate h_j by using (4.2.10)
        Calculate the Jacobian matrix of h_j, H_j
        Calculate v_k, S_k, K_k, m_k, P_k by using equation (4.2.5) - (4.2.9)
    end for
    Get final m_k, P_k
    Save m_k, P_k in array m_s, P_s
    $m_k = m_{s-1}$; $P_k = P_{s-1}$
end for
Get a location point set that estimated by KF $x_s = m_s$

4. Construct a mathematical model for TDoA and k-NN

Algorithm 6.6 TDoA and k-NN

Input: the transmit time t_t; the receiving time t_r;
Output: Approximated location x_s
for *a=1* to m-4 **do**
    Get the sensor number of first equation
    for *b=2* to m-3 **do**
        Get the sensor number of second equation
        for *p=3* to m-2 **do**
            Get the sensor number of third equation
            for *q=4* to m-1 **do**
                Get the sensor number of fourth equation
            end for
        end for
    end for
end for
for *v = 1* to m **do**
    if *v = a or b or p or q*
        Get the equation part by using (4.3.4)
    end if
end for
for *n = 1* to 3 **do**
        Combined equation part n and 4 by using (4.3.4)
end for
Calculate the point location

Find the geometric centre by using (5.4.2)
Get the initial point position in geometric centre at $P_0\ (x, y, z)$
for *w* = *1* to C_m^4 **do**
    Find the weight between *w*th point and geometric centre $\frac{1}{d_w}$
    Combined it with unit distance and special unit vector, get the shifting vector $v = (v_x, v_y, v_z)$
    Shifting the position to new point $P_w\ (x - v_x, y - v_y, z - v_z)$
    P_w is new P_0
end for
Get the final position *P*

Since all these methods are developed by modularized, so they can combine to the required structure and algorithm, that is TDoA and k-NN ----- KF/EKF ----- RTS Smoother.

7 Result and Calculation

7.1 Calculation

All the method will be coding in MATLAB, that means most of the calculation will be finished by computer. So, in this part, it will show the most important calculate process or result of the above four difference method.

TDoA and k-NN method
When it using TDoA method to estimate the position, it needs to do a coordinate transform here. that it is to change the geocentric coordinate system to a geographic coordinate system. The following is the transform function of geocentric coordinate system to geographic coordinate system. Notice that here the unit length of geocentric coordinate system is 1 km and the geocentric coordinate is (x, y, z); the geographic coordinate is (α, β, h), α is the longitude, β is the latitude and h is the altitude of the position, N is the radius of earth.

The geocentric coordinate system and geographic coordinate system have the following relations, it is similar to change a Cartesian coordinate to a Spherical coordinate. The only difference of that this is a Spherical coordinate start from a surface not from the central of the sphere. So that means here it will use only part of the radius that it is the altitude (the height between the point and surface) [51] [56].

$$\begin{cases} x = (N + h)\cos(\beta)\cos(\alpha) \\ y = (N + h)\cos(\beta)\sin(\alpha) \\ z = (N + h)\sin(\beta) \end{cases} \tag{7.1.1}$$

By rearrange and solved this equation, the following is the function that change the geocentric coordinate to geographic coordinate.

$$\begin{cases} \alpha = \arctan\left(\frac{y}{x}\right) \\ \beta = \arctan\left(\frac{z}{\sqrt{x^2+y^2}}\right) \\ h = \sqrt{x^2 + y^2 + z^2} - N \end{cases} \tag{7.1.2}$$

Here since the α and β are the longitude and latitude, so there has a range on that, that is $0° < \alpha < 180°$, $0° < \beta < 90$.

Notice that the coordinate transfer will used after k-NN runs, since for a k-NN method that based on the distance, it is more convenience

For k-NN method, as the above show that the weight of each difference input is determined by their distance between the last modified position point (or the initial position point). By using the distance, it can get a weight and then it will be multiplied by a unit distance which is determined by difference errors and the aircrafts' velocity. Notice that here the distance will have direction, that it the vector. So actually, here it will use a unit vector which It is the unit vector from the position point to the point that determined by TDoA method. Then, the initial value will shift the distance of this special vector in the direction of this vector. It can represent by the flowing picture:

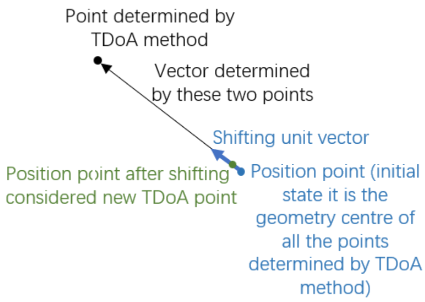


Figure 7.1 The explain of k-NN method

First, by using the position point (p_x, p_y, p_z) and the point that determined by single TDoA method (x, y, z), it can get a vector that from the position point to the point determined by TDoA:

$$\vec{v} = (x - p_x, y - p_y, z - p_z) \tag{7.1.3}$$

In the same time, since the distance will use on the weight, that the weight is the reciprocal of distance between these two points, so it can get that here the weight w_k is:

$$w_k = \frac{1}{d_{position\ point\ to\ TDoA\ point}} = \frac{1}{\sqrt{(x-p_x)^2 + \left(y-p_y\right)^2 + (z-p_z)^2}} \tag{7.1.4}$$

Then, since the distance and vector coordinate between these two points have been know, the unit vector can get here, that it is:

$$\overrightarrow{u_v} = \frac{\vec{v}}{|\vec{v}|} = \frac{(x-p_x, y-p_y, z-p_z)}{\sqrt{(x-p_x)^2+\left(y-p_y\right)^2+(z-p_z)^2}} \tag{7.1.5}$$

By combined the weight and the unit vector, then the final shifting vector can get:

$$\overrightarrow{s_v} = w_k\overrightarrow{u_v} = \frac{(x-p_x, y-p_y, z-p_z)}{(x-p_x)^2+\left(y-p_y\right)^2+(z-p_z)^2} \tag{7.1.6}$$

Finally, the new position point is:

$$\left(x + \frac{x-p_x}{(x-p_x)^2+\left(y-p_y\right)^2+(z-p_z)^2}, y + \frac{y-p_x}{(x-p_x)^2+\left(y-p_y\right)^2+(z-p_z)^2}, z + \frac{z-p_x}{(x-p_x)^2+\left(y-p_y\right)^2+(z-p_z)^2}\right)$$

By repeating this operation, then the final position can get. Theatrically, this is the most accurate position that it can be get by this method.

KF and GPS method

As part 4 shows, besides the measurement of position, there are lots of parameters that needs to be calculate or find in KF, most of that are the variance, the variance of the position or the variance of the noise. So, here in order to get these parameters, it can be started from the matrix one by one.

First it is the time step, in normal KF, the time step is a constant value so that the other parameter of matrix can be calculated by using the time step. But now, it has been changed to a variable value since in the aircraft tracking, the position data will not broadcast in a fixed time interval due to the different response delay of difference server and aircraft. So, here the time step will determine the interval current server time stamp and last server time stamp. That it can represent as the following in *k* step, the time step that can used here is: $t_{sk} =$

$t_k - t_{k-1}$. Here the t_k and t_{k-1} are the difference state server time of the airplane. It has been given by the database.

Next, it is the parameters in the covariance matrix, for a normal distribution $N \sim (m_0, P_0)$, here the P_0 is the variance of the position. For a typical commercial aircraft, the airway of that is 4 nautical miles (7.4 km) two sides respectively [57] [58]. Here since it uses the geographic coordinate, for two integer latitude, the distance between them are 111 km. So, for the covariance in longitude and latitude, it can get that is:

$$p_x = p_y = \left(\frac{7.4}{111}\right)^2 = 8.5 \times 10^{-3} \approx 0.01 \tag{7.1.7}$$

For the altitude of an airway, it is 300 meters up and down side respectively. But for an commercial aircraft, the vertical stability will keep in 10 meters when it maintain level flight [57] [59]. So here by consider this, it can calculate the covariance of altitude, that is:

$$p_z = (10)^2 = 100 \tag{7.1.8}$$

Then, the last is the covariance of velocity, for a commercial aircraft, the typical cruise speed is 800-900 km/h, for a Boeing 737-800 aircraft, it is 828 km/h and for Airbus A350-900, it is 903 km/h [60] [61] [62]. For difference flight, the cruise speed is difference. Here, 850 km/h has been chosen as the mean to calculate the covariance. In order to find the covariance, here it will use the difference between maximum speed and cruise speed to calculate that. For a commercial aircraft, the maximum speed is usually 0.85 Mach, that it is 1050 km/h. So, by transfer that into longitude and latitude (degree per second), it can get the covariance of velocity on horizontal is:

$$p_{vx} = p_{vy} = \left(\frac{1050-850}{111\times3600}\right)^2 \approx 10^{-6} \tag{7.1.9}$$

Difference to the horizontal direction, in the vertical direction, the speed of aircraft is much smaller since the rate of climb of a commercial should not be very high usually. For a common case, the maximum climb rate of a commercial aircraft is 3-10 m/s [63]. Here, it will use 5 m/s as the border of that, in this situation, the covariance in vertical direction is:

$$p_z = (5)^2 = 25 \tag{7.1.10}$$

Matrix $\boldsymbol{Q}$ is the covariance of processing noise that included in the state of aircraft that needs to estimate, so the parameter of that will keep same as that in matrix $\boldsymbol{P}$, that it is $q_1 = q_2 = 0.01$, $q_3 = 100$ here. The final one $\boldsymbol{R}$ is the measurement noise. That it is related to the measurement sensors. In the explain of Opensky database, it shows that the resolution rate of these sensor is usually on 10MHz order of magnitude. But after observe the data base. The time have big delay that all the delay time are in 100 microsecond order of magnitude. So, here the covariance of horizontal direction will be based on this, combined with the speed of light, it can get like:

$$\sigma_x^2 = \sigma_y^2 = \left(\frac{300000\times100\times10^{-6}}{111}\right)^2 \approx 0.07 \tag{7.1.11}$$

For $\boldsymbol{R}$ in vertical direction, since the GPS location accuracy is 10m. Then, the covariance can write as:

$$\sigma_z^2 = (10)^2 = 100 \tag{7.1.12}$$

As it shows on the above, some time there will have a mistake value in the database, so here, the only method is to use a pre filter to remove that from the data base. In order to remove that, it need to set a function for the three components of coordinate to the time separately. Since in this method, all the data are discrete since the measurement can only able to measure the information in a time point. In order to find the overall concentration of all these points, and find a describable function use to fliting those mistake point. Here it will use the polynomial regression method. It is a kinds of regression method to find a regression function to describe a group of point that it didn't have a linear relation approximately. the following is the model of that:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_n x^n + \epsilon \tag{7.1.13}$$

In this method, since MATLAB have the build-in tools to find the highest order and the coefficients of all the terms, so it will use the tool to find a suitable regression function. And then use it to fitting the mistake value.

EKF and RSSI method

For EKF, since much of the matrix are same as the KF, so all the parameters are same as the KF. it will no longer to show the calculation here since they are all same. The only difference one is the $\boldsymbol{H}$ here it is using the Jacobian matrix of power loss exponential function. The function has been given by equation (4.2.10). so, here the $\boldsymbol{H}$ can be getting as the following:

$$\boldsymbol{H} = \begin{bmatrix} \frac{-10\gamma\left(x_{kx}-m_x^j\right)}{\ln(10)[\left(x_{kx}-m_x^j\right)^2+\left(x_{ky}-m_y^j\right)^2+\left(x_{kz}-m_z^j\right)^2]} \\ \frac{-10\gamma\left(x_{ky}-m_y^j\right)}{\ln(10)[\left(x_{kx}-m_x^j\right)^2+\left(x_{ky}-m_y^j\right)^2+\left(x_{kz}-m_z^j\right)^2]} \\ \frac{-10\gamma\left(x_{kz}-m_z^j\right)}{\ln(10)[\left(x_{kx}-m_x^j\right)^2+\left(x_{ky}-m_y^j\right)^2+\left(x_{kz}-m_z^j\right)^2]} \end{bmatrix}^{\boldsymbol{T}} \tag{7.1.14}$$

For the power loss exponential function h, it also has two difference parameters that need to determine here, the first is the path loss exponent γ, since in SNR calculating, it has a coefficient of 2, so here it chosen $\gamma = 2$ as the path loss exponent here. and another is the received power at a reference distance P_0, the accurate value cannot be finding in any database or instrument, but for most ADS-B equipment, the P_0 of that usually equal to 70 dB [64]. So, here it will use this value to calculate.

RTS Smoother method

Since the RTS Smoother is developed from KF. So, all the parameters in RTS Smoother has been solved out on KF part. It will no longer the show that here. for the post-filter, it has the same principle and method like the pre-filter in KF, so it will not show that too. For RTS Smoother, there is no other calculation since all the method here are developed for KF.

7.2 Result

In the Opensky database, it has two difference kinds of aircraft in that, the first one is that it has an initial GPS data directly. For these aircraft, it can use the KF and EKF directly without using other method to locate the position first. This is easier so it can directly use to implement a KF and EKF. So, in this result part, it will not use the order of the total tracking algorithm, that is first TDoA & k-NN, then KF or EKF, and finally RTS Smoother. Since in the real construct of that, the TDoA & k-NN method are the hardest one, so it has been analysis and implement at the last. In real developed, the KF was first finished since it is used in a linear system and will just use the location data itself. Next it is the EKF data since the initial location can be confirmed by the GPS data directly. In this part, it will show the result in this order.

For RTS Smoother, it can be used in both KF and EKF method, in this method, all the coding work are based on different module. For the different model, they can combination with the right part in right order randomly. In order to reduce the length of this article, here it will be adding the RTS Smoother on the KF only to show the result of that.

