TUMSAT-OACIS Repository - Tokyo University of Marine Science and Technology (東京海洋大学)

Performance evaluation of GNSS/INS integration

メタデータ	言語: English 出版者: 公開日: 2021-11-30 キーワード (Ja): キーワード (En): 作成者: 郭, 暁亮 メールアドレス: 所属:
URL	https://oacis.repo.nii.ac.jp/records/2255

Master's Thesis

PERFORMANCE EVALUATION OF GNSS/INS INTEGRATION

September 2021

Graduate School of Marine Science and Technology

Tokyo University of Marine Science and Technology

Master's Course of Maritime Technology and Logistic

GUO XIAOLIANG

Master's Thesis

PERFORMANCE EVALUATION OF GNSS/INS INTEGRATION

September 2021

Graduate School of Marine Science and Technology

Tokyo University of Marine Science and Technology

Master's Course of Maritime Technology and Logistic

GUO　XIAOLIANG

LIST OF ABBREVIATIONS

The list is arranged by the order of the appearance

VNS	Vehicle navigation system
GNSS	Global navigation satellite system
INS	Inertial Navigation System
IMU	Inertial Measurement Unit
WSS	Wheel speed sensor
VNS	Vehicle navigation system
MEMS	Micro-Electro-Mechanical System
LC	Loosely coupled
TC	Tightly coupled
DGNSS	Differential GNSS
RTK	Real - time kinematic
GPS	Global positioning system
QZSS	Quasi zenith satellite system
GLONASS	The global navigation satellite system of Russia
BDS	Chinese Beidou navigation satellite system
GALILEO	Galileo satellite navigation system
DOP	Dilution of precision
NASA	National Aeronautics and Space Administration
KF	Kalman filter
EKF	Extended Kalman filter
UKF	Unscented Kalman filter
ENU	East-North-Up
PRN	Pseudo random noise
SNR	Signal-to-noise ratio
ECEF	Earth-centered, earth-fixed
ECI	Earth-centered inertial
PSD	Power spectral density
SD	Stander deviation
PDOP	Position dilution of precision
IAE	Innovation-based adaptive estimation
LS	Least square
St.Dev	Stander deviation
95-th Pc.ile	95th percentile

LIST OF FIGURES

Figure 2-1 Differential GNSS 7
Figure 2-2 GNSS error sources 9
Figure 2-3 Local tangent plane coordinate and body coordinate 10
Figure 2-4 Station center coordinate and ECEF coordinate system 11
Figure 2-5 Navigation equation in ECEF 13
Figure 2-6 Error model of hypothetical accelerometer 18
Figure 2-7 6 stable positions for accelerometer calibration 19
Figure 2-8 Acceleration alignment 19
Figure 2-9 Comparison before and after calibration 20
Figure 2-10 Heading from GNSS velocity 20
Figure 3-1 Kalman filter data flow 22
Figure 3-2 Wrong position solution in horizontal direction 22
Figure 3-3 Loosely coupled flow chart 32
Figure 3-4 Tightly coupled flow chart 48
Figure 3-5 Three sections IAE-KF 51
Figure 4-1 Wheel speed sensor working principle 53
Figure 4-2 Barometer height and RTK height in ENU 54
Figure 4-3 Estelle IMU temperature and barometer 55
Figure 4-4 Barometer height and RTK height in ENU 55
Figure 4-5 GNSS compass 56
Figure 5-1 Setup mounted on board of a car 58
Figure 5-2 RTK result in Tsukishima 58
Figure 5-3 Comparison before and after adding only velocity update 59
Figure 5-4 Compare Classical KF and IAE-KF 60
Figure 5-5 Trajectory followed during the experimental tests in Tokyo 61
Figure 5-6 Tsukishima dataset track 62
Figure 5-7 Tsukishima dataset track in No. 1 place 63
Figure 5-8 Yaw angles under the viaduct 65
Figure 5-9 Tsukishima dataset track in No. 3 place 66
Figure 5-10 Tsukishima dataset track in No. 4 place 67
Figure 5-11 Tsukishima 20210412 error 68
Figure 5-12 Tsukishima 20201105 error 70
Figure 5-13 Marunouchi track of 0302 1st test 71
Figure 5-14 Marunouchi error of 0302 1st test 73
Figure 5-15 Marunouchi track of 0302 2nd test 74
Figure 5-16 Marunouchi error of 0302 2nd test 75
Figure 5-19 Multi-sensor fusion flow chart 77

Figure 5-20 Multi-Sensor fusion result in the Shuto Expressway..... 77
Figure 5-21 Multi-Sensor fusion result in the Marunouchi 2nd..... 79

LIST OF TABLES

Table 1-1 Global navigation system frequencies 1
Table 1-2 Some open source code for GNSS/INS Tight coupled 4
Table 2-1 Difference in ECI and ECEF 10
Table 2-2 The acceleration scale factor of hexahedral rotation 19
Table 5-1 List of the equipment 57
Table 5-1 Error of Classical KF and IAE-KF in horizontal 61
Table 5-2 Error of Tsukishima 70
Table 5-3 Error of Marunouchi 76
Table 5-4 Error of different methods 79

Table of Contents

1. INTRODUCTION 1
1.1. Background 1
1.1.1 Global Navigation System 1
1.1.2 INS 2
1.1.3 GNSS/INS 2
1.2. Previous Research 3
1.3. Contributions 4
1.4. Outline 4
2. OVERVIEW OF GNSS AND INS 6
2.1. GNSS measurements 6
2.1.1 Pseudorange Measurement 6
2.1.2 Carrier-Phase Measurement 6
2.1.3 Doppler Measurement 7
2.2 GNSS Error 7
2.2.1 Ionospheric Error 8
2.2.2 Troposphere Error 8
2.2.3 Satellite and Receiver Clock Errors 8
2.2.4 Multipath Effect 9
2.3 Coordinate Frames Used in Inertial Navigation 9
2.3.1 Definition of Coordinate System 9
2.3.2 Transformation Relation of Coordinate System 11
2.4 Mechanization Equations 12
2.5 IMU Error 15
2.5.1 Sensor Bias Offset 16
2.5.2 Scale Factor Error 16
2.5.3 Installation Error 16
2.5.4 Calibration of Internal Parameters 17
2.6 Initial Alignment 20
3. GNSS/INS INTEGRATION 21
3.1 Kalman Filter 21
3.2 GNSS Error Model 22
3.3 INS Error Model 23
3.4 GNSS/INS Loosely Coupled Methods 26
3.5 GNSS/INS Tightly Coupled Methods 32
3.6 Optimization of Integrated Navigation Algorithm 48
3.6.1 Optimization Algorithm for GNSS/INS Loosely Coupled 48

3.6.2 Optimization Algorithm for GNSS/INS Tightly Coupled 50
4. MULTI-SENSOR FUSION VEHICLE ASSIST 53
4.1 Vehicle Motion Model 53
4.2 Wheel Speed Sensor 53
4.3 Barometer 54
4.4 Zero Angular Rate Update 55
4.5 GNSS Compass 56
5. EXPERIMENT AND RESULT 57
5.1 Compare the with velocity and without velocity update 58
5.2 Compare the Classical KF and IAE-KF 59
5.3 Compare the GNSS/INS LC and GNSS/INS TC 61
5.3.1 Tsukishima Dataset 61
5.3.2 Marunouchi Dataset 71
5.3 Multi-Sensor Fusion 76
6. CONCLUSION 80
Reference 81
Acknowledgment 84

1. INTRODUCTION

1.1. Background

This section provides an overview of the navigation satellite systems currently in operation, the research and development of inertial navigation and the research status of integrated navigation.

1.1.1 Global Navigation System

The whole GNSS constellation can use more than 120 satellites, including the global positioning system (GPS) of the United States, the quasi zenith satellite system (QZSS) of Japan, and the global navigation satellite system (GLONASS) of Russia, Chinese Beidou navigation satellite system (BDS) and the Galileo satellite navigation system (GALILEO) developed by the European Union[1, 2].

GNSS navigation method is to use satellites to send signals in different frequency bands to provide different services. At the same time, GNSS satellite is equipped with an on-board atomic clock, which can provide accurate time service. It is the basic method of satellite navigation to solve the pseudo range by using the time difference of the received satellite signal. The position can be solved by using the data from any three of the four satellites, and the other satellite data is used to solve the clock offset of the receiver[3, 4].

Table 1-1 Global navigation system frequencies

GPS	
L1-C/A, L1C	1575.42 MHz
L2	1227.60 MHz
L5	1176.45 MHz
GAILIEO	
E1	1575.42 MHz
E5	1191.795 MHz
E5A	1176.45 MHz
E5B	1207.14 MHz
E6	1278.75 MHz
GLONASS	
L1(FDMA)	1602 MHz + n * 0.5625 MHz
L2(FDMA)	1246 MHz + n * 0.4375 MHz
L3(CDMA)	1202.025 MHz
BEIDOU	
BL-2	1589.742 MHz
B1I	1561.098 MHz
B3	1268.52 MHz
B2I	1207.14 MHz
QZSS	
L1-C/A, L1C	1575.42 MHz
L2C	1227.60 MHz
L5	1176.45 MHz

According to the known position and pseudo range of the satellite, the estimated position and time can obtain more satellites that can provide navigation services by using the GNSS satellite constellation

compared with using a single satellite system as the satellite navigation source, which makes it possible to have more visible satellites in dense cities or other difficult terrain areas, and because of the lower DOP (dilution of precision), it has higher positioning accuracy. Although more satellites may not improve the positioning accuracy, the geometry of the satellites used will be better. GNSS will be used as the source of satellite navigation in the current work. There are two situations when GNSS is used as the source of satellite navigation. The first is that when GNSS is applied to land vehicles in open area, because the satellite signal is not obstructed, it has ideal effect[5]. In the second case, for example, in the urban environment, there are usually dense leaves, rugged terrain, tunnels, bridges or high-rise buildings. At this time, the signal cannot pass or bypass the obstacles, resulting in signal blocking, which leads to the GNSS receiver unable to continuously provide navigation position for vehicles. Therefore, when the satellite signal is blocked, other navigation methods are needed to provide supplement, such as INS which is not easy to be interfered by the outside world[6].

1.1.2 INS

INS is based on Newton's mechanics. Inertial sensor only measures the rate of change. It is necessary to integrate the output signal to obtain position and direction data, and combine the integrated results to provide navigation position. Inertial sensor is not easy to be interfered by the outside world, does not rely on the external information, can provide the inertial measurement value continuously, and is widely used in navigation system[7, 8]. Compared with GNSS alone to provide navigation results, the inertial sensor provides the supplementary function when there is no satellite signal or the satellite signal is poor. The ideal inertial sensor does not need external information except the initial attitude estimation. It can provide a good signal-to-noise ratio when the direction changes rapidly and the speed is high. Due to the influence of noise or some bias, the cumulative drift will be significant when the integration time is longer[9]. When there is obvious temperature change, the sensor usually needs to be recalibrated.

MEMS is emerging in inertial navigation technology. After more than 40 years of development, MEMS has gradually entered the core vision of the public[10]. In recent years, the progress of MEMS technology makes the complete inertial unit composed of accelerometer and gyroscope can be built on chip. MEMS features: 1) fast start, 2) low power consumption, 3) light weight, 4) low cost, can meet the requirements and specifications required by commercial applications, such as vehicle navigation[11]. However, because of the immature MEMS technology and the limited performance of sensors, the INS performance of MEMS as an inertial measurement unit does not meet the accuracy requirements of many navigation applications. It is necessary to provide irregular calibration for INS based on MEMS and limit its error to acceptable water level. Satellite navigation can provide a reference for INS calibration based on MEMS. Among many integrated navigation methods, GNSS and INS are the most common[6, 10]. The combination of satellite and inertial navigation can improve the accuracy of vehicle navigation, and reduce the development and use cost of navigation system, and there are huge application prospects in many fields.

1.1.3 GNSS/INS

Because different navigation systems have different characteristics, using GNSS / INS integrated navigation system can combine two or more different navigation systems. By measuring the same

information source, the measured values of different navigation systems are compared, and some correction method is used to correct the system error. GNSS / INS integrated navigation system combines the two subsystems by using the method of data fusion to realize the complementary advantages of each subsystem, so as to improve the accuracy and reliability of the system[12]. Integrated navigation technology has been paid more and more attention by scholars since 1980s. Using the integration of satellite system and inertial navigation system can provide users with high reliability and high precision navigation information, which has been recognized by the navigation industry. After many years of development, coupled with the rapid development of satellite positioning technology, computer technology, inertial positioning technology and modern mathematical theory, GNSS / INS integrated technology has made great progress. GNSS information contains pseudo range, carrier and Doppler information, and provides a variety of combination schemes. At present, the common combination methods are loose coupling, tight coupling and ultra-tightly coupling[1].

1.2. Previous Research

In 1959, NASA (National Aeronautics and Space Administration) began to study the manned spacecraft landing program. The main solution to the problem of landing on the moon is to estimate the motion state of the spacecraft. At the beginning, recursive least square method and Wiener filter were used in the estimation method, but they were abandoned because of low accuracy and complex calculation. In 1960, Kalman R.E. visited NASA and proposed Kalman filter (KF) algorithm for this problem[13]. The object of Kalman filtering theory is Gaussian linear system, and the measurement information or state equation must be linear. The filtering theory is a standard Kalman filter. After that, researchers continue to optimize the Kalman filter. Based on the standard Kalman filter, they propose the UDU^T decomposition filtering algorithm and the Potter and Carlson square root filtering algorithm[14, 15], which ensures that the filtering variance matrix in the Kalman filter gain loop remains positive definite, and greatly improves the filtering performance. For nonlinear eigenstate estimation, Bucy and Sunahara proposed extended Kalman filter (EKF)[16]. The idea of EKF is to linearize the model of nonlinear system and approximate the system, which will inevitably lead to linearization error. In order to solve the estimation problem under strong nonlinearity, S.J., Julier and J.K. Uhlmann proposed unscented Kalman filter (UKF) Algorithm in 1955, which was later improved by E.A. Wan and R. Vander Merwe[17, 18].

In the vehicle GNSS / INS Integrated Navigation and positioning system, according to the depth of fusion degree, the most commonly used fusion methods of GNSS and INS are LC and TC[19]. LC is the combination of GNSS and INS in position, velocity and attitude. TC is the combination of GNSS and INS in pseudo range, pseudo range rate and carrier phase. Compared with loose coupling, the biggest advantage of TC is that even if the number of satellites does not meet the requirements of positioning, the whole system can still complete the positioning work normally. The GNSS / INS TC navigation and positioning methods used by researchers can make full use of the received satellite signals to correct the INS positioning error, avoid the INS independent positioning for a long time and reduce the accuracy, and realize the continuous and reliable vehicle positioning. Because tight coupling can keep the continuity of positioning, it is widely concerned in the field of vehicle positioning.

At present, the rich network resources give the integrated navigation beginners a lot of valuable

resources[20-23]. Thanks to GitHub and other websites for providing open source code, learners can freely share and discuss each other's code, and mainland academic problems in various industries are also solved in issue. Professor Paul groves published a book called "principles of GNSS, inertial, and Multi-sensor Integrated Navigation Systems" in 2015[1], which has influenced many navigation scholars. In addition, the GNSS signal processing part of the navigation software is completed by rtklib by Mr. Takasu from Tokyo university of marine science and technology[24]. There are many LC codes on the Internet. Here are the known open source TC codes.

Table 1-2 Some open source code for GNSS/INS Tight coupled

Name	URL	Language	Environment
gnssins	https://github.com/marcoamm/gnssins	C 95%, Fortran 4.8%	Linux
ignav	https://github.com/Erensu/ignava	C 64.9%, C++ 34.5%	Linux
TightlyCoupled INSGNSS	https://github.com/benzenemo/TightlyCoupledINSGNSS	MATLAB 100%	Windows
GINav	https://github.com/kaichen686/GINav	MATLAB 100%	Windows

1.3. Contributions

The contribution of this work can be summarized as follows:

- Modified the source code from "principles of GNSS, inertial, and Multi-sensor Integrated Navigation Systems", and using the source code for laboratory data set;
- Changed the loosely coupled Kalam filter parts for position update, velocity update, position and velocity update, and position, velocity, and attitude update;
- Designed a loosely coupled program in East-North-Up (ENU) coordination, and using barometer to detect the wrong RTK solution;
- Based on vehicle motion model, using attitude and WSS for dead-reckoning when GNSS not available.

1.4. Outline

Chapter 1 introduces the GNSS and INS research. This paper mainly talking about the difference between LC and TC, so the referenced source codes were introduced and made a list.

Chapter 2 shows the GNSS positioning principle and INS mechanization equation. And analyzes the GNSS and INS error. During the LC and TC, there are lots of coordination transformations, so introduces each coordination. Finally, using the 6 stable positions for accelerometer calibration, to check the IMU installation error.

Chapter 3 introduces the GNSS/INS integration methods. The KF methods base on the INS error estimates the INS errors, and using GNSS measurements as KF’s measurement values, update the INS error estimation. Compare the GNSS/INS integration methods with formula and MATLAB code in each step. Finally, introduces some optimization methods to modify the LC and TC.

Chapter 4 introduces the multi-sensor fusion ideas. When GNSS information is not available, these methods can provide the velocity measurement and resist the influence of GNS error.

Chapter 5 shows the results of LC and TC. According to compare the position error and the yaw angles in different environments, it is easy to draw the conclusion of the advantages and disadvantages between LC and TC. Finally, using multi-sensor fusion result shows the error under the viaduct.

Chapter 6 summarizes the conclusion of the performance evaluation in GNSS/INS integration, planning the next step of research.

2. OVERVIEW OF GNSS AND INS

2.1. GNSS measurements

Global navigation satellite system (GNSS) is the general name of global positioning system (GPS) of the United States, quasi zenith satellite system (QZSS) of Japan, GLONASS of Russia, Beidou System (BDS) of China and Galileo satellite system of Europe. The principle of navigation and positioning system is similar, which uses pseudo range, carrier phase or Doppler observations to location.

2.1.1 Pseudorange Measurement

Pseudo range measurements include the time required for the signal to travel from the satellite to the receiver multiplied by the speed of light. From space to land, this distance experienced the refraction and reflection of ionosphere, troposphere, city buildings and lake surface. Many factors cause ranging error, such as the clock of satellite and receiver is not completely synchronized with constellation time base. Let's take GPS as an example[1].

The original satellite signal is sent through two or more predefined frequencies. These signals are modulated by navigation information (50 bps) and C / A code (transmitted at 1.023 MHz on L1) and P (Y) code (transmitted at 10.23 MHz on L1 and L2). The C / A code is modulated in the orthogonal part of the signal, and the P (Y) code is modulated in the in-phase part of the signal. The receiver knows what the unique code (called pseudo-random noise, PRN code) of each satellite is. When the original signal reaches the processing part of the receiver, the received PRN code is shifted until it is aligned with the internal PRN code. The time offset required to arrange the two PRN codes is the movement time of the signal. The pseudo range equation is as follows:

$$P_{(m)} = \rho + c(\delta T - \delta t) + \delta_{\text{ion}} + \delta_{\text{trop}} + \varepsilon_{\text{code}} \tag{2-1}$$

Where:

$P_{(m)}$ is the measured pseudorange (meters)

ρ is the true range between the satellite and receiver antenna (meters)

c is the speed of light in vacuum (m/s)

δT is the receiver clock error (seconds)

δt is the satellite clock error (seconds)

δ_{ion} is the ionosphere induced error (meters)

δ_{trop} is the troposphere induced error (meters)

$\varepsilon_{\text{code}}$ is the code level multipath plus noise (meters)

2.1.2 Carrier-Phase Measurement

For relative positioning, at least four satellites should be visible at the same time, which is similar to single point positioning. The base station and receiver with known precise coordinates need to receive observation data at the same time, and their distance needs to be kept within a certain range. RTK uses carrier phase observations for positioning, which can get high accuracy positioning results, usually up to centimeter level[25]. Figure 2-1 shows the relative positioning diagram:

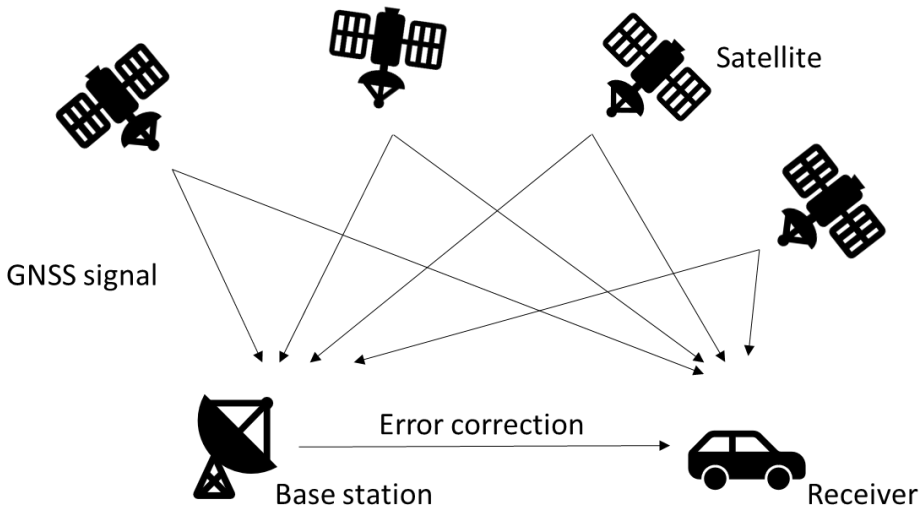


Figure 2-1 Differential GNSS

The carrier-phase equation is as follows:

$$\Phi_{(m)} = \rho + c(\delta T - \delta t) - \delta_{\text{ion}} + \delta_{\text{trop}} + \varepsilon_{\text{carrier}} + \lambda N \tag{2-2}$$

The difference with Equation (2-1) are:

$\varepsilon_{\text{carrier}}$ is the cm level multipath plus noise from the carrier (meters)

λN is the integer ambiguity multiplied by the carrier wavelength (meters)

2.1.3 Doppler Measurement

Velocity measurement includes position differential velocity measurement, carrier phase differential velocity measurement and Doppler velocity measurement. In the case of high position accuracy (RTK), Doppler velocity measurement is generally selected. This observation is not impacted by the integer ambiguities.

$$\dot{\text{P}}_{(m/s)} = \dot{\rho} + c(\dot{\delta}T - \dot{\delta}t) - \dot{\delta}_{\text{ion}} + \dot{\delta}_{\text{trop}} + \dot{\varepsilon} \tag{2-3}$$

Where:

$\dot{\text{P}}_{(m/s)}$ is the observed rate of range (m/s)

$\dot{\rho}$ is the true range rate between the satellite and the receiver (m/s)

$\dot{\delta}T$ is the receiver clock error drift (m/s)

$\dot{\delta}t$ is the satellite clock error drift (m/s)

$\dot{\delta}_{\text{ion}}$ is the ionosphere induced error drift (m/s)

$\dot{\delta}_{\text{trop}}$ is the troposphere induced error drift (m/s)

$\dot{\varepsilon}$ is the drift due to multipath and noise (m/s)

2.2 GNSS Error

Good integrated navigation results are inseparable from accurate measurement information. GNSS signal is transmitted from space to land through refraction and reflection of many media. In the process of signal propagation, refraction and reflection will occur in different media. In addition, the time synchronization between satellite and receiver will also cause positioning error. The signal error will change with the spatial distribution of the satellite, the error and the receiver itself. For Kalman filter, it is very

important to know the GNSS error at every moment accurately.

2.2.1 Ionospheric Error

Due to the influence of ionosphere, GNSS signal will produce time delay in the process of propagation, resulting in the reduction of positioning accuracy. Ionospheric error has the greatest impact on positioning accuracy, ranging from several meters to tens of meters. Ionospheric errors will also vary with time and location.

(1) Code group delay and carrier phase advance

The code group delay makes the distance measured by the code observation longer than the real distance; The carrier phase advance makes the distance measured by carrier phase observation shorter than the real distance.

(2) Doppler time delay

Doppler time delay, or ionospheric Doppler frequency shift, is caused by the temporal variation of the ionosphere. The ionospheric Doppler frequency shift depends on three factors: the diurnal variation rate of the ionosphere, the wide range irregularity and the speed of GPS Dynamic users.

(3) Amplitude flicker

The irregularity of the earth's ionosphere makes the GPS signal diffract and refract, which leads to frequent short period attenuation. This phenomenon is called amplitude flicker. It will seriously affect the tracking ability of GPS receiver and cause the signal to lose lock frequently, so it must be recaptured.

In order to remove ionospheric errors, multiple receivers are usually used, and multi frequency differential positioning method is used.

2.2.2 Troposphere Error

The electron content in troposphere is less, which has less influence on electromagnetic wave propagation. GNSS signal usually receives the influence of ground temperature, humidity, atmospheric pressure and so on when passing through the process. Compared with the current layer delay, the tropospheric delay has less influence on the positioning error, but there are more error sources and the error analysis is more complex. Tropospheric refraction is mainly related to satellite elevation. With the decrease of satellite elevation, the distance side of GNSS signal passing through troposphere increases, and the delay increases.

The main measures to weaken the influence of tropospheric refraction are as follows: first, the tropospheric model is used to correct, and its meteorological parameters can be directly measured at the observation station; The second is to introduce additional parameters to be estimated to describe the tropospheric effects, which can be obtained in data processing; The third is to use the synchronous observation method.

2.2.3 Satellite and Receiver Clock Errors

Because the receiver and satellite are not fully synchronized with the time reference set by GNSS (e.g., GPS time of GPS). Due to the characteristics of RTK, these two clock deviations can be offset. Due to the different performance of the receiver, the clock error of the receiver ranges from several meters to several thousand kilometers. If it is running in single point mode (i.e., without using differential receiver)In addition to the three-dimensional position coordinates, the receiver clock error is also calculated in the filter, and the

satellite clock estimation error is calculated by the satellite clock coefficient broadcast in the navigation message.

2.2.4 Multipath Effect

The multi-path effect is caused by the reflection of electromagnetic wave. The reflection of the signal transmitted by the line of sight results in the larger pseudo distance than the actual distance[25]. In reality, if the antenna is set up near the lake or in the urban canyon environment, the reflection of the signal is very obvious. It is difficult to model and analyze such errors. The existing method is to use the anti-multipath error antenna to reduce the antenna erection in challenging environment[26]. It is difficult to avoid the influence of multi-path effect on vehicle positioning in the city. At present, some scholars hope to calculate the multipath error by sensing the surrounding building model and combining with satellite sky map. The experimental results show that the error can be improved.