KF and RTS Smoother method

Here it has been applied in three different aircrafts. Each aircraft represents one kind of aircraft in the database. The first one is flight 1602, it is a flight fly across the border of Italy and Austria:

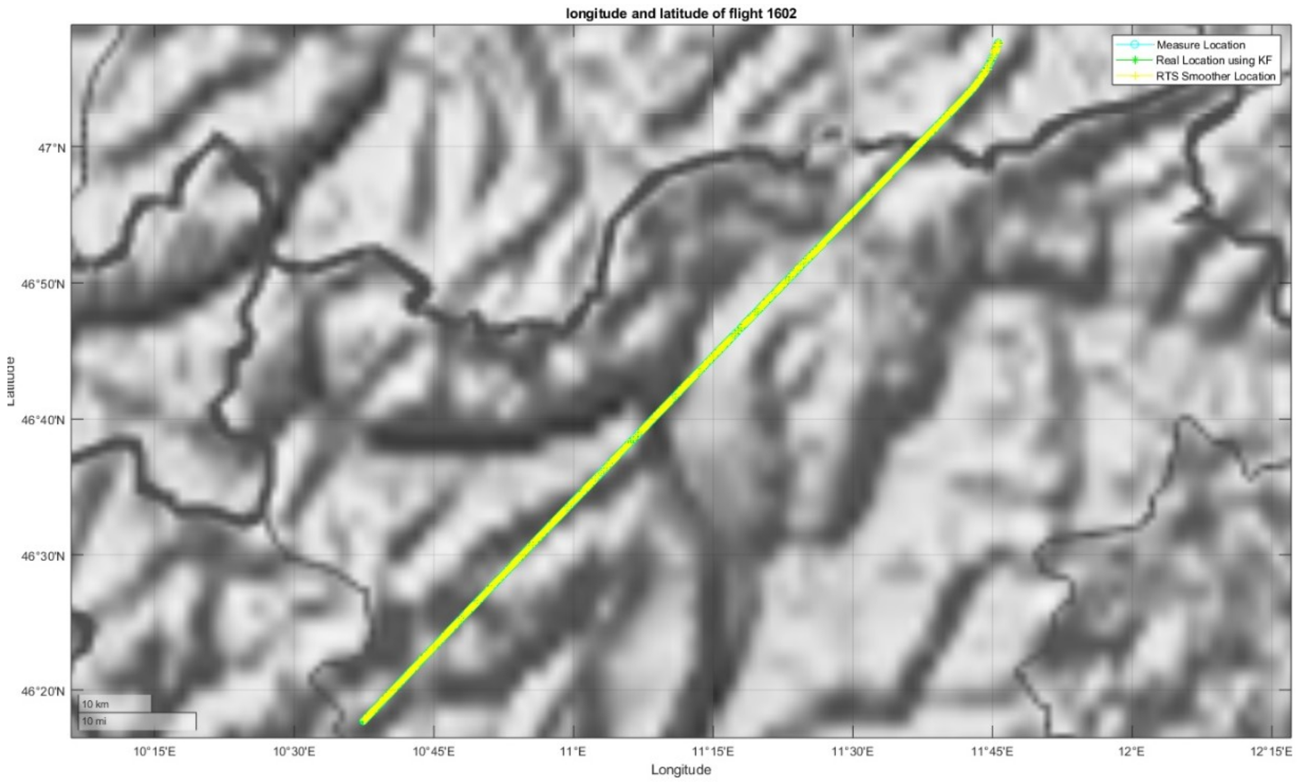


Figure 7.1 the longitude and latitude plot of 1602

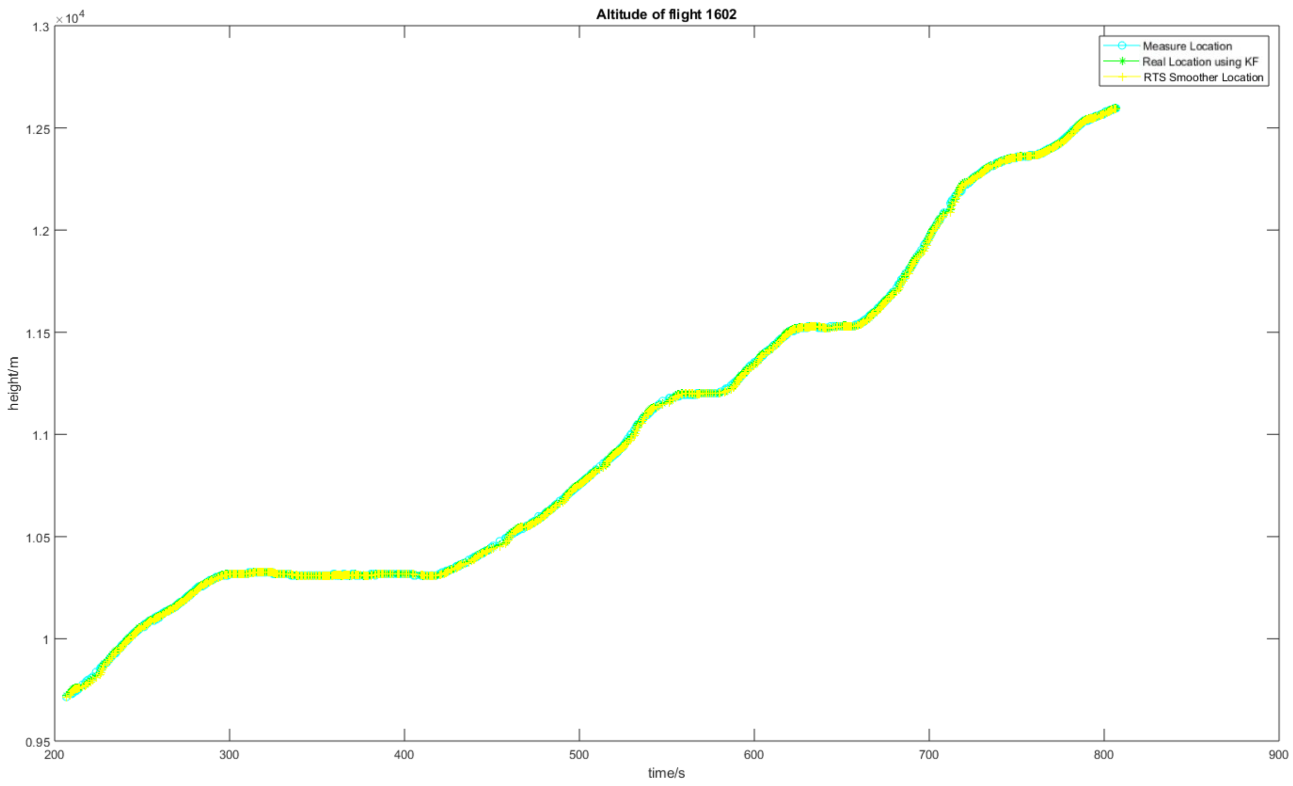


Figure 7.2 the altitude plot of 1602

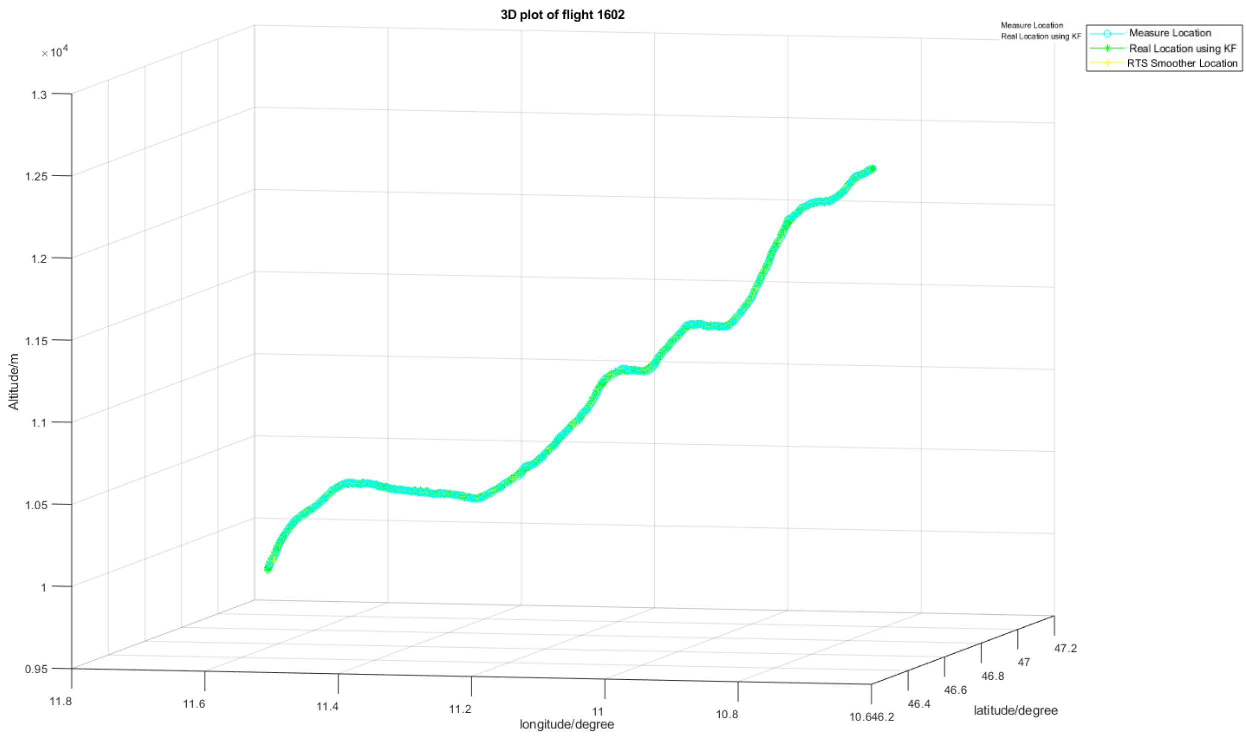


Figure 7.3 the 3D plot of 1602

From these figures, it shows that for flight 1602 (an aircraft type that it didn't have big no-signal zone) the estimation of KF and EKF will followed the measurement data urgent. That means the KF and RTS has a good estimation of that.

Next one is flight 1787, it is a flight fly near Frankfurt, Germany:

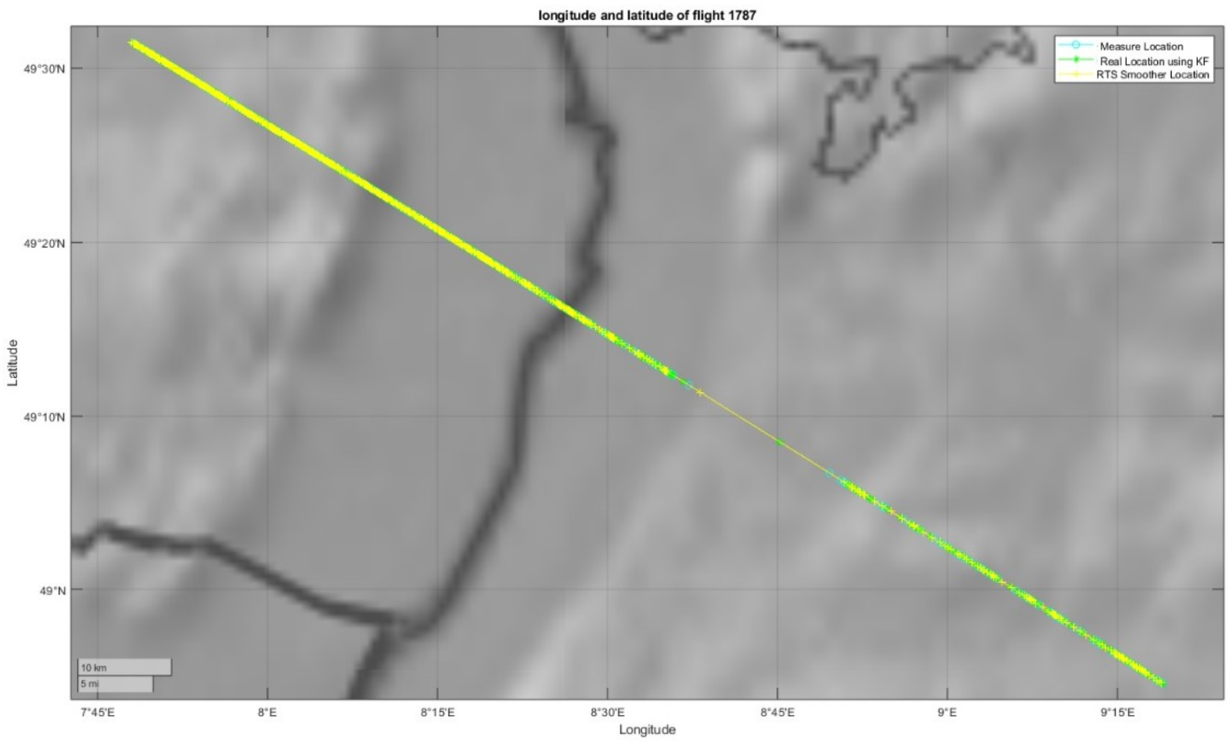


Figure 7.4 the longitude and latitude plot of 1787

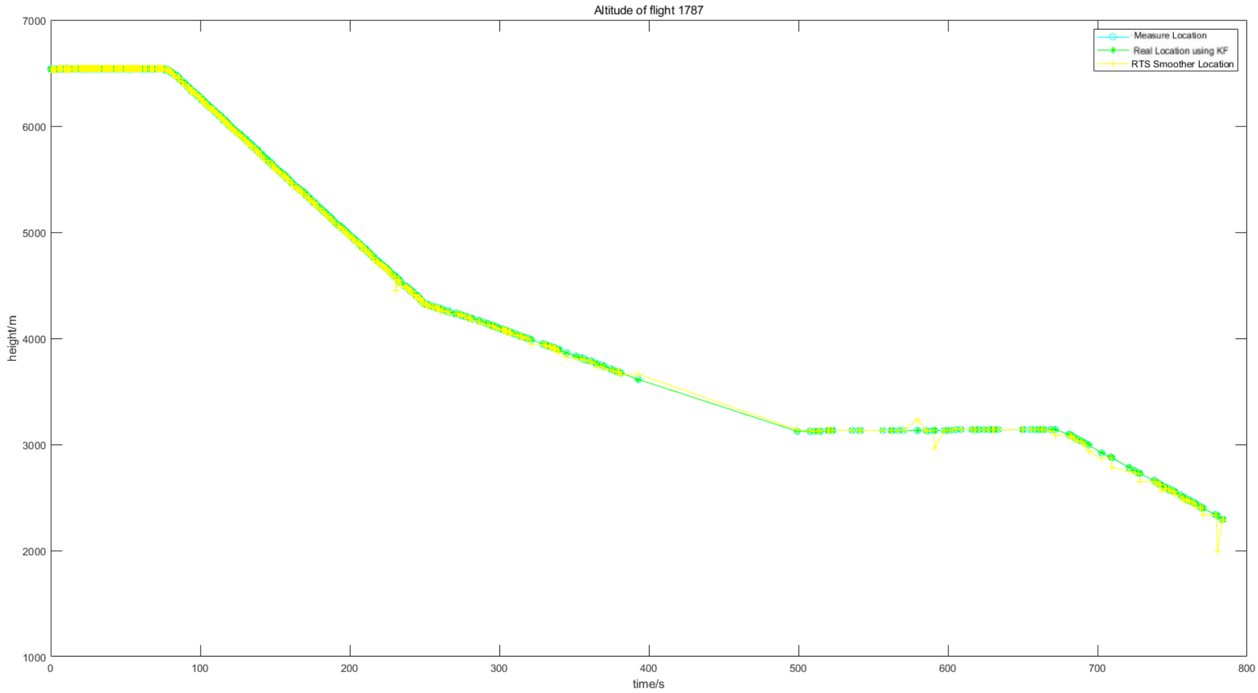


Figure 7.5 the Altitude plot of 1787

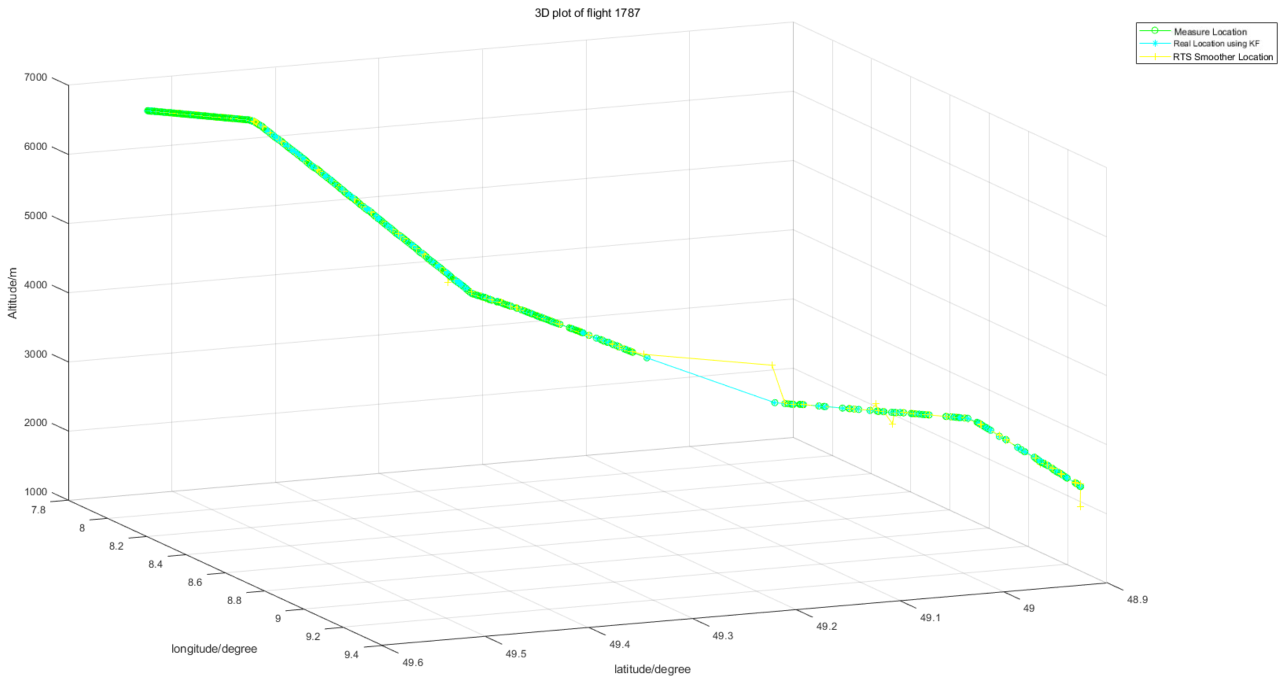


Figure 7.6 the 3D plot of 1787

Here, for this airplane (an aircraft type that it has a big no-signal zone), the KF and RTS follows the measurement good too. But for the RTS Smoother, it has some small shake, but these are all under the allowable range of error. The reason why this happened is may because of the parameter setting and velocity estimating. For further error analysis, it will show on the part 8.

Finally, it is flight 2022, it is a flight fly from Corsica, France to Milan, Italy:

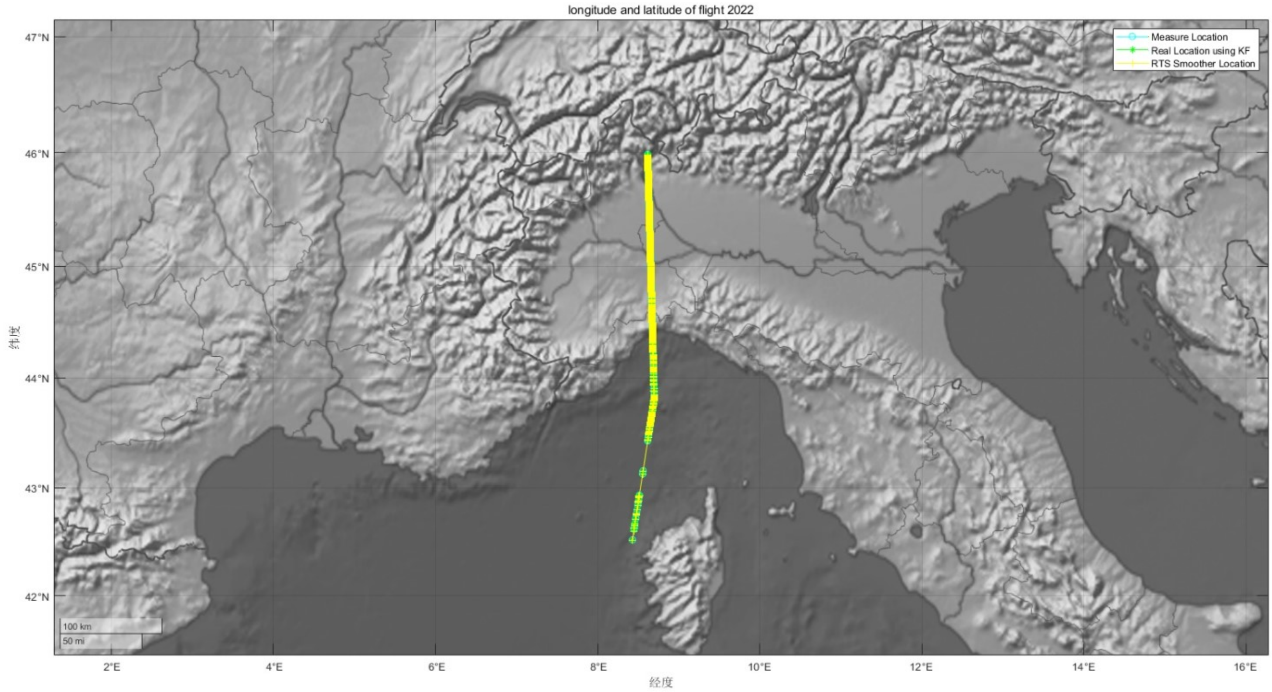


Figure 7.7 the longitude and latitude plot of 2022 (without post-filter)

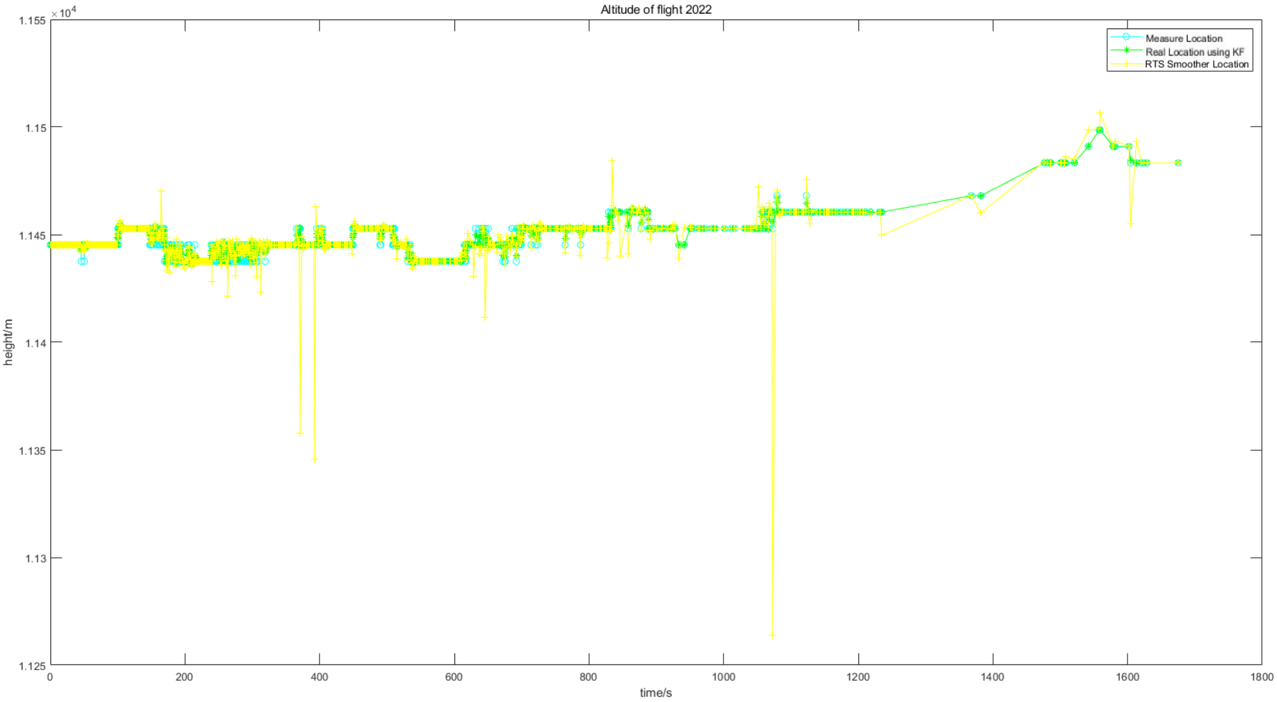


Figure 7.8 the Altitude plot of 2022 (without post-filter)

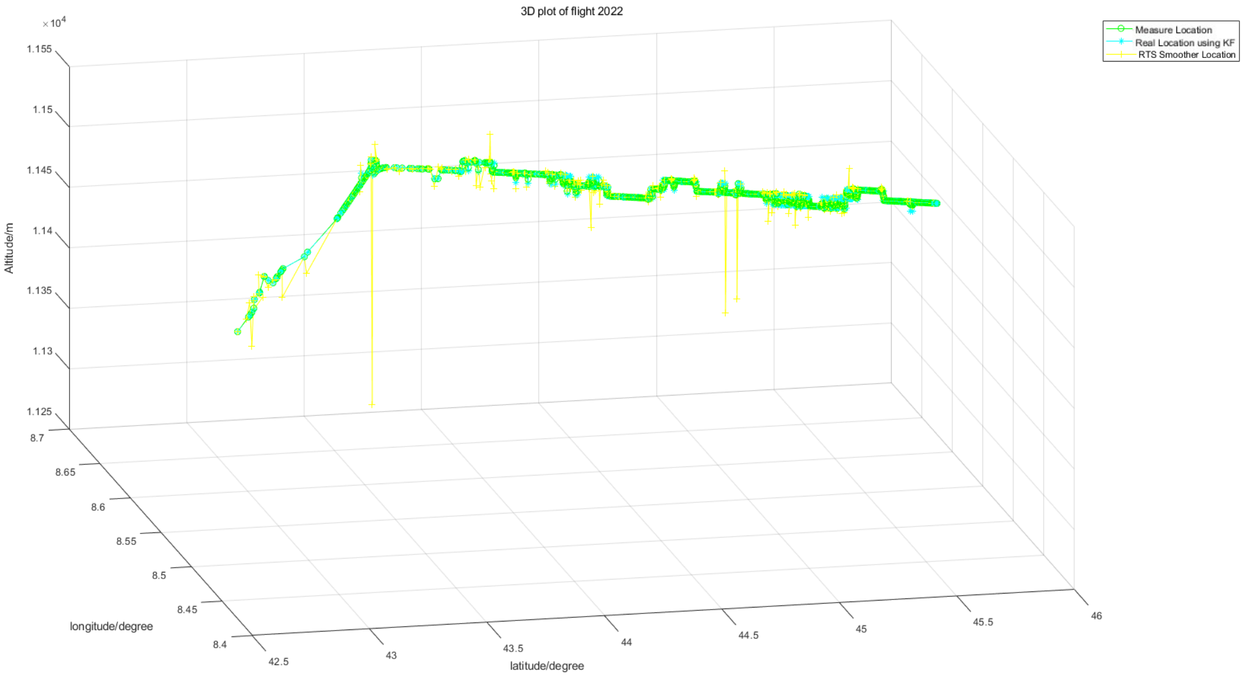


Figure 7.9 the 3D plot of 2022 (without post-filter)