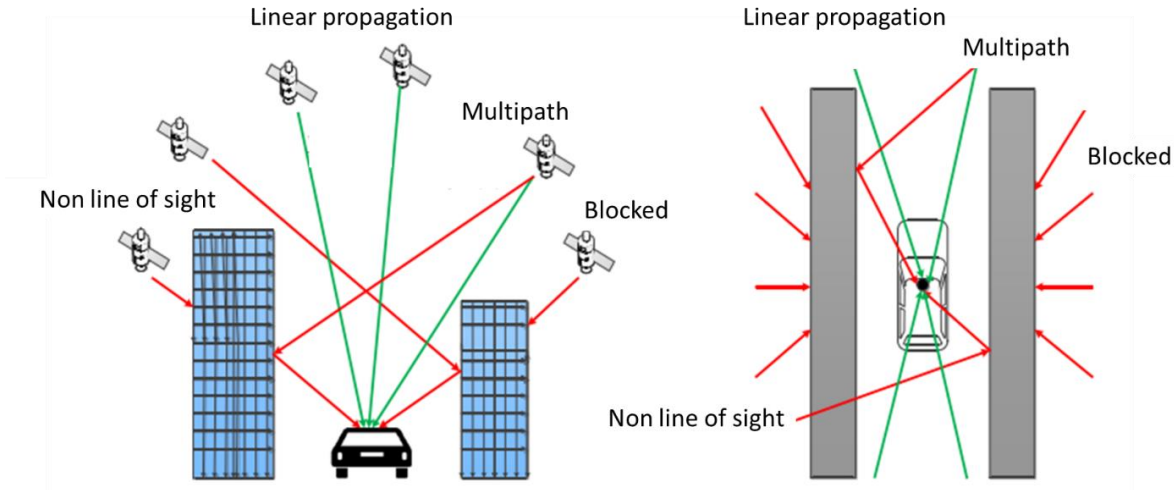


Figure 2-2 GNSS error sources

2.3 Coordinate Frames Used in Inertial Navigation

GNSS / INS needs to describe the motion state of the object. In GNSS positioning, the center of the earth is taken as the origin of the coordinate axis to calculate the position of the satellite and the receiver. IMU is a kind of inertial equipment. With the Earth rotating, it cannot know the relative position to the earth's center by itself, but it can output the attitude change and velocity change in a period of time. Usually, the static coordinate system is selected as the navigation coordinate system, and the dynamic coordinate system of IMU needs to be converted to the static coordinate system through the rotation matrix for integrated navigation.

2.3.1 Definition of Coordinate System

(1) Earth-centered, earth-fixed (ECEF) coordinate system

Earth-centered, earth-fixed coordinate system includes geocentric space rectangular coordinate system and geocentric geodetic coordinate system. As shown in Figure 2-4, the origin of geocentric space rectangular coordinate system Oe is the earth's mass center. The $OeZe$ axis points to the north pole of the earth, the $OeXe$ axis points to the intersection of the reference meridian plane (Greenwich meridian plane) and the earth's equator, and the $OeYe$ axis is perpendicular to the $XeOeZe$ plane, thus forming the right-

handed rectangular coordinate system. A datum ellipsoid is defined in geocentric geodetic coordinate system. The center of the datum ellipsoid coincides with the earth's mass center, and the minor axis of the datum ellipsoid coincides with the earth's rotation axis. The geodetic latitude is the angle between the normal of the ellipsoid passing through a point s on the ground and the equatorial plane of the ellipsoid. The geodetic longitude λ is the angle between the ellipsoidal meridian plane passing through a point s on the ground and the Greenwich meridian plane, and the geodetic height h is the distance from the point s along the normal line of the ellipsoid to the datum ellipsoid.

For MEME sensors, the noise is too huge to detect the earth rotation, sometimes the program doesn't consider the earth rotation. Set the coordinate system as an inertial system, it was called Earth-centered inertial (ECI)[27].

Table 2-1 Difference in ECI and ECEF

Coordination	Difference
ECI(Earth-centered inertial)	inertial, not accelerated, fixed w.r.t stars; useful to describe motion of celestial bodies and spacecraft.
ECEF (earth-centered, earth-fixed)	not inertial, accelerated, rotating w.r.t stars; useful to describe motion of objects on Earth surface.

(2) Local tangent plane coordinate system

The origin of the local tangent plane coordinate system is located at a point on the earth surface near the vehicle. OtXt point to the East, OtYt point to the North and OeZe point to the Up. It is a right-handed rectangular coordinate system composed of the axis pointing to the East, the axis pointing to the north and the axis pointing to the sky. When the vehicle moves in the local range of the earth plane, this coordinate system is often used as the reference datum, which is convenient and intuitive.

(3) Car body coordinate system

The car body coordinate system is fixedly connected with the vehicle. The origin Ob is at the center of gravity of the vehicle, the ObXb axis points to the front of the vehicle along the longitudinal axis of the vehicle, the ObYb axis points to the left side of the vehicle along the transverse axis of the vehicle, and the ObZb axis is perpendicular to the vehicle body upward, thus forming a right-handed rectangular coordinate system.

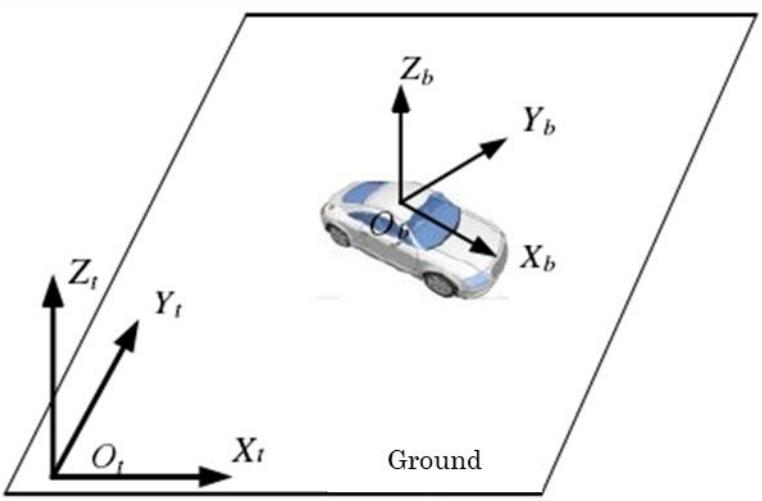


Figure 2-3 Local tangent plane coordinate and body coordinate

(4) Station center coordinate system

The origin of the station center coordinate system is usually the location of the user's receiver. The X, Y, and Z axes refer to the points, respectively. East, North, and sky. In this paper, the station center coordinate system is used to calculate the observation vector and elevation angle of the satellite relative to the vehicle, and azimuth. This figure shows the relationship between the station center coordinate system and the ECEF coordinate system.

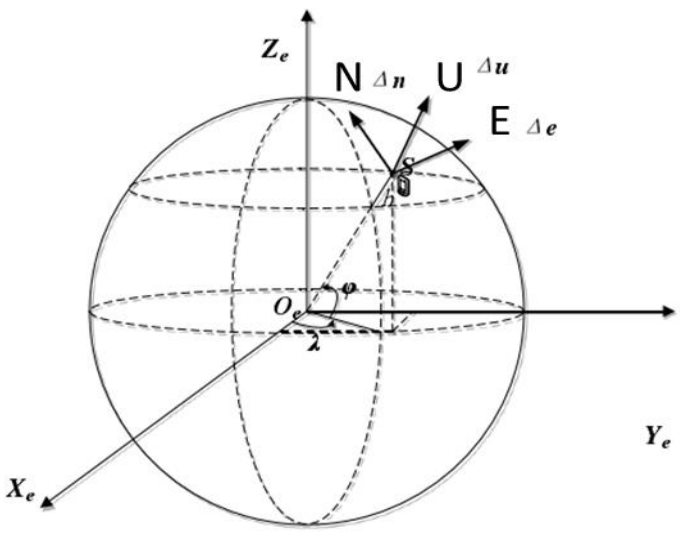


Figure 2-4 Station center coordinate and ECEF coordinate system

2.3.2 Transformation Relation of Coordinate System

(1) Geocentric rectangular coordinate system and geocentric geodetic coordinate system

The ECEF coordinates (x, y, z) and geocentric geodetic coordinates (φ, λ, h) can exchange like this:

$$\begin{cases} x = (R_N + h)\cos\ L_b \cos\ \lambda_b \\ y = (R_N + h)\cos\ L_b \sin\ \lambda_b \\ z = [R_N(1 - e^2) + h]\sin\ \lambda_b \end{cases} \tag{2-4}$$

Where:

L_b is the latitude

λ_b is the longitude

h is the altitude

R_N is the curvature radius of the prime unitary circle of the ellipsoid

e is the first eccentricity of the ellipsoid

(2) Body coordinate and local tangent plane coordinate system

By left multiplying the rotation matrix, the transformation of vector in two projection coordinate systems can be completed. For any vector x:

$$x^\beta = C_\alpha^\beta x^\alpha \tag{2-5}$$

Where:

The superscript of X represents the projection axis

The subscript of the matrix represents the source coordinate system

The superscript represents the target coordinate system

For the three-dimensional space coordinate system, it needs three times of vector rotation to transform to the target coordinate system. In different integrated navigation application scenarios, the rotation order of Euler angle in volume coordinate system is different. Because of the gimbal lock characteristic of Euler angle, it is necessary to transform the axis with frequent numerical changes first, and the axis with low frequency needs to rotate later. For the vehicle motion model, the speed of the vehicle in the sky fixed direction can be considered as 0 all the time, and occasionally the pitch angle changes in climbing, downhill and other road sections. Therefore, this paper selects the rotation mode of "heading - pitch – roll".

Here is the equation of rotation matrix, form body frame (β) to navigation frame (α)

$$C_{\beta}^{\alpha}=\begin{bmatrix} \cos\theta_{\beta\alpha}\cos\psi_{\beta\alpha} & \cos\theta_{\beta\alpha}\sin\psi_{\beta\alpha} & -\sin\theta_{\beta\alpha} \\ \begin{pmatrix}-\cos\phi_{\beta\alpha}\sin\psi_{\beta\alpha}\\ +\sin\phi_{\beta\alpha}\sin\theta_{\beta\alpha}\cos\psi_{\beta\alpha}\end{pmatrix} & \begin{pmatrix}\cos\phi_{\beta\alpha}\cos\psi_{\beta\alpha}\\ +\sin\phi_{\beta\alpha}\sin\theta_{\beta\alpha}\sin\psi_{\beta\alpha}\end{pmatrix} & \sin\phi_{\beta\alpha}\cos\theta_{\beta\alpha} \\ \begin{pmatrix}\sin\phi_{\beta\alpha}\sin\psi_{\beta\alpha}\\ +\cos\phi_{\beta\alpha}\sin\theta_{\beta\alpha}\cos\psi_{\beta\alpha}\end{pmatrix} & \begin{pmatrix}-\sin\phi_{\beta\alpha}\cos\psi_{\beta\alpha}\\ +\cos\phi_{\beta\alpha}\sin\theta_{\beta\alpha}\sin\psi_{\beta\alpha}\end{pmatrix} & \cos\phi_{\beta\alpha}\cos\theta_{\beta\alpha} \end{bmatrix} \tag{2-5}$$

Where:

$\phi_{\beta\alpha}$ is the Roll angle

$\theta_{\beta\alpha}$ is the Pitch angle

$\psi_{\beta\alpha}$ is the Yaw angle

(3) ECEF coordinate system and local tangent plane coordinate system

when the vehicle is driving in a small range on the earth surface, the local tangent plane coordinate system is usually used as the reference for vehicle navigation and positioning. In the experimental part of this paper, the positioning results are converted into local tangent plane coordinates for comparison.

$$\begin{bmatrix}x\\y\\z\end{bmatrix}^{t}=C_{e}^{n}\left(\begin{bmatrix}x\\y\\z\end{bmatrix}^{e}-\begin{bmatrix}x_0\\y_0\\z_0\end{bmatrix}^{e}\right) \tag{2-6}$$

$$C_{e}^{n}=\begin{pmatrix}-\sin L_b\cos\lambda_b & -\sin L_b\sin\lambda_b & \cos L_b\\ -\sin\lambda_b & \cos\lambda_b & 0\\ -\cos L_b\cos\lambda_b & -\cos L_b\sin\lambda_b & -\sin L_b\end{pmatrix} \tag{2-7}$$

Where:

L_b is the latitude

λ_b is the longitude

$[x, y, z]^t$ is the target position in local tangent plane coordinate

$[x, y, z]^e$ is the position in ECEF

$[x_0, y_0, z_0]^e$ is the coordinate origin position in ECEF

C_e^n is the rotation matrix form ECEF to local tangent plane coordinate

2.4 Mechanization Equations

Inertial navigation system mainly includes navigation computer and inertial sensor. The navigation computer completes the navigation calculation function. The inertial sensor consists of accelerometer and

gyroscope. Gyroscope outputs angular velocity information and azimuth of carrier, while accelerometer outputs acceleration information of carrier. Firstly, the angular velocity information of the carrier is used to calculate the attitude matrix in real time. Through the coordinate transformation of the matrix, the acceleration of the carrier is transformed into the navigation coordinate system, and the navigation parameters (position, velocity and attitude) of the carrier are obtained by the navigation computer.

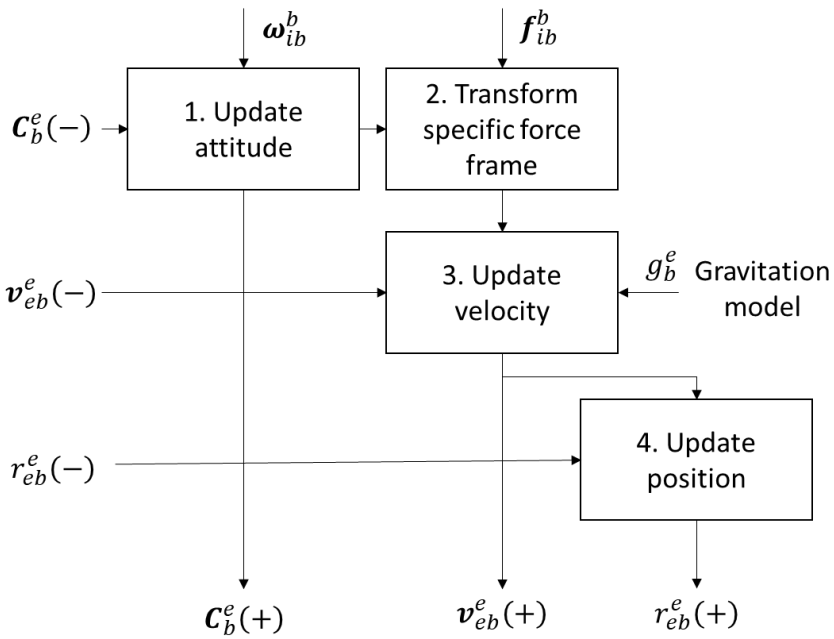


Figure 2-5 Navigation equation in ECEF

Here, shows the mechanization equation in ECEF for example:

For a more intuitive display, I will combine the formula and MATLAB code to show.

The source code is from “"Principles of GNSS, Inertial, and Multi-sensor Integrated Navigation Systems," Second Edition.”, function name is “Nav_equations_ECEF.m”.

(1) Update attitude

① Earth rotation

When the attitude updating of inertial devices is converted to the ECEF coordinate system, only the projection axis system needs to be converted because the reference position of the coordinate system is the same.

The rotation matrix form ECI to ECEF is:

$$C_e^i = \begin{pmatrix} \cos\ \omega_{ie}(t-t_0) & -\sin\ \omega_{ie}(t-t_0) & 0 \\ \sin\ \omega_{ie}(t-t_0) & \cos\ \omega_{ie}(t-t_0) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2-8}$$

Where:

ω_{ie} is the earth rotation rate (7.292115E-5 rad/s)

$t - t_0$ is the time interval between epochs (s)

```
% parameters
omega_ie = 7.292115E-5;  % Earth rotation rate (rad/s)
alpha_ie = omega_ie * tor_i;
```

```
C_Earth = [cos(alpha_ie), sin(alpha_ie), 0;...
        -sin(alpha_ie), cos(alpha_ie), 0;...
                0,          0,  1];
```

② Attitude increasement

The angle variation of gyroscope is calculated, and the rotation matrix is calculated according to the skew-symmetric matrix of vector length, next, obtain coordinate transformation matrix from the new attitude to the old using Rodrigues' formula. When the denominator part of the formula approaches zero (less than 1e-8), it is replaced by an approximate form.

$$\boldsymbol{C}_{b+}^{b-} = \boldsymbol{I}_3 + \frac{\sin|\boldsymbol{\alpha}_{ib}^b|}{|\boldsymbol{\alpha}_{ib}^b|}\left[\boldsymbol{\alpha}_{ib}^b \wedge\right] + \frac{1-\cos|\boldsymbol{\alpha}_{ib}^b|}{|\boldsymbol{\alpha}_{ib}^b|^2}\left[\boldsymbol{\alpha}_{ib}^b \wedge\right]^2 \tag{2-9}$$

Where:

$\boldsymbol{\alpha}_{ib}^b$ is the mode length of attitude change vector

$\boldsymbol{\alpha}_{ib}^b \wedge$ is the skew-symmetric matrix of $\boldsymbol{\alpha}_{ib}^b$

```
if mag_alpha>1.E-8
   C_new_old = eye(3) + sin(mag_alpha) / mag_alpha * Alpha_ib_b +...
      (1 - cos(mag_alpha)) / mag_alpha^2 * Alpha_ib_b * Alpha_ib_b;
else
   C_new_old = eye(3) + Alpha_ib_b;
end
```

Finally, Using the rotation matrix, update the attitude.

$$\boldsymbol{C}_b^b(+) \approx C_b^e(-)C_{b+}^{b-} - \Omega_{ie}^e C_b^e(-)\tau_i \tag{2-10}$$

```
C_b_e = C_Earth * old_C_b_e * C_new_old;
```

(2) Transform specific force frame

Calculate the average body to ECEF frame transformation matrix. When the denominator part of the formula approaches zero (less than 1e-8), it is replaced by an approximate form.

$$\bar{\boldsymbol{C}}_b^e = I_3 + \frac{1-\cos|\alpha_{ib}^b|}{|\alpha_{ib}^b|^2}\left[\alpha_{ib}^b \wedge\right] + \frac{1}{|\alpha_{ib}^b|^2}\left(1 - \frac{\sin|\alpha_{ib}^b|}{|\alpha_{ib}^b|}\right)\left[\alpha_{ib}^b \wedge\right]^2 \tag{2-11}$$

```
if mag_alpha>1.E-8
   ave_C_b_e = old_C_b_e * (eye(3) + (1 - cos(mag_alpha)) / mag_alpha^2 ...
      * Alpha_ib_b + (1 - sin(mag_alpha) / mag_alpha) / mag_alpha^2 ...
      * Alpha_ib_b * Alpha_ib_b) - 0.5 * Skew_symmetric([0;0;alpha_ie])...
      * old_C_b_e;
else
    ave_C_b_e = old_C_b_e - 0.5 * Skew_symmetric([0;0;alpha_ie]) *...
       old_C_b_e;
end
```

Convert the specific force to ECEF frame

$$\boldsymbol{f}_{ib}^{e} = \bar{\boldsymbol{C}}_{b}^{e}\boldsymbol{f}_{ib}^{b} \tag{2-12}$$

Where:

$\boldsymbol{f}_{ib}^{b}$ is the measurement of IMU acceleration (m/s/s)

```
C_b_e = C_Earth * old_C_b_e * C_new_old;
```

(3) Update velocity

When get the value of old speed, new attitude and new space force, we can calculate the new speed.

$$\boldsymbol{v}_{eb}^{e}(+) = \boldsymbol{v}_{eb}^{e}(-) + (\boldsymbol{g}_{b}^{e}(\boldsymbol{r}_{eb}^{e}(-)) - 2\boldsymbol{\Omega}_{ie}^{e}\boldsymbol{v}_{eb}^{e}(-))\tau_i \tag{2-13}$$

Where:

$\boldsymbol{v}_{eb}^{e}(-)$ is the last epoch velocity

$\boldsymbol{g}_{b}^{e}(\boldsymbol{r}_{eb}^{e}(-))$ is the space force

$\boldsymbol{\Omega}_{ie}^{e}$ is the skew-symmetric matrix

τ_i is the time interval between epochs (s)

```
v_eb_e = old_v_eb_e + tor_i * (f_ib_e + Gravity_ECEF(old_r_eb_e) -...
   2 * Skew_symmetric([0;0;omega_ie]) * old_v_eb_e);
```

(4) Update position

New position is from old position and new speed. In the median method, the average acceleration and average angular velocity of the two endpoints are used for integration, while the Euler method only uses one endpoint for integration. The median method has obvious advantages when the motion is relatively violent and complex. The median method is more accurate than the Eulerian method, and it is more suitable for state estimation in severe exercise.

$$r_{eb}^{e}(+)\& = r_{eb}^{e}(-) + (v_{eb}^{e}(-) + v_{eb}^{e}(+))\frac{\tau_i}{2} \tag{2-14}$$

```
r_eb_e = old_r_eb_e + (v_eb_e + old_v_eb_e) * 0.5 * tor_i;
```

2.5 IMU Error

The errors of sensors can be classified into three categories

Bias: theoretically, the output should be 0, but actually the output has a small offset. For example, if the gyroscope is absolutely stationary in the inertial frame, then theoretically the three-axis output is 0,0,0. But it's impossible. There will always be bias. And this bias is not a parameter, it will drift slowly and randomly in a certain range.

Scale factor: scale error, for example, the scale error of gyroscope is 0.02. If the gyroscope rotates at 10deg / s, when the actual output is 10deg / s * 1.02 = 10.2deg/s, there is a scale factor error.

Installation error: This error only exists in triaxial equipment, which is not present in a single axis gyro or a uniaxial accelerometer. In ideal case, the axis of coordinate system is absolutely orthogonal, but in reality, the coordinate axis of IMU is not completely orthogonal. This index will have a great impact on the violent irregular high motor movement.

2.5.1 Sensor Bias Offset

When IMU remains stationary, it still has a small output, which is zero offset. It will be affected by the power on state, temperature, internal structure of IMU, for example, gyroscope should be 0 when it is still theoretically °/ s. In fact, the output of the gyroscope stationary is a value not zero[28].

Static component of zero deviation, also known as fixed zero deviation, starts zero deviation or zero deviation repeatability, including starting zero deviation one by one and zero deviation of remaining constant value items after calibration compensation. The static component of zero deviation remains unchanged during the whole process of one start, and changes when each start is carried out. In practice, it can only be approximately constant for a period of time.

The IMU gyro bias is:

$$b_a = [b_{ax} \quad b_{ay} \quad b_{az}] \tag{2-15}$$

The IMU acceleration bias is:

$$b_g = [b_{gx} \quad b_{gy} \quad b_{gz}] \tag{2-16}$$

2.5.2 Scale Factor Error

Scale factor error is the proportional error when the electrical signal is converted into a numerical value. The error characteristic is not necessarily a constant value, it will change with the input size. If the degree of nonlinearity is relatively large, the nonlinear curve needs to be compensated to be linear before calibration.

The acceleration scale factor is:

$$K_a = \begin{bmatrix} K_{ax} & & \\ & K_{ay} & \\ & & K_{az} \end{bmatrix} \tag{2-17}$$

The gyro scale factor is:

$$K_g = \begin{bmatrix} K_{gx} & & \\ & K_{gy} & \\ & & K_{gz} \end{bmatrix} \tag{2-18}$$

2.5.3 Installation Error

The error is because of the gyro nonorthogonality, and the coordinate axes do not coincide. This error only exists in triaxial equipment, which is not present in a single axis gyro or a uniaxial accelerometer. In ideal case, the axis of coordinate system is absolutely orthogonal, but in reality, the coordinate axis of IMU is not completely orthogonal. This index will have a great impact on the violent irregular high motor movement. Usually, calculate the installation error by rotating the IMU, and calibration.

The acceleration installation error is:

$$S_a = \begin{bmatrix} 0 & S_{axy} & S_{axz} \\ S_{ayx} & 0 & S_{ayz} \\ S_{azx} & S_{azy} & 0 \end{bmatrix} \tag{2-19}$$

The gyro installation error is:

$$S_g = \begin{bmatrix} 0 & S_{gxy} & S_{gxz} \\ S_{gyx} & 0 & S_{gyz} \\ S_{gzx} & S_{gzy} & 0 \end{bmatrix} \tag{2-20}$$

2.5.4 Calibration of Internal Parameters

Summing up the above errors, we can write the IMU overall error equation, the error model is:

$$W = K_g(I + S_g)\omega + b_g \approx (K_g + S_g)\omega + b_g \tag{2-21}$$

The acceleration error model is:

$$\begin{matrix} A_x \\ [A_y] \\ A_z \end{matrix} = \begin{matrix} K_{ax} & S_{axy} & S_{axz} \\ [S_{ayx} & K_{ay} & S_{ayz}] \\ S_{azx} & S_{azy} & K_{az} \end{matrix} \begin{matrix} a_x \\ [a_y] \\ a_z \end{matrix} + \begin{matrix} b_{ax} \\ [b_{ay}] \\ b_{gz} \end{matrix} \tag{2-22}$$

The gyro error model is:

$$\begin{matrix} W_x \\ [W_y] \\ W_z \end{matrix} = \begin{matrix} K_{gx} & S_{gxy} & S_{gxz} \\ [S_{gyx} & K_{gy} & S_{gyz}] \\ S_{gzx} & S_{gzy} & K_{gz} \end{matrix} \begin{matrix} \omega_x \\ [\omega_y] \\ \omega_z \end{matrix} + \begin{matrix} b_{gx} \\ [b_{gy}] \\ b_{gz} \end{matrix} \tag{2-23}$$

Compare the true value and measurement value, the error can be calculated. The specific calculation steps will be shown in Chapter 5.

For the local gravity vector, the functions are as follow:

$$\gamma(h, L_b) = \gamma(L_b)\left(1 - \frac{2}{a}(1 + f + m - 2f\sin^2 L_b)h + \frac{3}{a^2}h^2\right) \tag{2-24}$$

$$\gamma(L_b) = \frac{a\gamma_a\cos^2 L_b + b\gamma_b\sin^2 L_b}{\sqrt{a^2\cos^2 L_b + b^2\sin^2 L_b}} \tag{2-25}$$

$$m = \frac{\omega_{ie}^2 a^2 b}{GM} \tag{2-26}$$

Where:

L_b is the latitude

h is the height

a is the length of the long half axis of the earth's ellipsoid

b is the length of the short half axis of the earth ellipsoid

f is the earth oblateness

ω_{ie} is the Angular velocity of the earth

GM is the gravitational constant of the earth

γ_a is the gravity value of the equator

γ_b is the gravity value of the pole

For the Geodetic Reference System 1980 (GRS 80), the parameter is:

$a = 6378137.0\ m$

$b = 6356752.3141\ m$

$f = 1/298.257222101$

$\omega_{ie} = 7.292115E - 5\ rad/s$

$GM = 3.986005E14\ m^3/s^2$

$\gamma_a = 9.7803267715\ m/s^2$

$\gamma_b = 9.8321863685\ m/s^2$

For the Etchujima campus of Tokyo university of marine science and technology, the normal gravity is 9.7978 m/s^2.

Following the reference[28], and did an experience in Estelle IMU. Assume that the x-axis coincides, and X_bOY_b and X_aOY_a are coplanar.