Here as it can see on the above, the KF follows the measurement very well. In Longitude and latitude, the RTS Smoother follows good too. But in altitude, the RTS Smoother has a big error. The reason might be the setting of parameters, difference to the above two flight, this flight is maintaining level flight state and the GPS data have some beating. Additionally, the velocity estimation is not very good in this method. So, under this circumstance, the RTS Smoother cannot get the right value here. Since it is harder to change the estimation method of velocity and change the parameters, so here a post-filter is added to get a better result, the following is the plot of 2022 after go through the post filter:

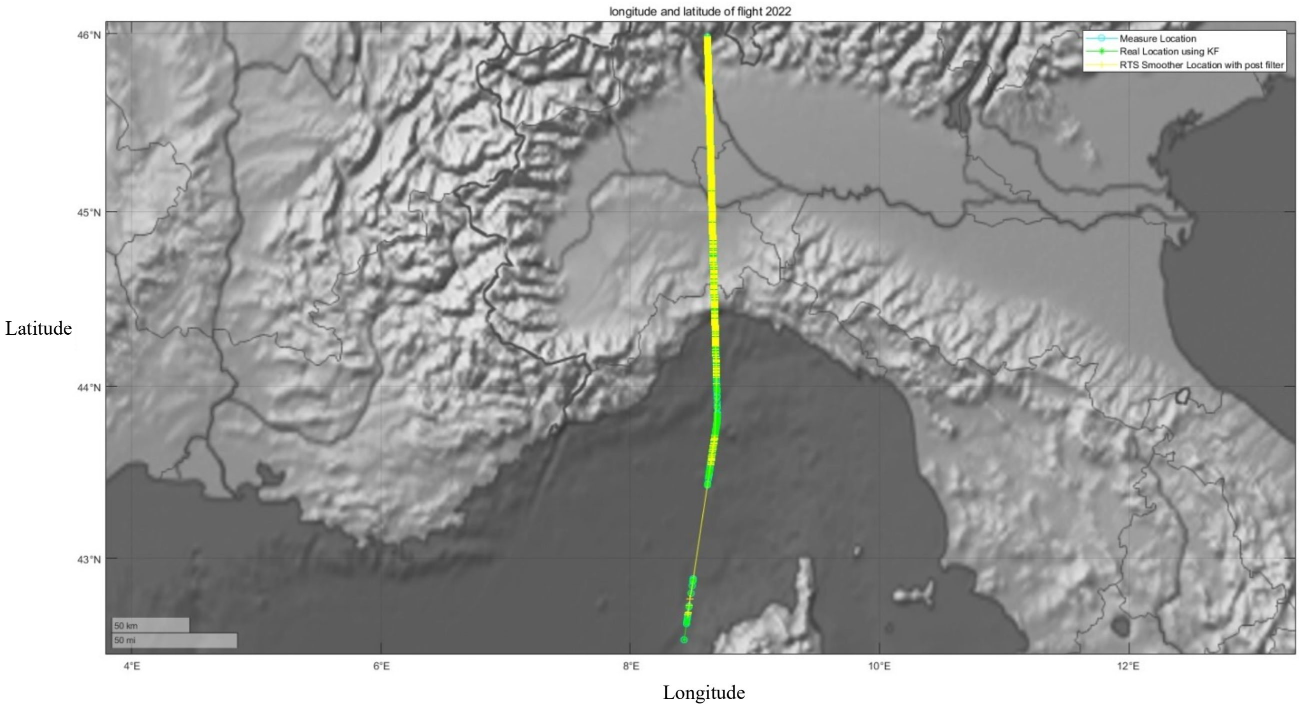


Figure 7.10 the longitude and latitude plot of 2022 (with post-filter)

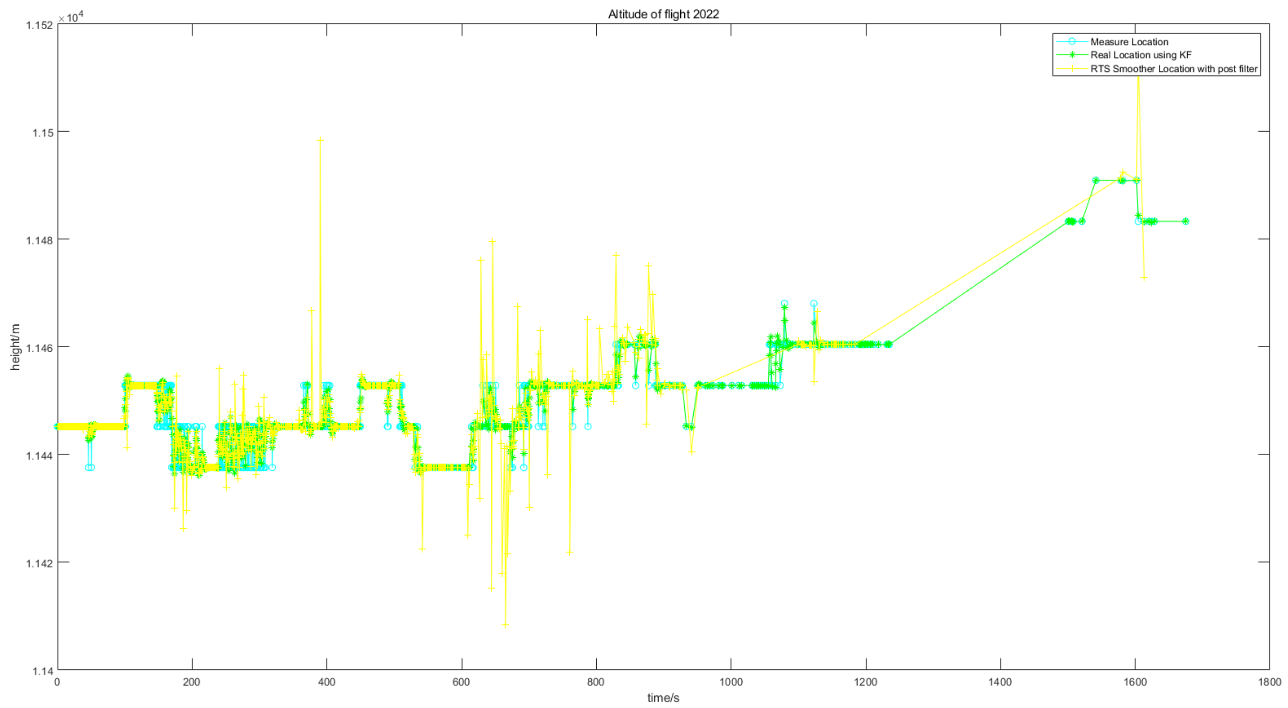


Figure 7.11 the Altitude plot of 2022 (with post-filter)

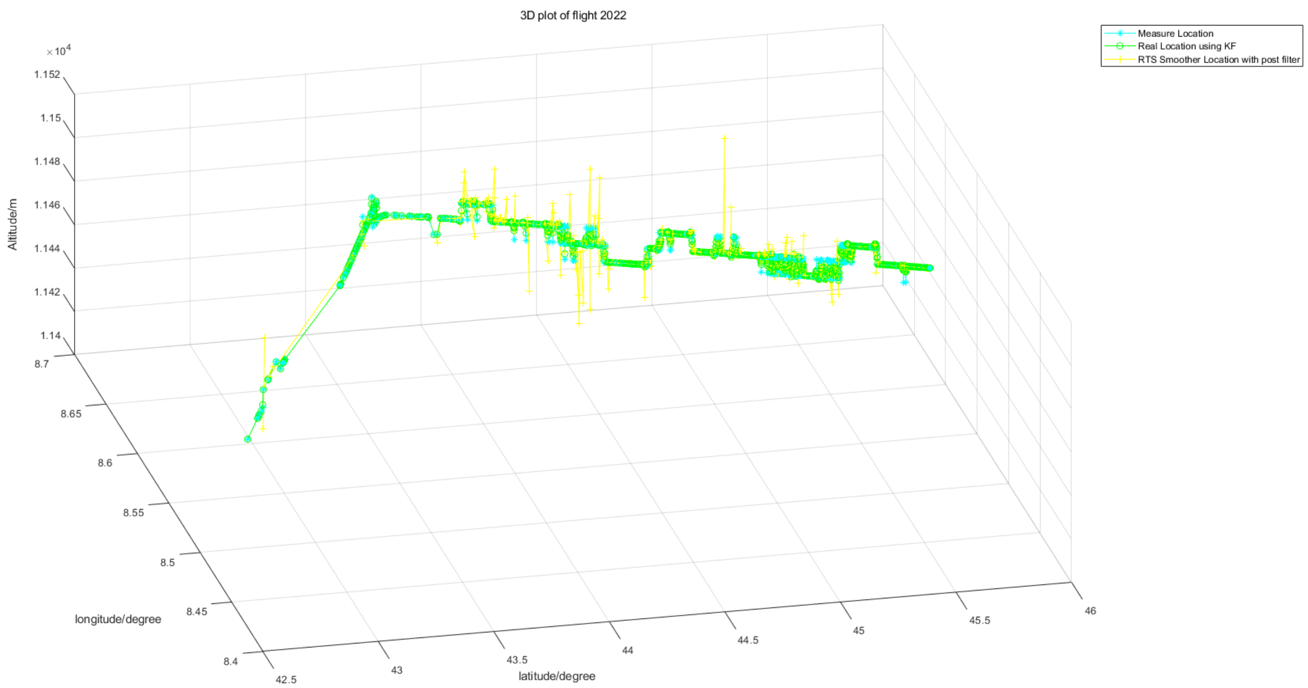


Figure 7.12 the 3D plot of 2022 (with post-filter)

Here the RTS Smoother estimation still have some shake, but it has been much better since here all the shake is under the allowable range of error.

EKF and RSSI method

For EKF and RSSI method, due to the quantitative limitation (for one position, there are only 2 or 3 sensors' data to estimate). So, it will remove the velocity components here to estimate. Since the measurement of this method is RSSI. It will not use measurement position to estimate. So, in this EKF, it will only include the position of aircraft in final result. Additionally, it will not have any type distinction since the input of that are RSSI data. So, here it just shows flight 1602's result:

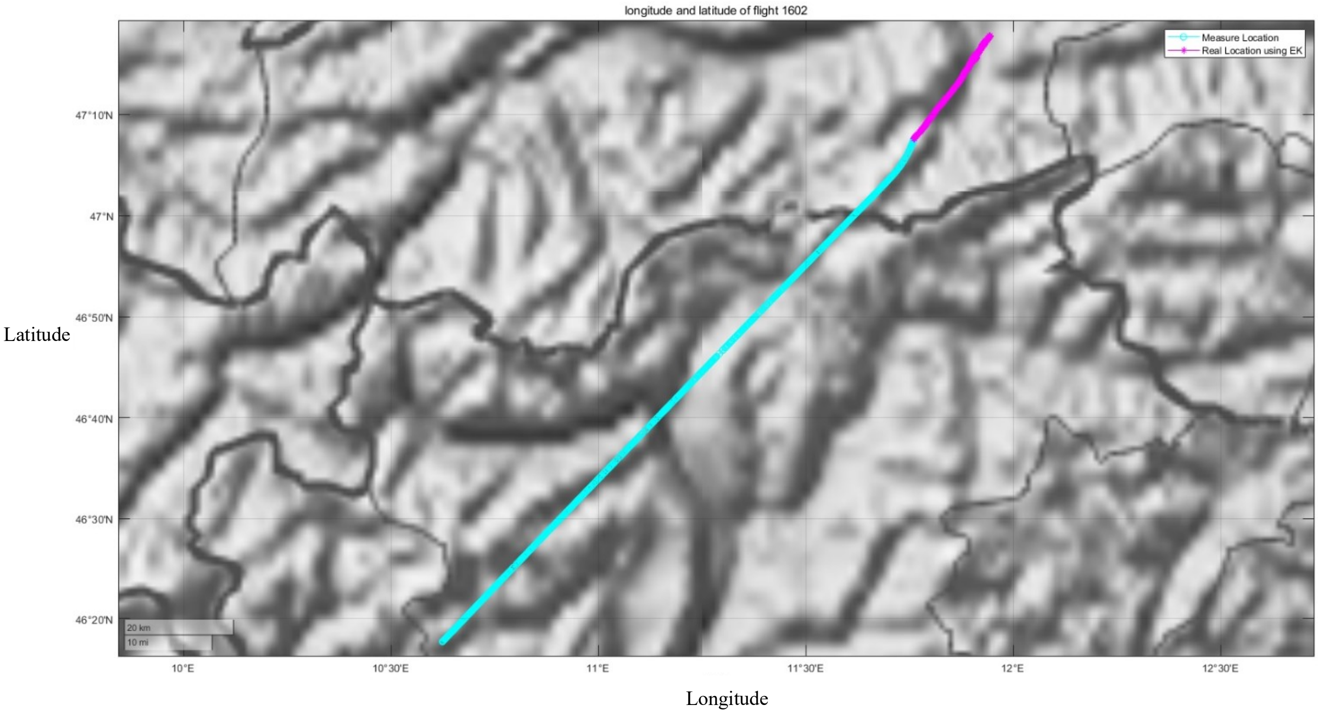


Figure 7.13 the longitude and latitude plot of 1602 using EKF and RSSI estimation