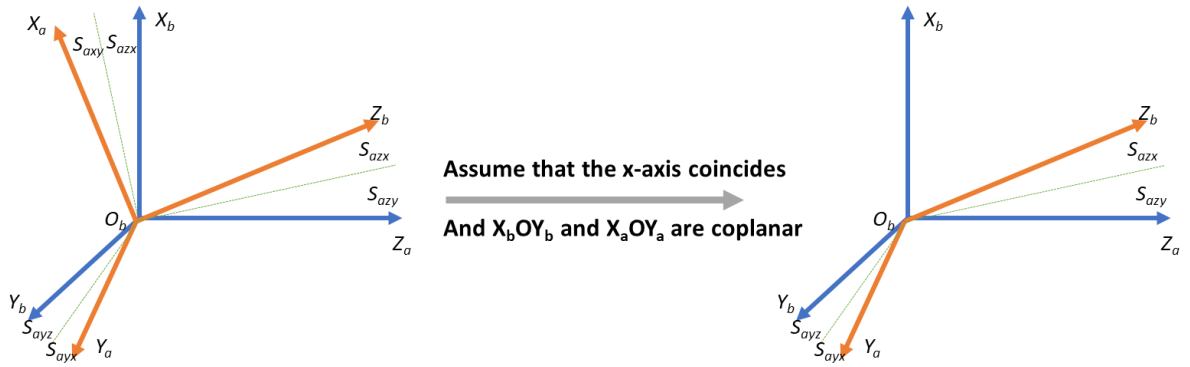


Figure 2-6 Error model of hypothetical accelerometer

From formula (2-20), the acceleration installation error will be:

$$S_a = \begin{bmatrix} 0 & 0 & 0 \\ S_{ayx} & 0 & 0 \\ S_{azx} & S_{azy} & 0 \end{bmatrix} \tag{2-27}$$

From formular (2-22), the output from IMU acceleration is:

$$a = (I + S_a)^{-1} K_a^{-1} (A - b_a) \tag{2-28}$$

The gravity vector of Tokyo University of Marice and Technology is:

$$g = [0, 0, 9.7978] \tag{2-29}$$

When the internal parameters are correct:

$$\|g\|^2 = \|a\|^2 \tag{2-30}$$

Use the 6 stable positions for accelerometer calibration[29] to make each axis coincide with the gravity vector once. After each rotation, the IMU will be static for about 500 seconds.

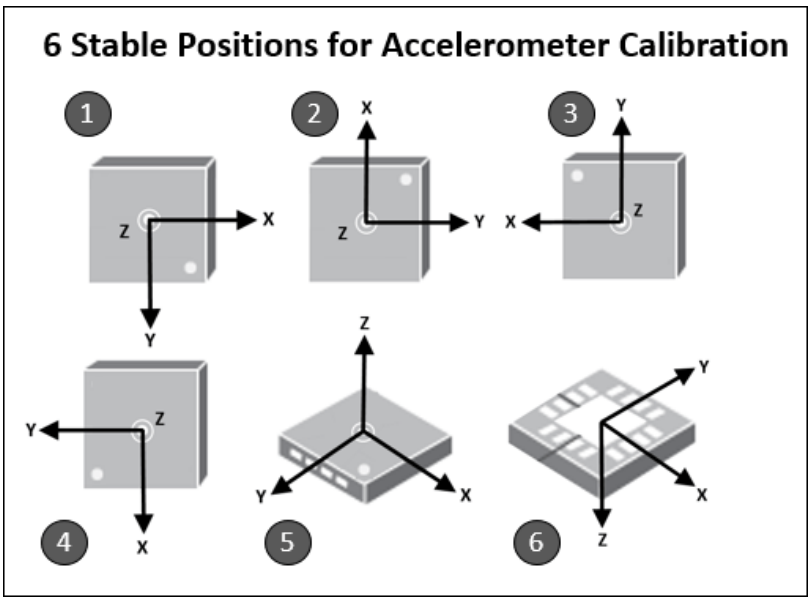


Figure 2-7 6 stable positions for accelerometer calibration

Table 2-2 The acceleration scale factor of hexahedral rotation

		X		Y		Z
1	Back	0.018153	Right	-0.00443	Down	0.998284
2	Back	0.043775	Down	-0.99834	Left	0.004385
3	Back	-0.06068	Left	0.004725	Up	-1.00496
4	Back	-0.05057	Up	0.988496	Right	-0.0064
5	Down	-1.00449	Back	-0.10214	Right	0.015054
6	Up	0.990415	Front	0.081077	Right	-0.04803

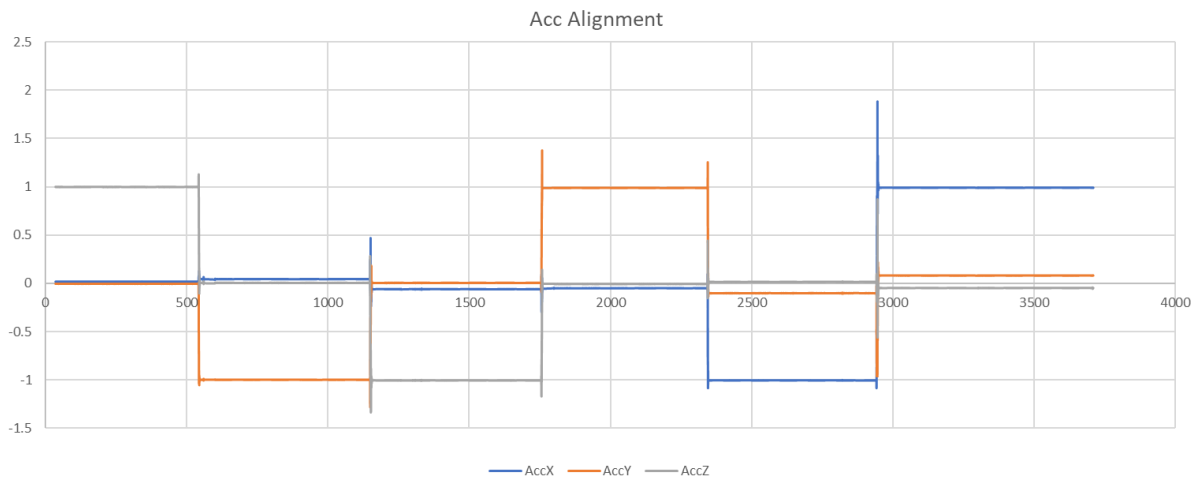


Figure 2-8 Acceleration alignment

Using the opensource tool box[30] for test, this figure shows the IMU acceleration output of without correction and after correction in 3D view. Considering the IMU has bias and there is a little error in IMU attitude during each placement, it is reasonable to say that the Estelle IMU acceleration scale factor error and installation error can be ignored.

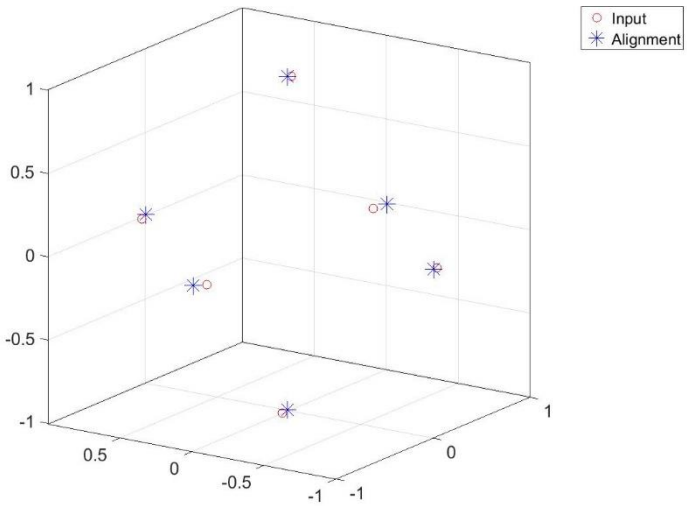


Figure 2-9 Comparison before and after calibration

2.6 Initial Alignment

Before using IMU, the initial attitude of IMU needs to be calculated. For high-precision IMU, the earth rotation can be sensed, and the attitude of IMU relative to the earth can be calculated by using the earth rotation. Therefore, for the high-precision IMU, the attitude can be aligned by static placement for a period of time. For low precision IMU, take MEMS devices as an example, the noise of MEMS IMU is very large, the earth rotation is submerged in the noise, and the relative attitude cannot be calculated by the method of static placement. In practical application, usually in vehicle operation, the heading calculated by GNSS velocity vector is used as the heading angle of IMU, and the roll angle and pitch angle of IMU are set to 0 to calculate the attitude[21, 22].

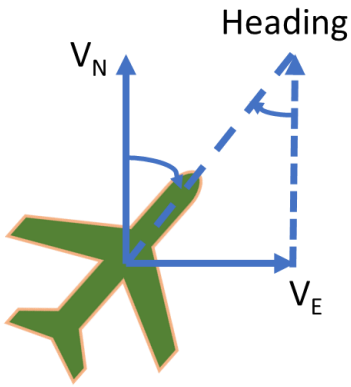


Figure 2-10 Heading from GNSS velocity

3. GNSS/INS INTEGRATION

There are several methods for GNSS and INS coupling. This paper follows Prof. Paul D. Groves' demo to introduce the LC and TC, mainly talking about classical Kalman filter. Different sensor measurement has different error model, rough error parameters can't estimate the state very well. It is necessary to analyze each error model.

3.1 Kalman Filter

The discrete-time Kalman filter algorithm comprises the following steps[1]:

1. Calculate the transition matrix, Φ_{k-1}.
2. Calculate the system noise covariance matrix, Q_{k-1}.
3. Propagate the state vector estimate from $\hat{\mathbf{x}}_{k-1}^{+}$ and $\hat{\mathbf{x}}_{k}^{-}$.
4. Propagate the error covariance matrix from $\widehat{\mathbf{P}}_{k-1}^{+}$ to $\widehat{\mathbf{P}}_{k}^{+}$.
5. Calculate the measurement matrix, H_k.
6. Calculate the measurement noise covariance matrix, R_k.
7. Calculate the Kalman gain matrix, K_k.
8. Formulate the measurement, Z_k.
9. Update the state vector estimate from $\hat{\mathbf{x}}_{k}^{-}$ to $\hat{\mathbf{x}}_{k}^{+}$.
10. Update the error covariance matrix from $\widehat{\mathbf{P}}_{k}^{-}$ to $\widehat{\mathbf{P}}_{k}^{+}$.

In Kalman filter, the state vector estimation is:

$$\hat{\boldsymbol{x}}_k^- = \boldsymbol{\Phi}_{k-1}\hat{\boldsymbol{x}}_{k-1}^+ \tag{3-1}$$

Error covariance matrix is:

$$\boldsymbol{P}_k^- = \boldsymbol{\Phi}_{k-1}\boldsymbol{P}_{k-1}^+\boldsymbol{\Phi}_{k-1}^{\mathrm{T}} + \boldsymbol{Q}_{k-1} \tag{3-2}$$

Observation matrix is:

$$\boldsymbol{h}(\boldsymbol{x}_k, t_k) = \boldsymbol{H}_k\boldsymbol{x}_k \tag{3-3}$$

Kalman gain is:

$$\boldsymbol{K}_k = \boldsymbol{P}_k^-\boldsymbol{H}_k^{\mathrm{T}}\left(\boldsymbol{H}_k\boldsymbol{P}_k^-\boldsymbol{H}_k^{\mathrm{T}} + \boldsymbol{R}_k\right)^{-1} \tag{3-4}$$

Update state vector:

$$\begin{aligned}\hat{\boldsymbol{x}}_k^+ &= \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_k^-)\\ &= \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k\delta z_k^-\end{aligned} \tag{3-5}$$

Update of error covariance matrix:

$$\boldsymbol{P}_k^+ = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{H}_k)\boldsymbol{P}_k^- \tag{3-6}$$

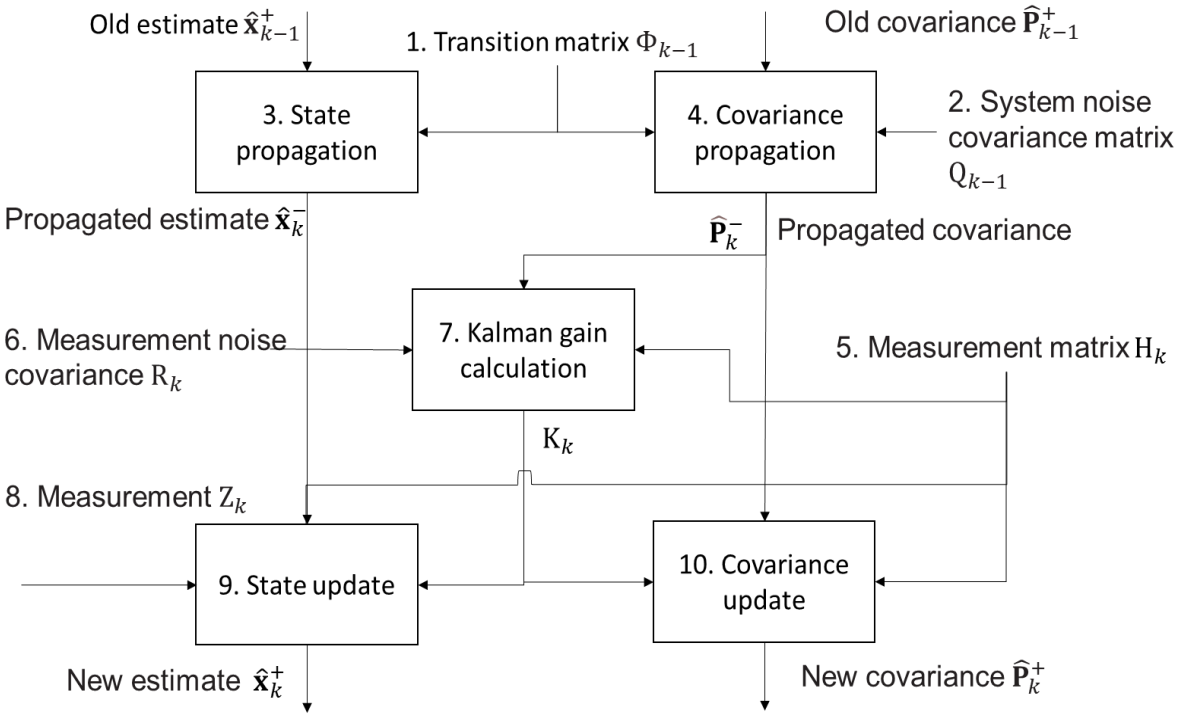


Figure 3-1 Kalman filter data flow

3.2 GNSS Error Model

This paper mainly using rtklib 2.4.3 calculates the RTK position. Calculating the GNSS velocity, GNSS compass and other GNSS information for TC by using my laboratory program. In rtklib ".pos" file, there are several information to quantifying expression noise. Such as quality of the positioning (Fix/Float/None), number of the satellites, standard deviation and so on. In general, the more visible satellites, the lower the standard deviation, the fixed solution of RTK, which means the better positioning accuracy. Sometimes, a small amount of fixed solution positioning error is greater than 0.1m. In this case, other means are needed to identify the wrong positioning results and correct the GNSS measurement error in time.