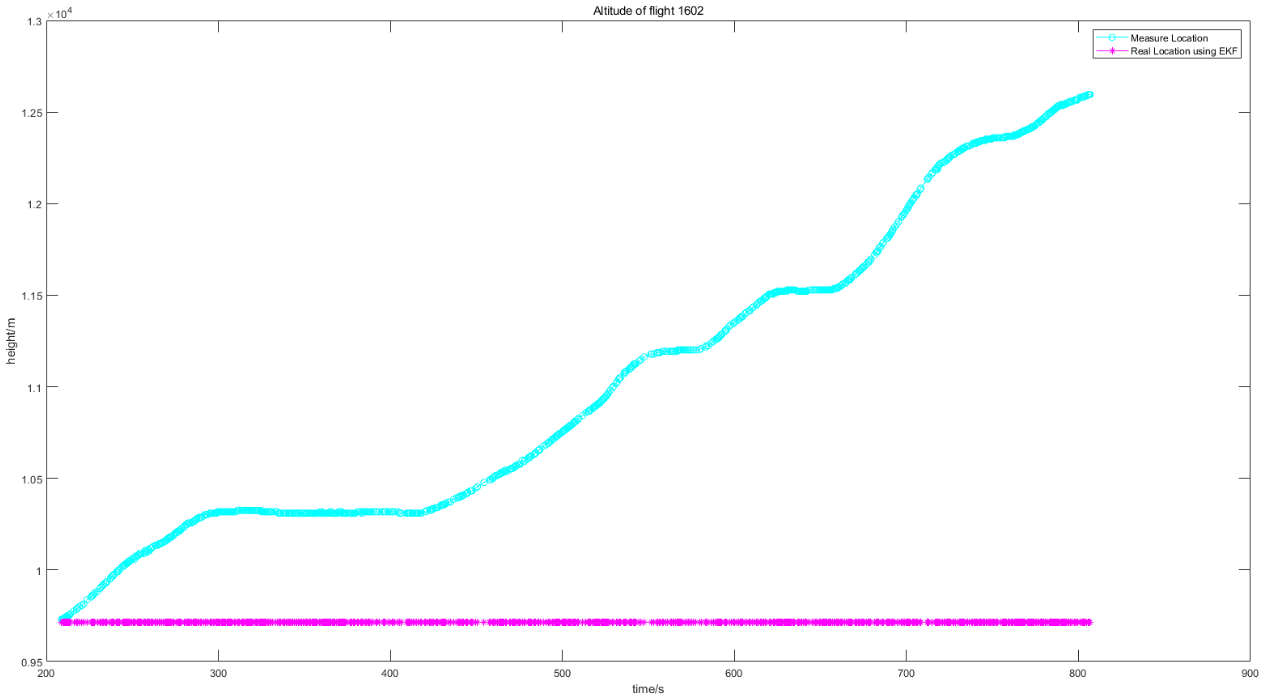


Figure 7.14 the Altitude plot of 1602 using EKF and RSSI estimation

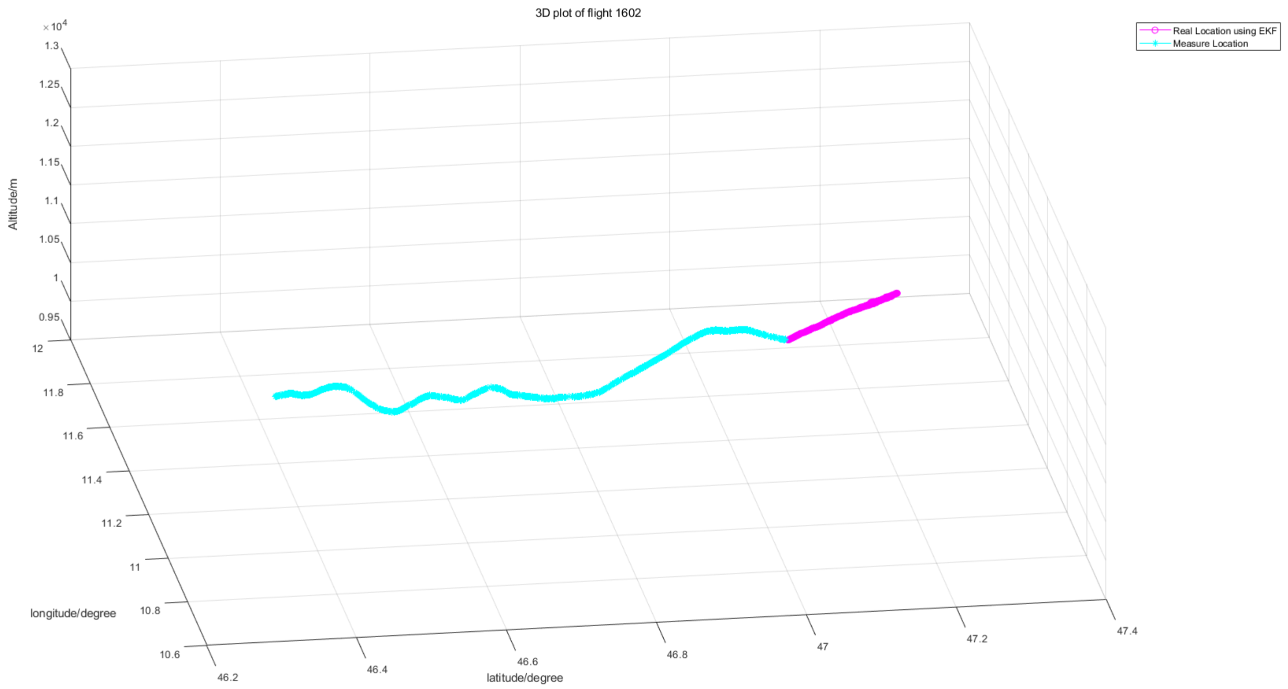


Figure 7.15 the 3D plot of 1602 using EKF and RSSI estimation

As it can be seen in the above, the estimation has a different direction with the GPS data. Here the reason why it become here is that here the P_0 is wrong here. Since it didn't have any reference to calculate or find the correct P_0 for all these ADS-B sensors, here the P_0 is used a common value, that it is 70 dB here. But due to the figures here, it is totally wrong. If a correct is get, here the estimation must be much better. Another reason is that it didn't have enough sensors to estimate, as it known, all the estimation of position in EKF are highly depend on the former one. that means if the one position is incorrect, then after that, all the position will wrong. So here as it can see, the altitude estimation maintained in a level. That because many positions only have two sensors data, which is not enough to get the correct location. So, in order to solve this problem, it is to use the baro-altitude that it is measured by barometer. But it may influence by the surrounding environment strongly. So, for real world using, it needs to use some adaption method or artificial intelligence method to adjust that. But in the database, it didn't give any weather or environment data, so it cannot be optimized by using this. For further analysis, it will show on the part 8.

TDoA and k-NN method

Due to the limitation of computer performance (the computer did not have an ability to training a k-NN network since the loop of that are too much), software limitation (MATLAB cannot solve a complex equation like that in part 4 and part 5) and data quantities (for some position that need to confirm, there only have 2 or 3 sensors, that it is even not enough for TDoA method). So, it cannot be implemented by coding on MATLAB. Here, it will no longer to change that into a simple program here. Here it will just have some theoretical research and approach for that. the result of theoretical analysis has been shown on the last few parts. So, it will no longer to tautology here. For further evaluation, it will show on the discussion part in the following.

8 Discussion

8.1 Error Analysis

From part 4, it can know that for this method, the error of measurement has been considered in that. Since for KF and EKF, there are two difference kind of noise included in KF and EKF model, the process noise q_k that it is related to the state that needs to estimate. Another is the measurement noise that is r_k that it is related to the measurement data, that is the input of a KF or EKF. So, for a KF or EKF, theoretically, it can reduce the error not only for the measurement but also for the estimation. Besides, since the all the noise is a normal distribution. That means of the whole process of the calculation, the KF or EKF will chose a suitable by considered as the normal distribution. Which is that here the error can adjust by the filter itself according to the position. So, by choosing the suitable parameters, the KF and EKF will have a highly accuracy. As it shown as the flight 1602 and 1787 in result. Then, since the RTS Smoother is a kind of backward recursion that based on the KF or EKF estimation, it will approach the real value further than the KF and EKF. Theoretically, the RTS Smoother will have a higher accuracy than KF and EKF. From the result part, it can see that for 1602 and 1787, it is true that the RTS is more accurate.

But actually, the results are not always like that expected, in the RTS Smoother estimation of flight 2022 and the EKF estimation of flight 1602, it can be seen that the estimation is quite inaccurate. For EKF, the reason of that may have two, the first is that it didn't find a correct power reference that it is the P_0 in the power loss exponential model. And the power loss exponential will be related directly to some parameter matrix of KF. another one is the sensors number. Here the location principle of RSSI method is similar to the TDoA method. The core of that is the multiliterate. For RSSI method, it needs at least 3 difference sensors to estimate the location since for each position coordinate, it has 3 different components. But in the database, as it shows on the following. May of it only have 2 sensors for a location. Additionally, though the RSSI data will used on EKF, and the current state will be based on the former state. Since it cannot get the correct P_0 of the power loss model. So, only by using two sensors, it cannot get the correct position data. Once one position did not have a correct estimation, then the rest will all incorrect since the EKF estimation are strongly determined on the previous state.

1	206.709	1602	47.12631	11.76011	9715.	2	[125,207671993609,104],[131,207671936687,78]]
2	208.838	1602	47.12242	11.75819	9730.7	2	[131,209811947000,69],[125,209812003718,83]]
3	209.441	1602	47.12137	11.75764	9730.7	2	[130,210402263421,70],[125,210401994312,84]]
4	209.903	1602	47.12064	11.75729	9730.7	2	[125,210882065984,104],[131,210882009359,61]]
5	210.437	1602	47.11958	11.75674	9738.3	2	[131,211411943718,65],[130,211412268515,46]]
6	210.98	1602	47.11873	11.75622	9738.3	2	[131,211951926984,72],[130,211952251296,40]]
7	211.481	1602	47.11784	11.75578	9745.9	2	[130,212462274515,91],[131,212461950718,80]]
8	211.999	1602	47.11696	11.7553	9745.9	2	[130,212982335875,44],[131,212982012531,61]]
9	212.449	1602	47.1161	11.75475	9745.9	3	[130,213432237421,68],[125,213431970984,56],[131,213431914562,79]]
10	212.956	1602	47.11524	11.75424	9753.	2	[125,213931989656,96],[131,213931933218,64]]
11	213.498	1602	47.11436	11.75379	9753.	3	[130,214482347250,53],[125,214482081718,59],[131,214482025406,86]]
12	214.077	1602	47.11336	11.75311	9761.2	2	[131,215041929703,46],[125,215041986000,68]]
13	215.999	1602	47.11007	11.75108	9776.4	2	[130,216982335296,35],[131,216982015828,95]]
14	217.589	1602	47.10737	11.74924	9784.0	3	[125,218561993843,68],[130,218562255953,65],[131,218561937906,106]]
15	217.995	1602	47.10667	11.74875	9791.	2	[131,218961920656,100],[125,218961976562,36]]
16	219.096	1602	47.10484	11.74747	9799.3	2	[125,220071981390,79],[131,220071925593,95]]
17	220.687	1602	47.10214	11.74548	9806.9	2	[130,221672242812,83],[131,221671927578,93]]
18	221.79	1602	47.1003	11.7441	9814.5	2	[130,222762224656,78],[131,222761910453,79]]
19	223.698	1602	47.0972	11.74157	9837.4	2	[130,224662219921,80],[131,224661907343,85]]
20	225.955	1602	47.09345	11.7384	9852.6	3	[125,226921984375,92],[131,226921929437,81],[10,227922300890,47]]
21	226.437	1602	47.09267	11.73772	9860.2	2	[130,227412206187,71],[131,227411896031,71]]
22	227.016	1602	47.09171	11.73683	9860.2	2	[125,227991976125,112],[131,227991921359,79]]
23	227.554	1602	47.09089	11.73615	9867.	2	[130,228542207484,50],[131,228541898156,82]]
24	228.134	1602	47.08992	11.73523	9875.5	2	[10,229102271859,56],[125,229101957062,115]]
25	229.709	1602	47.08759	11.73299	9883.1	3	[10,230682346015,38],[320,230682341203,41],[131,230681978031,77]]
26	231.143	1602	47.08521	11.73065	9898.3	3	[130,232112192203,50],[125,232111939921,84],[131,232111885703,95]]
27	231.704	1602	47.08443	11.7299	990	3	[130,232662202078,70],[131,232661896125,52],[125,232661950203,94]]
28	232.708	1602	47.08292	11.72832	9913.6	2	[130,233672193281,73],[131,233671888078,50]]
29	233.273	1602	47.08195	11.72736	9921.2	2	[10,234252273859,35],[125,234251963234,88]]
30	234.299	1602	47.08044	11.72571	9928.8	2	[10,235282346500,46],[131,235281983031,90]]
31	234.715	1602	47.07967	11.72488	9928.8	2	[125,235691971875,103],[131,235691918234,109]]
32	235.215	1602	47.07893	11.72406	9936.4	3	[130,236192212546,55],[125,236191962828,91],[131,236191909265,94]]

Figure 8.1 Parts of the database used in this project
Here, as it shows in the red frame, this is the number of sensors can be used to estimate the position. As it can see, many of that are 2 sensors

A possible solution is that it can use the barometry altitude (the above database is a database after some handling, it just keep the data that needs to use in this method, in original database, it has the barometry data that from airborne barometer) as the altitude of the aircraft though it has some error to the real altitude (usually 200-300 meters). Here, by using this, it just needs to solve only other 2 different unknown value, that it is the longitude and latitude. If it has been used, the altitude plot will not a level line and the longitude and latitude will not have a big error like the above figure shows in part 7.