The course of the vehicle is very stable in operation, and there will be no large angle deviation in the horizontal or vertical direction. Through this characteristic, the error can be detected effectively. In general, there are two methods to identify the wrong RTK fix solution.

(1) When the deviation angle between the position and the previous trajectory is large, it can be considered that there is a large error in the current positioning. As this figure for example, heading of $\overrightarrow{AB}$ has a huge difference with heading of $\overrightarrow{BC}$, it is obviously to know that the position C is a wrong fix solution.

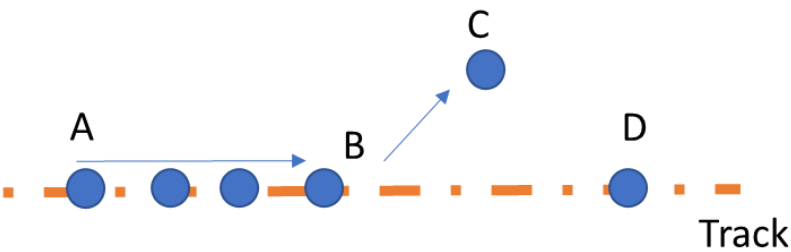


Figure 3-2 Wrong position solution in horizontal direction

(2) When the speed of the vehicle in the direction of zenith is 0, the current positioning error is considered to be large when the height changes greatly in a short time. Considering that the speed of vehicle height change is fast when climbing and downhill, the barometer can be used to help judge the current height change.

3.3 INS Error Model

INS positioning, as an inertia dead-reckoning, has huge accumulate error. It needs accurate position, velocity, attitude and bias. It is necessary to estimate them. According to the state values error model, the expression of equation of state is derived.

If the attitude, velocity and position errors estimated by Kalman filter are relative to the ECEF coordinate system and projected on the ECEF coordinate system, the state vector is as follows:

$$\boldsymbol{x}_{\mathrm{INS}}^{e} = \begin{pmatrix} \delta\boldsymbol{\psi}_{eb}^{e} \\ \delta\boldsymbol{v}_{eb}^{e} \\ \delta\boldsymbol{r}_{eb}^{e} \\ \boldsymbol{b}_{a} \\ \boldsymbol{b}_{g} \end{pmatrix} \tag{3-7}$$

Where:

The superscript e denotes the ECEF coordinate system

$\delta\boldsymbol{\psi}_{eb}^{e}$ is the attitude error derivative

$\delta\boldsymbol{v}_{eb}^{e}$ is the velocity error derivative

$\delta\boldsymbol{r}_{eb}^{e}$ is the position error derivative

$\boldsymbol{b}_{a}$ is the acceleration bias error derivative

$\boldsymbol{b}_{g}$ is the gyro bias error derivative

The acceleration bias and gyroscope bias usually set as a constant value, the derivative is zero. Here we mainly show the derivation of attitude, velocity and position.

$$E\left(\delta\dot{\boldsymbol{b}}_{a}\right) = 0 \tag{3-8}$$

$$E\left(\delta\dot{\boldsymbol{g}}_{a}\right) = 0 \tag{3-9}$$

(1) Attitude

The propagation of attitude depends on the angular rate of the earth rotation and gyro measurement, and the derivative of attitude error is：

$$\delta\boldsymbol{\psi}_{eb}^{e} \approx \widehat{\boldsymbol{C}}_{b}^{e}\left(\widehat{\omega}_{eb}^{e} - \omega_{eb}^{b}\right) \tag{3-10}$$

Where:

$\widehat{\boldsymbol{C}}_{b}^{e}$ is the estimated body to ECEF transformation matrix

$\widehat{\omega}_{eb}^{e}$ is the earth rotation angular rate

ω_{eb}^{b} is the gyro measurement

Splitting this up into gyro measurement and Earth-rate terms:

$$\begin{aligned} \delta\boldsymbol{\psi}_{eb}^{e} &\approx \widehat{\boldsymbol{C}}_{b}^{e}\delta\boldsymbol{\omega}_{ib}^{b} - \widehat{\boldsymbol{C}}_{b}^{e}\left(\widehat{\boldsymbol{C}}_{e}^{b} - \boldsymbol{C}_{e}^{b}\right)\boldsymbol{\omega}_{ie}^{e} \\ &= \widehat{\boldsymbol{C}}_{b}^{e}\delta\boldsymbol{\omega}_{ib}^{b} - \boldsymbol{\Omega}_{ie}^{e}\delta\boldsymbol{\psi}_{eb}^{e} \end{aligned} \tag{3-11}$$

the expectation of the rate of change of the attitude error is:

$$E\left(\delta\dot{\psi}_{eb}^{e}\right) \approx -\boldsymbol{\Omega}_{ie}^{e}\delta\boldsymbol{\psi}_{eb}^{e} + \widehat{\boldsymbol{C}}_{b}^{e}\boldsymbol{b}_{g} \tag{3-12}$$

Where:

$\boldsymbol{\Omega}_{ie}^{e}$ is the skew symmetric matrix of Earth rate

From continuous time to discrete time, it is as follows:

$$\delta\dot{\boldsymbol{\psi}}_{eb}^{e} = (\boldsymbol{I}_3 - \boldsymbol{\Omega}_{ie}^{e}\tau_s)\delta\boldsymbol{\psi}_{eb}^{e} + \widehat{\boldsymbol{C}}_b^e\tau_s\boldsymbol{b}_g \tag{3-13}$$

Where:

τ_s is the propagation interval

In the demo of "LC_KF_Epoch", the program is written like this:

```
omega_ie = 7.292115E-5;  % Earth rotation rate in rad/s
% Skew symmetric matrix of Earth rate
Omega_ie = Skew_symmetric([0,0,omega_ie]);
% Attitude error
Phi_matrix = eye(15);
Phi_matrix(1:3,1:3) = Phi_matrix(1:3,1:3) - Omega_ie * tor_s;
Phi_matrix(1:3,13:15) = est_C_b_e_old * tor_s;
```

(2) Velocity

The change rate of ground velocity is:

$$\dot{\boldsymbol{v}}_{eb}^{e} = \boldsymbol{f}_{ib}^{e} + \boldsymbol{g}_b^e(r_{eb}^e) - 2\boldsymbol{\Omega}_{ie}^{e}\boldsymbol{v}_{eb}^{e} \tag{3-14}$$

Time derivative of velocity error is:

$$\delta v_{eb}^{e} = \tilde{f}_{eb}^{e} - \boldsymbol{f}_{eb}^{e} + \boldsymbol{g}_b^e(\hat{r}_{eb}^e) - \boldsymbol{g}_b^e(r_{eb}^e) - 2\boldsymbol{\Omega}_{eb}^{e}(\dot{\boldsymbol{v}}_{eb}^{e} - \boldsymbol{v}_{eb}^{e}) \tag{3-15}$$

The expectation of specific force is:

$$E\left(\hat{\boldsymbol{f}}_{ib}^{e} - \boldsymbol{f}_{ib}^{e}\right) \approx -\left(\widehat{\boldsymbol{C}}_b^e, \hat{\boldsymbol{f}}_{ib}^{b}\right) \wedge \delta\boldsymbol{\psi}_{eb}^{e} + \widehat{\boldsymbol{C}}_b^e\boldsymbol{b}_a \tag{3-16}$$

The gravity error can be expressed as:

$$\boldsymbol{g}_b^e(\boldsymbol{r}_{eb}^e) - \boldsymbol{g}_b^e(\boldsymbol{r}_{eb}^e) \approx -\frac{2\hat{\gamma}_{ib}^e}{r_{eS}^e(\hat{L}_b)}\frac{\hat{r}_{eb}^{e\,\mathrm{T}}}{|\hat{r}_{eb}^e|}\delta r_{eb}^e \tag{3-17}$$

Expectation of state time derivative of velocity error is:

$$E(\delta\dot{v}_{eb}^{e}) \approx -\left(\widehat{\boldsymbol{C}}_b^e\hat{f}_{ib}^{b}\right) \wedge \delta\boldsymbol{\psi}_{eb}^{e} - 2\boldsymbol{\Omega}_{ie}^{e}\delta v_{eb}^{e} - \frac{2\hat{\gamma}_{ib}^e}{r_{eS}^e(\hat{L}_b)}\frac{\hat{r}_{eb}^{e\,\mathrm{T}}}{|\hat{r}_{eb}^e|}\delta r_{eb}^e + \widehat{\boldsymbol{C}}_b^e\boldsymbol{b}_a \tag{3-18}$$

From continuous time to discrete time, it is as follows:

$$\delta\dot{v}_{eb}^{e} = \left[-\left(\widehat{\boldsymbol{C}}_b^e\hat{f}_{ib}^{b}\right)\wedge\right]\tau_s\delta\boldsymbol{\psi}_{eb}^{e} + (I_3 - 2\boldsymbol{\Omega}_{ie}^{e}\tau_s)\delta\boldsymbol{v}_{eb}^{e} + \frac{2\hat{\gamma}_{ib}^e}{r_{eS}^e(\hat{L}_b)}\frac{\hat{r}_{eb}^{e\,\mathrm{T}}}{|\hat{r}_{eb}^e|}\tau_s\delta\boldsymbol{r}_{eb}^e + \boldsymbol{C}_b^e\tau_s\boldsymbol{b}_a \tag{3-19}$$

In the demo of "LC_KF_Epoch", the program is written like this:

```
R_0 = 6378137; %WGS84 Equatorial radius in meters
e = 0.0818191908425; %WGS84 eccentricity

% SYSTEM PROPAGATION PHASE
% Velocity error
Phi_matrix(4:6,1:3) = -tor_s * Skew_symmetric(est_C_b_e_old * meas_f_ib_b);
Phi_matrix(4:6,4:6) = Phi_matrix(4:6,4:6) - 2 * Omega_ie * tor_s;
```

```
geocentric_radius = R_0 / sqrt(1 - (e * sin(est_L_b_old))^2) *...
   sqrt(cos(est_L_b_old)^2 + (1 - e^2)^2 * sin(est_L_b_old)^2);
Phi_matrix(4:6,7:9) = -tor_s * 2 * Gravity_ECEF(est_r_eb_e_old) /...
   geocentric_radius * est_r_eb_e_old' / sqrt (est_r_eb_e_old' *...
   est_r_eb_e_old);
Phi_matrix(4:6,10:12) = est_C_b_e_old * tor_s;
```

(3) Position

The time derivative of position error is:

$$\delta \dot{r}_{eb}^{e} = \delta v_{eb}^{e} \tag{3-20}$$

In the demo of "LC_KF_Epoch", the program is written like this:

```
% Position error
Phi_matrix(7:9,4:6) = eye(3) * tor_s;
```

The above error equations write in matrix form in the order of attitude, velocity, position, acceleration bias and gyroscope bias:

$$F_{\mathrm{INS}}^{e} = \begin{pmatrix} -\boldsymbol{\Omega}_{ie}^{e} & 0_3 & 0_3 & 0_3 & \widehat{\boldsymbol{C}}_b^e \\ \boldsymbol{F}_{21}^{e} & -\boldsymbol{\Omega}_{ie}^{e} & \boldsymbol{F}_{23}^{e} & \widehat{\boldsymbol{C}}_b^e & 0_3 \\ 0_3 & I_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \end{pmatrix} \tag{3-21}$$

Where:

$$F_{21}^{e} = \left[-\left(\widehat{\boldsymbol{C}}_b^e \hat{f}_{ib}^{b}\right) \wedge\right] \tag{3-22}$$

$$F_{23}^{e} = -\frac{2\hat{\gamma}_{ib}^{e}}{r_{es}^{e}(\hat{L}_b)} \frac{\hat{r}_{eb}^{e}{}^{\mathrm{T}}}{|\hat{r}_{eb}^{e}|} \tag{3-23}$$

From continuous time to discrete time, it is as follows:

$$\boldsymbol{\Phi}_{\mathrm{INS}}^{e} \approx \begin{bmatrix} \boldsymbol{I}_3 - \boldsymbol{\Omega}_{ie}^{e}\tau_s & 0_3 & 0_3 & 0_3 & \widehat{\boldsymbol{C}}_b^e \tau_s \\ \boldsymbol{F}_{21}^{e}\tau_s & I_3 - 2\boldsymbol{\Omega}_{ie}^{e}\tau_s & \boldsymbol{F}_{23}^{e}\tau_s & \boldsymbol{C}_b^e \tau_s & 0_3 \\ 0_3 & I_3\tau_s & I_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & I_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & I_3 \end{bmatrix} \tag{3-24}$$

For TC, the estimate values include receiver clock error and clock error rate. The matrix expands from 15 by 15 to 17 by 17.