Besides these big errors show on above, there still have many small errors that may influence the estimation here. it will show by start with difference method on the following:

TDoA and k-NN method

For TDoA and k-NN method, since it didn't base on normal distribution, previous state and it didn't include an error. So different to the Kalman Filter, here it needs to consider the measurement data like the time delay, the server response delay and the process delay, that it is the error on the algorithm itself. The following, it will classify the difference errors and have a simple analysis:

For TDoA part:
Time errors ---- the time error here is included the time delay and server response. The time delay is caused by the transmission error and the receive equipment error. For transmission error, since all the signal that used to ensure the aircraft's position, it will use the electromagnetic wave to transmit. In normal condition, the time error of it will only on nanosecond. That the distance error of it will just 0.3-3 meters. For an aircraft tracking method, it can be ignored since it is too small for the whole airway limitation of commercial aircraft [58].

Another it is the receiver delay and the GPS/ground station server delay. All these two delay times can consider as one type, that it is the delay error. For TDoA method, since it is used the time difference to calculate the distance between the potential position point and sensor location. So, here it has been considered the delay time here. But by using this method, it requires a high synchronization of these two sensors that used here, that it is to say, it needs to have a similar delay time. For an aircraft, there is only one receiver, so the receiver delay time will maintain same each time receiving outside signals. But for different sensors, the type of them may be difference, that means they may not have a fixed time delay. So, by using this method, it may reduce some time delay like the receiver delay and part of sensors server time delay. But the delay still exists. That it is why it needs to have some redundancy and use a kind of algorithm to fuse their result, find the optimal solution.

Environmental influence ---- For this method, the influence of environment should be also consider here too, one important is that the discern error, this is a small probability event but it may cause since though the resolution rate of that are in nanosecond, but in the busy time of a server or a receiver, it may confused it with other signal and response or label the original mistakenly. So, this is a k-NN used here, the aim of it is that by combined more and more different sensors data to get a more correct one. The other natural conditions will also influence it since some weather like the thunderstorm or thick clouds.

For k-NN part:
For a k-NN network, since it is simply using the distance between two points as the weight of the neural network, and some of the point position may have large mistake once it has been calculated. So, for the k-NN network, the biggest errors are from the sensors.

Sensor numbers --- For the k-NN parts, the accuracy will highly depend on the numbers of sensors used to located the airplane. For more sensors, the results will more accuracy since it references more sensors data. For a typical GPS location data, there are always 20 or more sensors for space-based location, that can get more than $C_{20}^{4} = 4845$ TDoA determined point to combined and estimated the final location of aircraft. By adding the data from ground station, that it is totally fulfilled the requirement to training an k-NN network. But in the database using in this project. Usually it has 2 or 3 sensors data, the most is 7, it is too less to training a k-NN network.

KF and GPS method
In the database, all the GPS data are from real world measurement, and they are all well-handled. So, theoretically, it will not have any too big errors. For the error that may cause on the location of satellite, it will no longer to mentioned here since it has shown on above partial.

For GPS part:
GPS location ---- As it just shows, the GPS data is well-handled, so the error for the GPS system itself can be ignored here. But since the weather and environment in aircraft position is unknown. So, the error may cause from the weather and environment surrounding the aircraft. As the above show, the bad weather like the thunderstorm or thick clouds may influenced the accuracy of that. The surrounding environment will also influence that, if the aircraft are fly in the valley, there will be less satellite or radar to locate, so the accuracy of

the position will reduce. That it may cause big error. That is way a KF used here to reduce the error and make a better estimation.

For KF part:
Parameters in KF ---- there are many difference parameters in KF, for difference sensor, it may have difference errors that may influence the estimation of KF. So, a possible method is that to consider difference values of parameters in difference location. For the error that caused by KF itself, the too big or too small of parameters will influenced the estimation result. For a KF in discrete data, if the parameters set the too big (usually the covariance of process noise), the result of KF will become too smooth. This is good for most situation, but some time in some point, the error will become very big since the measurement has little error with the real one, but after using KF and influence by the previous state. It has become much big since the previous state have a big error [65] [66]. If the parameters are too small. Then the aliasing of the Kalman Filter will become very obvious, since it is too small so that the point will have larger probability to appear around the point with a very short distance to the mean of that [66]. In the normal KF, the time step is fixed, so the value of time step will also influence the KF estimation. But here, the time step is changed with the time difference of two difference GPS position determination. So, the time step will not influence that. But it will influence other parameters, all the parameters here are setting as same. That it may cause error since difference time step and sensors may change the mean and covariance of error.

EKF and RSSI method
Similar to the KF, the error may from the EKF's parameter itself. But it still has some difference, that some matrix of KF are based on the RSSI data, which means if the RSSI sensor didn't have an enough number, the location may determine form the previous state more. Since it didn't have the velocity components here, so it may not get the right direction, so the estimation may have the different direction with the real location. This is one of the big errors that occurred on last part. Besides this, there are still some other errors may influence the result of estimation.

For RSSI part:
Parameters in RSSI ---- As the above show, in RSSI, the P_0 and γ are two difference parameters here, for difference kinds of sensors, it may have large difference. That means for difference type, it needs to set these parameters difference, this is similar to the parameter setting for KF. Another is that parameters finding, for all the type of sensors in the database, it cannot find the technique data of them on internet, many of them are commercial sensors, it can get from manufacturer. But for some of them, they may mainly for military using for most time. So, the technique data of them will not public. That is means it can only get by estimate. That it must have a big error. Like in this method, the P_0 and γ are all get by estimate, the detail for estimate these values has been shown above parts.

RSSI distance calculation method ---- For the RSSI data, here it uses the distance formula in Cartesian coordinate to calculate the RSSI value. In the surface, for a short distance flight, the latitude and longitude coordinate system can default as the Cartesian coordinate since in such a small area, the surface can be considered as a flat. But actually, the earth is an imperfect sphere. So, here it will have some errors if use the distance formula in Cartesian coordinate to calculate. The longer the flight route have, the higher the error will get. So, in order to solve this problem. It is better to use a distance formula on sphere coordinate system. For real world measurement system, it is better to use the Geodetic coordinate system (usually WGS 84) [67]. Since this is the constructed on the earth. So, the error of that will reduced too much. Furthermore, since the distance are all use KM as the unit. So, here it is difficult to use latitude and longitude coordinate system to estimate the real RSSI value and estimate. So, it needs to transform the coordinate system. For further details, it will show on the following.

Environmental influence ---- Same as that in GPS method of KF, here the RSSI will also influence by the bad weather, environment and topography. So, for the detail of that, it will no longer to mentioned here, since it is same with that in KF. But it have other two errors that may occurred here, since the RSSI data are always based on air control radar, must frequency of that is less than 300MHz that it is in microwave and short wave range [68]. So, the power loss of that may influence by other condition and the long the distance, the higher the errors will have, since the short wave will have a short effective propagation distance (hundreds of kilometres). So, for some aircraft that has a long distance with the sensor, the errors will become bigger. For some special using radar, it will use the groundwave to transmit to get a long transmission distance. So, the atmospheric refraction may influence the RSSI data too [69]. This will not only occur on long wave radar, but also will occurs on short wave radar in sometime.

For EKF part:
Parameters in EKF ---- The parameters in EKF are same as that in KF, the only difference is that the measurement model matrix H. But this has been determined by h that mentioned above, so here it will no longer to analysis the error of these parameters, since the details of that has been shown on above.

Noticed that for EKF, the error will also come from the previous estimation of position, since difference to the KF, it is used the position data to estimate position data only. But for EKF, it will use not only position data, but also the RSSI data to estimate the position data. So once the former one position has large error, then the current one will have too since if the sensor is not enough, it will have limited influence on the estimation. For a total estimation. The current state always determined by the former state and RSSI, that means the second state have a big error, then the rest one will have too.

Besides, the EKF didn't adding the velocity here, this will also cause the error to a certain extent.

RTS Smoother method
RTS Smoother is developed from KF method, so the error caused reason is same as that in KF, that it is the parameters and the KF data, since all of it will based on KF, from the parameters and the position data. So, it will not have the further analysis of error for this method. For the details of errors, it has been mentioned in the error analysis of KF on above of this part.

Other further Optimization or simulation method
The velocity simulation method ---- Here, in order to find the velocity of the aircraft in a specified position, a kinds of velocity simulation method that use the derivative of the polynomial regress function of the measurement data has been used here. As the picture in last parts shows, most situation, the regress function will be matching the measurement of position very well. But in some time, it has a big difference. In this point, the velocity will have some error and this will influence the estimation of KF, EKF (if adding the velocity components) and RTS. One way to solve this is to find a new regress method that it will fit the location points better. In real world using, the velocity will from the speedometer on the aircraft. So, the error will from the speedometer itself. And here the speed that use to calculated must be the ground-speed since the if it using the air-speed, it needs to remove the air speed here.

The pre filter and post filter ---- The only error that can find here is the confidence interval. For a 99% confidence interval and 95% confidence interval, it will have different number of points that required. That means, for a 99% confidence interval, the error will bigger since it will be considered more points that is not met the requirement. So, in order to get a more accuracy value, it needs to set a small confidence interval. The confidence interval is determined by adding the times of variance. The confidence interval will only set as a moderate value, if it is too big, then it may cause bigger error, but if it is too small, some correct points may be removed, this will also cause the error. Usually, here it will set as $\pm 2\sigma$, that it is 95% confidence interval.

Coordinate transform equation ---- As the above show, the earth is an imperfect sphere, the polar radius is shorter than the equatorial radius. So, even it used the sphere coordinate system here, it still has some errors. The best way to solve that is to use the WGS 84 coordinate. For further details, it will show on the improvement part.

8.2 Advantages

TDoA and k-NN method

There are two advantages here, the first one is the time-delay reduction. As the above shows, the TDoA method are use the time difference of two different sensors, and for one aircraft, it only has one receiver and response system, that it is to say, the time delay in a receiver is same for all signals. Besides, for an aircraft, it will only use one space-based ADS-B location system once. That means the time delay on a server will also same for the satellite data. So, by using TDoA method, it can reduce the time delay of receiver and satellite data. But the sensor in ground station has many different types, so it can only reduce some of the error using TDoA for ground station. In summary, for TDoA method, time-delay reduction is the advantages for it here.

Another strength for this method is sensor fusion, actually, this is also the advantages of EKF in following. But for TDoA and k-NN method, every sensor will used to confirm a location point combined with any other 3 sensors. That means for TDoA methods here, it will get the results from any combinations of sensors. And for k-NN, it will fuse all the point that get by TDoA and finally get an estimate position. So, here since it considered all the possibility and sensors, the advantage of it is the sensor fusion.

Additionally, since the TDoA and k-NN are all the simplest method. So, it is easy to construct and operational.

KF and GPS method

Here, in this method, the advantage is considered all the noise, this is also the advantage of EKF too. Since for a KF, not only the state and measurement, but also the process noise and measurement noise, they are all considered as a normal distribution. That means it will include all the errors and value of errors. Besides, since it has been included the velocity and last state. That means it can also reduce the measurement notable. And for an KF, it didn't have a complex structure, so it has a high robust and easy to develop by any languages. Besides, it has a predict step and an update step. So, it can also get the prediction without the measurement too, that it is also convenient and useful in inertial navigation.

EKF and RSSI method

This method has many advantages, it combined the advantages of k-NN and KF. Like a k-NN, the advantage of it is sensor fusion since it can combine many different sensors' measurement to get a comprehensive result, the result is the position data. And it also considered all the noise from the process to the measurement. In this project, it didn't consider the velocity. But for real world using, it will be considered that to get a better result. Besides, the KF is also a data fusion method since it can use a related value to estimate another value, here it is using the RSSI that it is related to power to get the position.

For RSSI data, the advantage of that it will not consider the time delay at all, since it is a method to estimate the distance by using power. And it has a higher accuracy than the TDoA since the order of magnitude is much higher and it is easy to find the error and adjust that.

RTS Smoother method

One advantage of it is same as the KF, since it is developed on KF and they have same parameters and similar methods. Another advantage is that further error reduced. Since it is developed on KF and use the estimation of KF. That means it can reduce the error based on and created KF. furthermore, since on the database, the first measurement position always used as the initial value of KF estimation. So, it can not able to reduce the error of the first value here, but RTS Smoother as a backward recursion. It can solve this problem.

Other further Optimization or simulation method

For pre filter and post filter, they are all used to reduce the errors and removed the wrong value. The advantage of that is simple and fast. It will remove the wrong value considered by the confidence interval of a polynomial regression function. So, it is easy to construct and adjustment. The MATLAB has been provided a tool to adjust the parameters of this polynomial regression. And compared with other method, it can calculate the analytic formula using the tool provided by MATLAB directly.

For the velocity simulation part, the advantage of it is also simple. It is easy to adjustment and can find the analytic formula to calculate the derivative fast.