The time derivative of clock error is equal to clock error rate:

$$\frac{\partial}{\partial t}\delta\rho_c^a = \delta\dot{\rho}_c^a \tag{3-25}$$

The clock error rate is:

$$E\left(\frac{\partial}{\partial t}\delta\dot{\rho}_c^a\right) = 0 \tag{3-26}$$

The TC transfer matrix is as follows:

$$\begin{pmatrix} \boldsymbol{\Phi}_{\mathrm{INS}}^{e} \\ \delta\rho_c^a \\ \delta\dot{\rho}_c^a \end{pmatrix} \approx \begin{bmatrix} \boldsymbol{I}_3 - \boldsymbol{\Omega}_{ie}^{e}\tau_s & 0_3 & 0_3 & 0_3 & \widehat{\boldsymbol{C}}_b^e\tau_s & 0_3 & 0_3 \\ \boldsymbol{F}_{21}^{e}\tau_s & I_3 - 2\boldsymbol{\Omega}_{ie}^{e}\tau_s & \boldsymbol{F}_{23}^{e}\tau_s & \boldsymbol{C}_b^e\tau_s & 0_3 & 0_3 & 0_3 \\ 0_3 & I_3\tau_s & I_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & I_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & I_3 & 0_3 & 0_3 \\ 0 & 0 & 0 & 0 & 0 & 1 & \tau_s \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3-27}$$

3.4 GNSS/INS Loosely Coupled Methods

In GNSS / INS LC, the positioning process is separated from INS. The position, velocity and attitude measurements needed in the integrated navigation algorithm can be obtained by RTK Positioning, Doppler velocity measurement and GNSS compass. GNSS position is the position of receiver antenna, and there is a level arm between GNSS and IMU. If this distance is ignored, a large error will occur in the case of vehicle turning and other movements.

Here, using my LC MATLAB code “Main_LC.m” for example, this code is modified from Prof. Paul Groves’ demo “INS_GNSS_Demo_3”[1].

1. Initial setting

The following parameters are required:

Initial uncertainty of attitude, velocity ,position, accelerometer bias and gyro bias will be used for Kalman filter P matrix initial setting.

```
% Initial attitude uncertainty per axis (deg, converted to rad)
LC_KF_config.init_att_unc = deg2rad(5);
% Initial velocity uncertainty per axis (m/s)
LC_KF_config.init_vel_unc = 0.1;
% Initial position uncertainty per axis (m)
LC_KF_config.init_pos_unc = 0.1;
% Initial accelerometer bias uncertainty per instrument (micro-g, converted to m/s^2)
LC_KF_config.init_b_a_unc = 1000 * micro_g_to_meters_per_second_squared;
% Initial gyro bias uncertainty per instrument (deg/hour, converted to rad/sec)
LC_KF_config.init_b_g_unc = 10 * deg_to_rad / 3600;
```

For Kalman filer Q matrix, need to input the IMU bias and random walk power spectral density (PSD). Here, the parameter is for Epson-G370.

```
% Gyro noise PSD (deg^2 per hour, converted to rad^2/s)
LC_KF_config.gyro_noise_PSD = 1.878e-12;
% Accelerometer noise PSD (micro-g^2 per Hz, converted to m^2 s^-3)
LC_KF_config.accel_noise_PSD = 1e-3;

% Accelerometer bias random walk PSD (m32 s^-5)
LC_KF_config.accel_bias_PSD = 5e-7;
% Gyro bias random walk PSD (rad^2 s^-3)
```

```
LC_KF_config.gyro_bias_PSD = 5.7453e-14;
```

For Kalman filter R matrix, need to input the GNSS position and velocity noise stand deviation (SD).

```
% GNSS pos measurement noise SD (m)
LC_KF_config.pos_meas_SD =0.03;
% GNSS vel measurement noise SD (m/s)
LC_KF_config.vel_meas_SD = 1e-4;
```

Setting the interval between GNSS epochs (s), usually set as 5 Hz.

```
GNSS_config.epoch_interval = 0.2;
```

Set the start time of LC, give the IMU accurate position, velocity, attitude, and level arm from IMU to GNSS antenna. The attitude order is Roll-Pitch-Yaw (rad). The level arm direction is Fore-Right-Down, unit is meter. Here is the 2021/06/09 data setting for example.

```
%% Data for 2021/06/09
 old_time = 284042;
 old_est_r_ea_e = [-3961766.0534;   3349008.9325;   3698311.0212];
 old_est_v_ea_e=[0.000  ;0.00   ;0.00];
 attitude_ini = [0;0;135/180*pi];
 L_ba_b=[-1;0;-0.7];
```

Convert the body frame to navigation frame (ECEF), the rotation order is Yaw-Pitch-Roll. Calculate the rotation matrix, and correct the IMU position by level arm.

```
%Coordinate transformation matrix from IMU to local horizontal coordinate system
old_est_C_b_n=Euler_to_CTM(attitude_ini)';
[old_est_L_a,old_est_lambda_a,old_est_h_a,old_est_v_ea_n] =...
   pv_ECEF_to_NED(old_est_r_ea_e,old_est_v_ea_e);
[~,~,old_est_C_b_e] = NED_to_ECEF(old_est_L_a,...
   old_est_lambda_a,old_est_h_a,old_est_v_ea_n,old_est_C_b_n);
old_est_r_eb_e=old_est_r_ea_e-old_est_C_b_e*L_ba_b;
old_est_v_eb_e=old_est_v_ea_e;%The lever arm effect of velocity is ignored here
```

2. Loosely coupler of GNSS/INS

This part of the code corresponds to my program “Loosly_coupled_INS_GNSS.m”, which modified from Prof. Paul Groves’ demo “Loosely_coupled_INS_GNSS.m”.

Firstly, read the GNSS and IMU data. In this program, the IMU body frame should be changed to Fore-Right-Down. Remember the direction is specific force, so the upward vertical force is positive. The IMU acceleration unit is m/s/s, the gyro unit is rad/s. For MEMS IMU, before doing the LC, usually place the IMU in static condition for several minutes, and remove the bias.

```
%% G370 imu body frame B-L-D to F-R-D
```

```
IMUData_ = IMUData;
IMUData_(:,5:7) = IMUData_(:,5:7)./180*pi;
IMUData_(:,2:7) = IMUData_(:,2:7) - mean(IMUData_(100:3000,2:7)); %remove bias
IMUData(:,1) =  IMUData_(:,1)+0.06;
IMUData(:,2) = IMUData_(:,2);
IMUData(:,3) = IMUData_(:,3);
IMUData(:,4) = IMUData_(:,4) - 9.7978; % 9.7978 is the local gravity of Tokyo
IMUData(:,5) = IMUData_(:,5);
IMUData(:,6) = IMUData_(:,6);
IMUData(:,7) = IMUData_(:,7);
```

Form the initial setting, it is easy to calculate the initial rotation matrix from body frame to ECEF.

```
% Determine Least-squares GNSS position solution
[old_est_L_b,old_est_lambda_b,old_est_h_b,old_est_v_eb_n]
=pv_ECEF_to_NED(old_est_r_eb_e,old_est_v_eb_e);
est_L_b = old_est_L_b;
est_lambda_b=old_est_lambda_b;
% Initialize estimated attitude solution
old_est_C_b_n=Euler_to_CTM(attitude_ini)';% To transformation matrix
[~,~,old_est_C_b_e] =
NED_to_ECEF(old_est_L_b,old_est_lambda_b,old_est_h_b,old_est_v_eb_n,old_est_C_b_n);
```

For better showing the result, usually store the result of GPS Time, IMU position converted to GNSS antenna, velocity, Euler angle and position in NEU:

```
% Generate output profile record
out_profile(1,1) = old_time;
out_profile(1,2:4)=(old_est_r_eb_e+old_est_C_b_e*L_ba_b)';
out_profile(1,5:7) = old_est_v_eb_e';
out_profile(1,8:10) = CTM_to_Euler(old_est_C_b_n')';
Rotation=[   -sin(est_L_b)*cos(est_lambda_b),-sin(est_L_b)*sin(est_lambda_b),cos(est_L_b);
   -sin(est_lambda_b),cos(est_lambda_b),0;
   cos(est_L_b)*cos(est_lambda_b),cos(est_L_b)*sin(est_lambda_b),sin(est_L_b)];
NEU=Rotation*(out_profile(1,2:4)-STA.STA(1).Coor(1:3))';
out_profile(1,11:13)=NEU';
```

Initialize Kalman filter P matrix and IMU bias states:

```
P_matrix = Initialize_LC_P_matrix(LC_KF_config);
```

Now, all setting and input information is ready, start the main loop for LC.

Read the IMU time and calculate the time interval.

```
% Input data from motion profile
```

```
time=IMU(epoch,1);
% Time interval
tor_i = time - old_time;
```

Remove the bias of acceleration and gyro.

```
meas_f_ib_b = meas_f_ib_b_ - est_IMU_bias(1:3);
meas_omega_ib_b = meas_omega_ib_b_ - est_IMU_bias(4:6);
```

Do mechanization equation, get the new position, velocity and rotation matrix of INS, the function detail has introduced in chapter 2.4.

```
[est_r_eb_e,est_v_eb_e,est_C_b_e] = Nav_equations_ECEF(tor_i,...
   old_est_r_eb_e,old_est_v_eb_e,old_est_C_b_e,meas_f_ib_b,...
   meas_omega_ib_b);
```

Looking for a matching GNSS epoch, if the GNSS epoch is not exist, the program will directly output the mechanization equation result. If there are both GNSS position and GNSS velocity available, the position and velocity Kalman filter is available.

The loosely coupled GNSS/INS part is in "LC_KF_Epoch.m". The Kalman filter method has introduced in the chapter 3.1, and the calculation process is basically the same.

(1) Determine transition matrix, this part has introduced in chapter 3.3.

```
Phi_matrix = eye(15);
Phi_matrix(1:3,1:3) = Phi_matrix(1:3,1:3) - Omega_ie * tor_s;
Phi_matrix(1:3,13:15) = est_C_b_e_old * tor_s;
Phi_matrix(4:6,1:3) = -tor_s * Skew_symmetric(est_C_b_e_old * meas_f_ib_b);
Phi_matrix(4:6,4:6) = Phi_matrix(4:6,4:6) - 2 * Omega_ie * tor_s;
geocentric_radius = R_0 / sqrt(1 - (e * sin(est_L_b_old))^2) *...
   sqrt(cos(est_L_b_old)^2 + (1 - e^2)^2 * sin(est_L_b_old)^2);
Phi_matrix(4:6,7:9) = -tor_s * 2 * Gravity_ECEF(est_r_eb_e_old) /...
   geocentric_radius * est_r_eb_e_old' / sqrt (est_r_eb_e_old' *...
   est_r_eb_e_old);
Phi_matrix(4:6,10:12) = est_C_b_e_old * tor_s;
Phi_matrix(7:9,4:6) = eye(3) * tor_s;
```

(2) Determine approximate system noise covariance matrix, this matrix is the IMU bias and random walk.

```
Q_prime_matrix = zeros(15);
Q_prime_matrix(1:3,1:3) = eye(3) * LC_KF_config.gyro_noise_PSD * tor_s;
Q_prime_matrix(4:6,4:6) = eye(3) * LC_KF_config.accel_noise_PSD * tor_s;
Q_prime_matrix(10:12,10:12) = eye(3) * LC_KF_config.accel_bias_PSD * tor_s;
Q_prime_matrix(13:15,13:15) = eye(3) * LC_KF_config.gyro_bias_PSD * tor_s;
```

(3) Propagate state estimates, all states are zero due to closed-loop correction. The state values are attitude, velocity, position, acceleration bias, gyroscope bias.

```
x_est_propagated(1:15,1) = 0;
```

(4) Propagate state estimation error covariance matrix

```
P_matrix_propagated = Phi_matrix * (P_matrix_old + 0.5 * Q_prime_matrix) *...
   Phi_matrix' + 0.5 * Q_prime_matrix;
```

(5) Set up observation matrix. In order to correspond to the corresponding state variables (attitude, velocity, position, acceleration bias, gyroscope bias), the matrix is written as follows.

$$\boldsymbol{H}_{G,k}^{n} = \begin{pmatrix} 0_3 & 0_3 & -\boldsymbol{I}_3 & 0_3 & 0_3 \\ 0_3 & -\boldsymbol{I}_3 & 0_3 & 0_3 & 0_3 \end{pmatrix} \tag{3-28}$$

```
H_matrix = zeros(6,15);
H_matrix(1:3,7:9) = -eye(3);
H_matrix(4:6,4:6) = -eye(3);
```

(6) Formulate measurement innovations, noting that zero lever arm is assumed here. The measurement innovations are position and velocity.

$$\delta z_{G,k}^{e-} = \begin{pmatrix} \hat{\boldsymbol{r}}_{eaG}^{e} - \hat{\boldsymbol{r}}_{eb}^{e} - \hat{C}_b^e \boldsymbol{L}_{ba}^b \\ \hat{\boldsymbol{v}}_{eaG}^{e} - \hat{\boldsymbol{v}}_{eb}^{e} - \hat{\boldsymbol{C}}_b^e \left(\hat{\boldsymbol{\omega}}_{ib}^b \wedge \boldsymbol{L}_{ba}^b\right) + \boldsymbol{\Omega}_{ie}^e \hat{\boldsymbol{C}}_b^e \boldsymbol{L}_{ba}^b \end{pmatrix}_k \tag{3-29}$$