8.3 Further Improvement

TDoA and k-NN method

There are 3 improvement that can do here, the first one is the coordinate that use here, as it shows on above, all the method are use the longitude and latitude based geographic coordinate. It is very inconvenient here since it is based on the unit of degree. But by using a distance to ensure the position, in this method, it is better to use a Cartesian coordinate system, and this system need to have a unit length at 1 km. So, on the process of calculate the position by using TDoA method. It will use the geocentric coordinate to estimation.

Another improvement here is that for TDoA method, though in some coding language, it can solve that directly, but it is still too hard to solve. So, in order to have a higher universality, it needs to be simplified. The following is the TDoA method that have 4 different sensors:

$$\begin{aligned} d_1 &= \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2} \\ d_2 &= \sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2 + (z_2 - z_0)^2} \\ d_3 &= \sqrt{(x_3 - x_0)^2 + (y_3 - y_0)^2 + (z_3 - z_0)^2} \\ d_4 &= \sqrt{(x_4 - x_0)^2 + (y_4 - y_0)^2 + (z_4 - z_0)^2} \end{aligned} \tag{8.3.1}$$

Here, it will use a method called least square method to estimate the position that determined by this groups of sensors based on TDoA method. Actually, the least square method can used in all sensors of that, but here in order to get a higher accuracy, it will use the method that it has been introduced above on TDoA first, and then use k-NN for further estimation.

By abandon the root sign on the right, and then all the rest equation subtracts the first equation, it can get that [52]:

$$\begin{aligned} d_2^2 - d_1^2 &= x_2^2 - x_1^2 - 2(x_2 - x_1)x_0 + y_2^2 - y_1^2 - 2(y_2 - y_1)y_0 + z_2^2 - z_1^2 - 2(z_2 - z_1)z_0 \\ d_3^2 - d_1^2 &= x_3^2 - x_1^2 - 2(x_3 - x_1)x_0 + y_3^2 - y_1^2 - 2(y_3 - y_1)y_0 + z_3^2 - z_1^2 - 2(z_3 - z_1)z_0 \\ d_3^2 - d_1^2 &= x_3^2 - x_1^2 - 2(x_3 - x_1)x_0 + y_3^2 - y_1^2 - 2(y_3 - y_1)y_0 + z_3^2 - z_1^2 - 2(z_3 - z_1)z_0 \end{aligned} \tag{8.3.2}$$

Rewrite it to a matrix form:

$$\begin{bmatrix} x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \\ x_4 - x_1 & y_4 - y_1 & z_4 - z_1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_2^2 + y_2^2 + z_2^2 - d_2^2 - (x_1^2 + y_1^2 + z_1^2 - d_1^2) \\ x_3^2 + y_3^2 + z_3^2 - d_3^2 - (x_1^2 + y_1^2 + z_1^2 - d_1^2) \\ x_4^2 + y_4^2 + z_4^2 - d_4^2 - (x_1^2 + y_1^2 + z_1^2 - d_1^2) \end{bmatrix} \tag{8.3.3}$$

It can be further implied: $\boldsymbol{Hx} = \boldsymbol{b}$

Here, $\boldsymbol{H} = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \\ x_4 - x_1 & y_4 - y_1 & z_4 - z_1 \end{bmatrix}$, $\boldsymbol{x} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}$ and $\boldsymbol{b} = \frac{1}{2} \begin{bmatrix} x_2^2 + y_2^2 + z_2^2 - d_2^2 - (x_1^2 + y_1^2 + z_1^2 - d_1^2) \\ x_3^2 + y_3^2 + z_3^2 - d_3^2 - (x_1^2 + y_1^2 + z_1^2 - d_1^2) \\ x_4^2 + y_4^2 + z_4^2 - d_4^2 - (x_1^2 + y_1^2 + z_1^2 - d_1^2) \end{bmatrix}$

By calculate this, the minimum variance solution is:

$$\boldsymbol{x} = \left(\boldsymbol{H^T H}\right)^{-1} \boldsymbol{H^T b} \tag{8.3.4}$$

The above is one solution of TDoA method.

Then, for k-NN network, since it is just using the distance of all the position that calculated by TDoA. Here it may consider some point that have a big error. In order to solve this, before the k-NN network, it can build another neural network, the weight of it will considered by the weather, the distance, the type of sensors and the time delay level. It is a multi-layer neural network and classified these points to different types. For the difference types of points, it will be adding a coefficient on the weight of k-NN. By doing this, it can reduce the errors to the minimum.

KF and GPS method

Here, the only improvement is for the Kalman Filter. Since on the KF above, the time step has been changed. So, the best improvement here is to change all the parameter that can be changed. Here, it can also use a multi-layer neural network that considered the influence of weather, the time step, the speed of aircraft, the aircraft type, the type of air-based location system and the errors of the ground station radar or sensor. Then given the different weight to the different influence type and finally adding that back to the datum value of these parameters. The datum value of that can use the value that calculated on the above, or it can simply set as 0 and then add the difference value on that.

EKF and RSSI method

Here since it is used the longitude and latitude based geographic coordinate, so a distance formula that similar to the Cartesian coordinate system is incorrect here since the unit of latitude and longitude is degree not the km. Additionally, the surface of earth is a curved surface not a flat surface. It may have some error to use a distance formula that may use on a flat to here, by using the transform equation in (7.1.1) and then bring back to (4.2.10). The real RSSI power loss exponential model in a longitude and latitude based geographic coordinate can be writing as. Notice, the $x_k(\alpha_k, \beta_k, h_k)$ is the location of aircraft and the $m^j(\alpha^j, \beta^j, h^j)$ is the sensor. Noticed that the N is the average radius of the earth.

$$h(x_k, m^j)$$
$$= P_0 - 10\gamma \log_{10} \sqrt{(h_k+N)^2 + (h^j+N)^2 - 2(h_k+N)(h^j+N)[\cos\beta_k \cos\beta^j \cos(\alpha^j - \alpha_k) + \sin\beta_k \sin\beta^j]} \tag{8.3.5}$$

For the parameters in RSSI model, it needs to estimate from the techque data of manufacturer, and for other parameters for EKF, since only the H is difference of that, and this can be solved by using the Jacobian matrix of the above function. So, the rest of that can use the same method like KF to solve that. Notice here it can use the neural network that same as KF since they will use the same sensors, in same recursion direction. And the error of that should be same.

RTS Smoother method

Since it is developed from the KF and it is a further smoother or optimization of KF, so the improvement of it is same as that in KF. Furthermore, since here this is a backward method, it is better to use the parameter as a value that difference with KF. Here, it can set an induvial neural network to re-estimate the variables. For the structure of this neural network. It can be same as that in KF.

Other further Optimization or simulation method

First it can improve is that the velocity simulation function. In above, it used the derivative of a polynomial regression function to estimate that. in some special place, it has a large error on that, so it is not suitable to use that in here if it needs a low error. By solving this problem, a new regression method that it is called smooth spline method has been used here.

The smooth spline method will use a partial polynomial regression which in each part it will have a different regression function, and in order to connect two partials, it needs to use some method to let the connection of that become smooth. In order to adjust it better, here it only has two different parameters, that first is the smoothing spline parameter p and next it is the specified weight w_i. The mathematical model of that is [70]:

$$f(x) = p \sum_i w_i \big(y_i - s(x_i)\big)^2 + (1-p) \int \left(\frac{d^2 s}{dx^2}\right)^2 dx \qquad (8.3.6)$$

Here, since in KF, EKF, it only needs to use the first velocity. This is a partial function. To get the initial velocity, it just needs to calculate the derivative for first part of that. For RTS Smoother, it will start from the last one, so it just needs to calculate the final one. MATLAB has a tool to regression by using this method, the following is the regression result of aircraft 1602. But it cannot show the analytic formula of that. So, in real using, it needs to use the above function to calculate that.

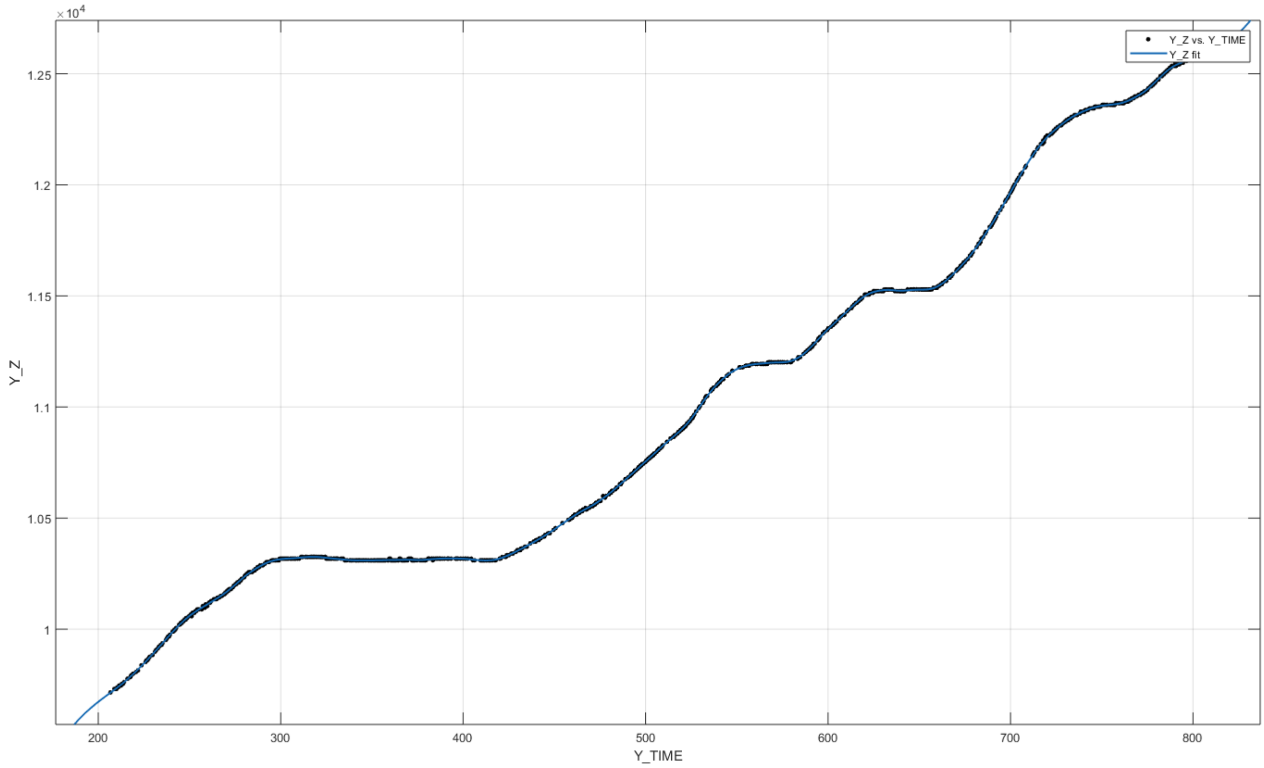


Figure 8.2 Regression using smooth spline of altitude
Since the polynomial regression are good at longitude and latitude, so it will not use the method to regression that two value.

As it shows on the figure, this is much better for using polynomial as it can see in the start and end, the change of the regression line is very stable.

Another method that it needs to improve is that the coordinate, since it has just said that on some of the method, it can use the geocentric coordinate to replace the geographic coordinate. But all the replacement is considered the earth as a perfect sphere. Actually, the earth is an ellipsoid that the polar radius is smaller than the equatorial radius. So, to get higher accuracy, it is better to use WGS 84 Systems. The following is the transform equation from geocentric coordinate(x, y, z) and WGS 84 (α, β, h) [51]:

$$\begin{cases} x = (N+h)\cos\alpha\cos\beta \\ x = (N+h)\cos\alpha\sin\beta \\ x = (N(1-e^2)+h)\sin\alpha \end{cases} \qquad (8.3.7)$$

Here, $N = \frac{r}{\sqrt{1-e^2\sin^2\alpha}}$, r is the equatorial radius of earth where $r = 6378.137\ km$ and e is the eccentricity where $e = 0.081819190842$.

9 Conclusions

In summary, here an algorithm that based on TDoA, k-NN, KF, EKF, RSSI, RTS Smoother and ADS-B data has been developed. This algorithm has three different parts, the first one is to use TDoA and k-NN to estimate the initial position for those aircraft that didn't provide position information. Next, there are two difference method to estimate the route of the aircraft, the first is to use the KF and GPS data, this can just use for those aircrafts that has position value in database. Next kind is to use EKF and RSSI data, it can used in all situations when the initial position of the aircraft has been known. And finally, the RTS Smoother will used to further optimize the result. For some data that obviously have mistaken, in this method, it has an extra filter to remove that. This method has some ability to reduce some errors like the time delay and measurement noise if all the parameters are well-adjusted. But it can still have some improvement like use a WGS 84 coordinate, use smooth spline to simulate the velocity data or using neural network to considered more condition so that it can get better parameters.

Additionally, due to the limitation of computer performance and data quantities, here it can only simulate the KF, EKF and RTS Smoother method on MATLAB. For the TDoA and k-NN method, it has some theoretical approach and modelling here.

Reference

[1] W. R. Richards, K. O'Brien and D. C. Miller, "New Air Traffc Surveillance Technology," *Boeing Aero Quarterly,* vol. 2, no. 10, pp. 07-13, 2014.

[2] Airservices, "Airservices Australia," 18 9 2015. [Online]. Available: http://www.airservicesaustralia.com/projects/ads-b/how-ads-b-works/. [Accessed 4 4 2020].

[3] GPS, "GPS Aviation Application," 14 6 2019. [Online]. Available: https://www.gps.gov/applications/aviation/. [Accessed 4 4 2020].

[4] FAA, "Navigation Programs — Satellite Navigation," 12 3 2020. [Online]. Available: https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/. [Accessed 4 4 2020].

[5] E. A. Lester, "Benefits and Incentives for ADS-B Equipage in the National Airspace System," MIT, Boston, 2007.

[6] J. Scardina, "Overview of the FAA ADS-B Link Decision," FAA, Washington DC, 2002.

[7] FAA, "Automatic Dependent Surveillance—Broadcast (ADS–B) Out Performance Requirements To Support Air Traffic Control (ATC) Service," *Federal Register,* pp. 30160-30195, 28 5 2010.

[8] M. Kastelein, "Preliminary Analysis of ADS-B Performance for Use in ACAS Systems," Ohio University, Colorado Springs, 2014.

[9] B. Prince, "Security Week," Wired Business Media, 27 7 2012. [Online]. Available: https://www.securityweek.com/air-traffic-control-systems-vulnerabilities-could-make-unfriendly-skies-black-hat. [Accessed 4 4 2020].

[10] A. Costin and A. Francillon, "Ghost in the Air (Traffic): On insecurity of ADS-B protocol and practical attacks on," EURECOM, Sophia-Antipolis, 2012.

[11] D. McCallie, J. Butts and R. Mills, "Security analysis of the ADS-B implementation in the next generation air transportation system," *INTERNATIONAL JOURNAL OF CRITICAL INFRASTRUCTURE PROTECTION,* pp. 78-87, 7 7 2011.

[12] S.-H. Lee, Y.-K. Kim, J.-W. Han and D.-G. Lee, "Protection Method for Data Communication between ADS-B Sensor and Next-Generation Air Traffic Control Systems," *information,* pp. 622-633, 1 12 2014.

[13] K. D. Wesson, T. E. Humphreys and B. L. Evans, "Can Cryptography Secure Next Generation," IEEE SECURITY & PRIVACY, New York, 2014.

[14] M. Geyer, "EARTH-REFERENCED AIRCRAFT NAVIGATION AND SURVEILLANCE ANALYSIS," FAA, Washington, 2016.

[15] Y. Chai, "Research on three-dimensional positioning algorithm of wireless sensor network nodes," Shangdong University, Jinan, 2011.

[16] A. Malik, "Tag-A-Palooza: RTLS Technology Tour," in *RTLS For Dummies*, Danver, Wiley, 2009, pp. 66-70.

[17] Aireon, "Space Based ADS-B," Aireon, McLean, 2016.

[18] M. S. Grewal, L. R. Weill and A. P. Andrews, "Fundamentals of Satellite and intertial navigation," in *Global Positioning Systems, Inertial Navigation, and Integration*, Danvers, Wiley, 2007, pp. 19-28.

[19] B. W. Parkinson and J. J. Spilker, “Introductory GPS System Description and Technical Design,” in *Global Positioning System: Theory and Applications*, Washington D.C., AIAA, 1996, pp. 10-20.

[20] China Satellite Navigation Office, “Update on BeiDou Navigation Satellite System,” Twelfth Meeting of the International Committee on Global Navigation Satellite Systems, Koyoto, 2017.

[21] ESA, “Fact Sheet for Galileo System,” ESA, Paris, 2013.

[22] C. J. Hegarty and E. Chatre, “Evolution of the Global Navigation Satellite System (GNSS),” *Proceedings of the IEEE,* pp. 1902-1917, 16 1 2009.

[23] N. S. Kuzmenko, I. V. Ostroumov and K. Marais, “An Accuracy and Availability Estimation of Aircraft Positioning by Navigational Aids,” IEEE, Kyiv, 2018.

[24] M. Paulsson, “High-Level Control of UAV Swarms with RSSI Based Position Estimation,” Lund University, Lund, 2017.

[25] O. Goodhue and P. R. Doral, “RADIA NAVIGATION,” in *Flying Operations*, Joint Base Langley–Eustis, US Air Force, 2001, p. 173.

[26] C. Perkins, L. Lei, M. Kuhlman, T. H. Lee, G. Gateau, S. Bergbreiter and P. Abshire, “Distance Sensing for Mini-robots: RSSI vs. TDOA,” IEEE, Washington D.C., 2011.

[27] A. T. Parameswaran, M. I. Husain 和 S. Upadhyaya, “Is RSSI a Reliable Parameter in Sensor Localization Algorithms – An Experimental Study,” State University of New York at Buffalo, New York, 2009.

[28] A. Kelly, “Kalman Fiter Noise Model,” in *A 3D State Space Formulation of a Navigation Kalman Filter for Autonomous Vehicle*, Pittsburgh, Carnegie Mellon Univesity, 1994, pp. 6-7.

[29] S. Särkkä, “Bayesian Filtering Equations and Exact Solutions,” in *Bayesian Filtering and Smoothing*, Cambridge, Cambridge University Press, 2013, pp. 51-63.

[30] P. Zarchan and H. Musoff, “Polynomial Kalman Filte,” in *Fundamentals of Kalman Filtering: A Practical Approach*, Reston, AIAA, 2015, pp. 107-121.

[31] L. R. Rabiner and B. H. juang, “An Introduction to Hidden,” *iEEE ASSP MAGAZINE,* pp. 4-16, 16 1 1986.

[32] Headlessplatter, “Kalman filter model,” 7 4 2009. [Online]. Available: https://en.wikipedia.org/wiki/Kalman_filter#/media/File:Kalman_filter_model_2.svg. [Accessed 11 4 2020].

[33] G. Welch and G. Bishop, “An Introduction to the Kalman Filter,” Welch & Bishop, Chapel Hill, 2006.

[34] Y. H. Yang and Y. Shi, “Application of Improved BP Neural Network in Information,” *Circuits, Systems, and Signal Processing,* pp. 1-13, 9 3 2020.

[35] S. Särkkä, “Extended and unscented Kalman Filter,” in *Bayesian Filtering and Smoothing* , Cambridge, Cambridge University Press, 2013, pp. 64-95.

[36] J. Wang and X. Luo, “Research on Airborne Passive Location Based on Extend Kalman Filter With,” *2016 3rd International Conference on Information Science and Control Engineering,* pp. 1389-1392, 8 7 2016.

[37] S. Särkkä, “Bayesian Smoothing equations and exact solutions,” in *Bayesian Filtering and Smoothing*, Cambridge, Cambridge University Press, 2013, pp. 134-142.

[38] M. Y. Jiang, S. J. Jang and J. W. Chang, “A New k-NN Query Processing Algorithm Enhancing Privacy Protection in Location-Based Serivers,” *2012 IEEE First International Conference on Mobile Services,* pp. 17-24, 9 8 2012.

[39] C. Zhang, C. Wang, Y. Wei and J. Wang, “Neural network adaptive position tracking control of underactuated autonomous surface vehicle,” *Journal of Mechanical Science and Technology,* pp. 855-865, 5 12 2019.

[40] FAA, “Automatic Dependent Surveillance-Broadcast (ADS-B),” FAA, 10 3 2020. [Online]. Available: https://www.faa.gov/nextgen/programs/adsb/. [Accessed 11 4 2020].

[41] Flightadar24, “How flight tracking works,” Flightadar24, 15 3 2020. [Online]. Available: https://www.flightradar24.com/how-it-works/. [Accessed 11 4 2020].

[42] J. DAVIDSON, “ADS-B UPDATE 2020 – WHAT YOU NEED TO KNOW,” Universal Weather and Aviation Inc., 25 2 2020. [Online]. Available: https://www.universalweather.com/blog/ads-b-update/. [Accessed 11 4 2020].

[43] C. Fredericks, “Breakthrough technology brings air traffic surveillance to Hudson Bay,” WINGS, 26 1 2009. [Online]. Available: https://www.wingsmagazine.com/breakthrough-technology-brings-air-traffic-surveillance-to-hudson-bay-2369/. [Accessed 11 4 2020].

[44] N. Canada, “TERMS OF REFERENCE: Canadian ADS-B Out Performance Requirement Mandate,” NAV Canada, Ottawa, 2017.

[45] S. Riisgaard and M. R. Blas, “The SLAM Process,” in *SLAM for Dummies*, Danver, Wiley, 2011, pp. 10-13.

[46] A. F. Garcia-Fernandez, Y. Xia, K. Granstrom, L. Svensson and J. L. Williams, “Gaussian implementation of the multi-Bernoulli mixture filter,” arXiv, Ithaca, 2019.

[47] F. Tronarp, Á. F. García-Fernández and S. Särkkä, “Iterative Filtering and Smoothing In Non-Linear and Non-Gaussian Systems Using Conditional Moments,” *IEEE SIGNAL PROCESSING LETTERS,* pp. 408-412, 22 1 2018.

[48] A. F. Garcia-Fernandez, R. Hostettler and S. Sarkka, “Rao-Blackwellized Posterior Linearization Backward SLAM,” *IEEE Transactions on vehicular technology,* vol. 68, no. 5, pp. 4734-4747, 5 5 2019.

[49] W. Peng, R. Huang, J. Guo, K. Jiang and D. He, “A location algorithm suitable for MLAT surveillance system at airport surface and its performance analysis,” *Acta Aeronautica et Astronautica Sinica,* pp. 3050-3059, 25 9 2015.

[50] M. Strohmeier, I. Martinovic and V. Lenders, “A k-NN-Based Localization Approach for Crowdsourced Air Traffic Communication Network,” *IEEE Transactions on aerospace and electronic systems,* pp. 1519-1529, 7 6 2018.

[51] B. Deng, L. Yang, Z. Sun and H. Peng, “Geolocation of a Known Altitude Target Using TDoA and GRoA in the Presence of Receiver Location Uncertainty,” *Hindawi,* pp. 1-11, 17 8 2016.

[52] F. Yang, “Research on Indoor WIFI Positioning Technology Based on Improved KNN Algorithm,” Northwestern Polytechnical University, Xi'an, 2016.

[53] T. Abeywickrama, M. A. Cheema and D. Taniar, “k-Nearest Neighbors on Road Networks: A Journey in Experimentation and In-Memory Implementation,” arXiv, Melboume, 2016.

[54] F. G. Stremler, “Angle Modulation,” in *ntroduction to communication systems [by] Ferrel G. Stremler*, Reading, Addison-Wesley Pub. Co., 1996, pp. 250-273.

[55] Opensky Network, “Documentation - Aircraft Localization Competition,” Opensky Network, 4 4 2019. [Online]. Available: https://competition.opensky-network.org/documentation.html. [Accessed 15 4 2020].

[56] P. Vaníček and R. R. Steeves, “Transformation of coordinates between two horizontal geodetic datums,” *Journal of Geodesy,* pp. 740-745, 6 11 1996.

[57] Y. Liu, “Design and Implementation of Obstacle Assessment in Flight Route,” *Command Information System and Technology,* vol. 3, no. 6, pp. 25-29, 10 9 2012.

[58] R. Dou, M. Hu and Y.-j. Wang, “Demand Based Dynamic Air Route Management,” *Aeronautical Computing Technique,* vol. 38, no. 6, pp. 25-27, 31, 2008.

[59] C. Cutler, “The 3 Types Of Static And Dynamic Aircraft Stability,” Boldmethod, 26 03 2015. [Online]. Available: https://www.boldmethod.com/learn-to-fly/aerodynamics/3-types-of-static-and-dynamic-stability-in-aircraft/. [Accessed 16 4 2020].

[60] M. S. Aktürk, A. Atamtürk and S. Gürel, “Aircraft Rescheduling with Cruise Speed Control,” *OPERATIONS RESEARCH,* vol. 62, no. 4, pp. 829-845, 23 5 2014.

[61] Boeing, “Technical Specs for Boeing 737,” Boeing, 1 1 2020. [Online]. Available: http://www.boeing.com/commercial/737ng/. [Accessed 16 4 2020].

[62] Airbus, “AIRCRAFT PERFORMANCE,” in *A350 AIRCRAFT CHARACTERISTICS AIRPORT AND MAINTENANCE PLANNING*, Blagnac Cedex, AIRBUS S.A.S, 2019, pp. 152-165.

[63] Clarkson University, *AE 429 - Aircraft Performance and Flight Mechanics,* Potsdam: Clarkson University, 2009.

[64] Y. Liu, Z. Liu and L. Lin, “Analysis and test of ADS-B message validity based on RSSI,” *Journal of Electronic Measurement and Instrumentation,* vol. 1, no. 9, pp. 37-45, 2018.

[65] T. M. Hamill, J. S. Whitaker and C. Snyder, “Distance-Dependent Filtering of Background Error Covariance Estimates in an Ensemble Kalman Filter,” *MONTHLY WEATHER REVIEW,* vol. 129, no. 11, pp. 2776-2790, 2001.

[66] T. Nishimura, “Modeling Errors in Kalman Filters,” NASA, Pasadena, 1970.

[67] National Geospatial-Intelligence Agency, “World Geodetic System,” National Geospatial-Intelligence Agency, 2 4 2012. [Online]. Available: https://www1.nga.mil/ProductsServices/GeodesyandGeophysics/WorldGeodeticSystem/Pages . [Accessed 17 4 2020].

[68] H. Hrasnica, A. Haidine and R. Lehnert, “Specific PLC Performance Problems,” in *Broadband Powerline Communications: Network Design*, Chichester, Wiley, 2004, pp. 31-38.

[69] E. Notes, “Ground Wave Propagation Tutorial,” Electronic Notes, 1 5 2019. [Online]. Available: https://www.electronics-notes.com/articles/antennas-propagation/ground-wave/basics-tutorial.php. [Accessed 18 4 2020].

[70] MATLAB, “Smoothing Splines,” MathWorks, 2 3 2020. [Online]. Available: https://www.mathworks.com/help/curvefit/smoothing-splines.html. [Accessed 18 4 2020].