In the program, the INS position has converted to antenna position, so the level arm is not mentioned here.

```
delta_z(1:3,1) = GNSS_r_eb_e -est_r_eb_e_old;
delta_z(4:6,1) = GNSS_v_eb_e -est_v_eb_e_old;
```

(7) Set-up measurement noise covariance matrix assuming all components of GNSS position and velocity are independent and have equal variance. For RTK positioning noise covariance, Float solution is worse than Fix solution, so the noise covariance is larger. It is hard to know the accurate noise covariance, so it is roughly assumed that the error is increased by 50 times when RTK is float solution.

```
if FIXFlag == 1 %Fix
   R_matrix(1:3,1:3) = eye(3) * LC_KF_config.pos_meas_SD^2 * 1;
   R_matrix(1:3,4:6) = zeros(3);
   R_matrix(4:6,1:3) = zeros(3);
   R_matrix(4:6,4:6) = eye(3) * LC_KF_config.vel_meas_SD^2 * 1;
else %Float
   R_matrix(1:3,1:3) = eye(3) * LC_KF_config.pos_meas_SD^2 .* 50;
   R_matrix(1:3,4:6) = zeros(3);
   R_matrix(4:6,1:3) = zeros(3);
   R_matrix(4:6,4:6) = eye(3) * LC_KF_config.vel_meas_SD^2 * 1;
end
```

(8) Calculate Kalman gain.

```
K_matrix = P_matrix_propagated * H_matrix' * inv(H_matrix *...
P_matrix_propagated * H_matrix' + R_matrix);
```

(9) Update state estimates.

```
x_est_new = x_est_propagated + K_matrix * delta_z;
```

(10) Update state estimation error covariance

```
P_matrix_new = (eye(15) - K_matrix * H_matrix) * P_matrix_propagated;
```

(11) Close loop correction, correct attitude, velocity, position, and update the IMU bias.

```
est_C_b_e_new = (eye(3) - Skew_symmetric(x_est_new(1:3))) * est_C_b_e_old;
est_v_eb_e_new = est_v_eb_e_old - x_est_new(4:6);
est_r_eb_e_new = est_r_eb_e_old - x_est_new(7:9);
est_IMU_bias_new = est_IMU_bias_old + x_est_new(10:15);
```

After loosely coupled, the GPS Time, position, velocity, attitude in Euler, and position in ENU results can be stored.

```
out_profile(epoch,1) = time;
   out_profile(epoch,2:4)=(est_r_eb_e+est_C_b_e*L_ba_b)';%Position of antenna
   out_profile(epoch,5:7) =
(est_v_eb_e+est_C_b_e*( Skew_symmetric(meas_omega_ib_b)*L_ba_b))';%Antenna velocity(m/s)
   out_profile(epoch,8:10) = CTM_to_Euler(est_C_b_n')' .* 180/pi; %Roll-Pitch-Yaw(degree)
   Rotation=[   -sin(est_L_b)*cos(est_lambda_b),-sin(est_L_b)*sin(est_lambda_b),cos(est_L_b);
      -sin(est_lambda_b),cos(est_lambda_b),0;
      cos(est_L_b)*cos(est_lambda_b),cos(est_L_b)*sin(est_lambda_b),sin(est_L_b)];
   NEU=Rotation*(out_profile(epoch,2:4)-STA.STA(1).Coor(1:3))';
   out_profile(epoch,11:13)=[NEU(2),NEU(1),NEU(3)]'; %Position in ENU
```

Reset the old values for next loop update.

```
old_time = time;
old_est_r_eb_e = est_r_eb_e;
old_est_v_eb_e = est_v_eb_e;
old_est_C_b_e = est_C_b_e;
```

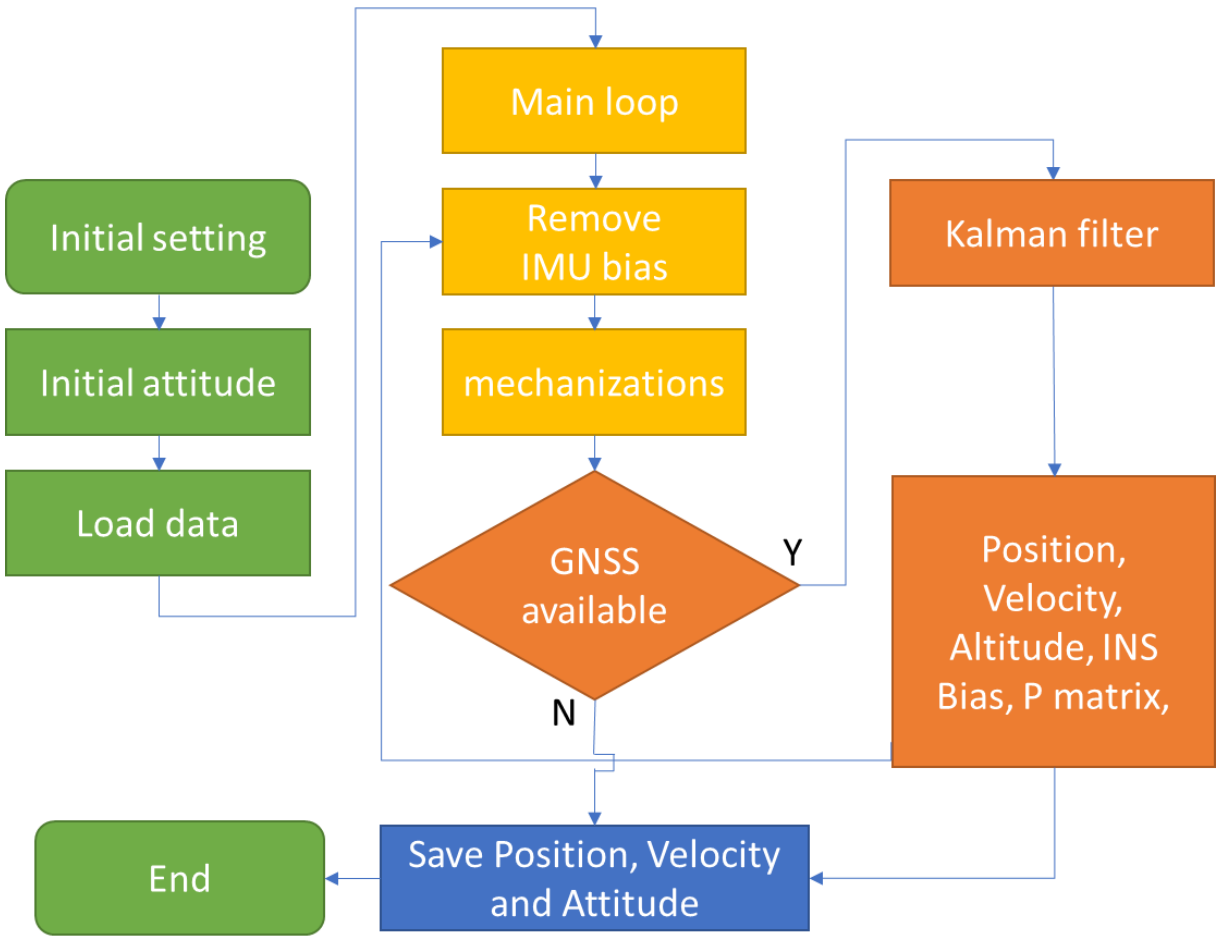


Figure 3-3 Loosely coupled flow chart

3.5 GNSS/INS Tightly Coupled Methods

The GNSS/INS Tightly Coupled Methods don't have a clear definition. Compare with LC using GNSS position and velocity as measurement update, the TC is closer to the raw data. Just as chapter 2.1 said that GNSS has several kinds of information available. We can get position form GNSS pseudorange, and get GNSS velocity from doppler frequency. Compare with LC, TC mainly changed the measurement values of Kalman filter and add the estimate value of GNSS receiver clock bias and clock bias rate.

Here, using my TC MATLAB code "Main_TC.m" for example, this code is modified from Prof. Paul Groves' demo "INS_GNSS_Demo_7"[1, 22]. The TC has some parts same as LC, for comparing easily, I marked the different parts in yellow background.

1. Initial setting

The following parameters are required:

Setting the interval between GNSS epochs (s), usually set as 5 Hz.

```
GNSS_config.epoch_interval = 0.2;
```

GNSS antenna mask angle usually set as 15 degrees, and mask carrier to noise ratio set as 30 dbhz.

```
GNSS_config.mask_angle = 15;
GNSS_config.mask_SignalStrenth = 30;
```

GNSS has many satellite systems, different have different characters, so it is necessary to define each satellite's information. Here using GPS+QZSS+GALLO+BDS for example. Too many satellites for TC will make the program very slow, here set a maximum value. For those satellites system don't want to use, set a omit list to remove them.

```
% Max Number of satellites
GNSS_config.intend_no_GNSS_meas = 30;%If the number of observations exceeds this value, start satellite screening
% Satellite to Omit
GNSS_config.omit =131:161;%Control the GNSS type involved in the solution through this configuration
% GNSS and PRN mapping list
GNSS_config.GPSPRNList=1:32;
GNSS_config.QZSSPRNList=33:40;
GNSS_config.GLOPRNList=41:80;
GNSS_config.BDSPRNList=81:130;
GNSS_config.GLONASSPRNList=131:160;
```

Initial uncertainty of attitude, velocity ,position, accelerometer bias and gyro bias will be used for Kalman filter P matrix initial setting. Compare with LC, TC added the GNSS receiver clock offset and clock drift uncertainty.

```
% Initial attitude uncertainty per axis (deg, converted to rad)
TC_KF_config.init_att_unc = deg2rad(20);
% Initial velocity uncertainty per axis (m/s)
TC_KF_config.init_vel_unc = 0.1;
% Initial position uncertainty per axis (m)
TC_KF_config.init_pos_unc = 10;
% Initial accelerometer bias uncertainty per instrument (micro-g, converted to m/s^2)
TC_KF_config.init_b_a_unc = 10000 * micro_g_to_meters_per_second_squared;
% Initial gyro bias uncertainty per instrument (deg/hour, converted to rad/sec)
TC_KF_config.init_b_g_unc = 10 * deg_to_rad / 3600;
% Initial clock offset uncertainty per axis (m)
TC_KF_config.init_clock_offset_unc = 10;
% Initial clock drift uncertainty per axis (m/s)
TC_KF_config.init_clock_drift_unc = 0.1;
```

For Kalman filer Q matrix, need to input the IMU bias and random walk power spectral density (PSD). Here, the parameter is for Epson-G370. It is same as LC.

```
% Gyro noise PSD (deg^2 per hour, converted to rad^2/s)
TC_KF_config.gyro_noise_PSD = 1.878e-12;
```

```
% Accelerometer noise PSD (micro-g^2 per Hz, converted to m^2 s^-3)
TC_KF_config.accel_noise_PSD = 1e-3;
% Accelerometer bias random walk PSD (m32 s^-5)
TC_KF_config.accel_bias_PSD = 5e-5;
% Gyro bias random walk PSD (rad^2 s^-3)
TC_KF_config.gyro_bias_PSD = 5.7453e-13;
```

For Kalman filter R matrix, need to input the GNSS receiver clock pseudo range measurement noise and pseudo range rate measurement noise stand deviation.

```
% Receiver clock phase-drift PSD (m^2/s)
TC_KF_config.pseudo_range_SD = 1;
% Pseudo-range rate measurement noise SD (m/s)
TC_KF_config.range_rate_SD = 6e-4;
```

It is common to see clock jitter in GNSS positioning. When clock jet happed, the estimate value is not accurate, so there are two choices, one is receiver lock error not be corrected, the other is corrected. Here choice the receiver lock error has been corrected.

```
;% The default value is 0,If the receiver clock jitter, the value is 2
TC_KF_config.RecClockPreprocOptions=2
```

Compare with LC, TC has a lot of measurements. It is easier to detect the gross error measurements, and has more change to choose with measurement be used. Because of this characteries, here are several kinds of Kalman filter. In this chapter, using classical Kalman filter for example.

```
TC_KF_config.KFMethod='ClassicKF';
```

The data set initial setting is same as LC, and include GNSS receiver clock error and clock error rate.

```
%% Data for 2021/06/09
 old_time = 284042;
 old_est_r_ea_e = [-3961766.0534;   3349008.9325;   3698311.0212];
 old_est_v_ea_e=[0.000  ;0.00   ;0.00];
 attitude_ini = [0;0;135/180*pi];
 L_ba_b=[-1;0;-0.7];
 est_clock=[928532.17477,376.826855];  %units are (m) and (m/s)
```

Convert the body frame to navigation frame (ECEF), the rotation order is Yaw-Pitch-Roll. Calculate the rotation matrix, and correct the IMU position by level arm. These are same as LC.

```
%Coordinate transformation matrix from IMU to local horizontal coordinate system
old_est_C_b_n=Euler_to_CTM(attitude_ini)';
[old_est_L_a,old_est_lambda_a,old_est_h_a,old_est_v_ea_n] =...
   pv_ECEF_to_NED(old_est_r_ea_e,old_est_v_ea_e);
```

```
[~,~,old_est_C_b_e] = NED_to_ECEF(old_est_L_a,...
   old_est_lambda_a,old_est_h_a,old_est_v_ea_n,old_est_C_b_n);
old_est_r_eb_e=old_est_r_ea_e-old_est_C_b_e*L_ba_b;
old_est_v_eb_e=old_est_v_ea_e;%The lever arm effect of velocity is ignored here
```

2. Read data and prepare for TC

This part of the code corresponds to my program "Tightly_coupled_INS_GNSS.m", which modified from Prof. Paul Groves' demo "Tightly_coupled_INS_GNSS.m".

Firstly, read the GNSS and IMU data. In this program, the IMU body frame should be changed to Fore-Right-Down. The IMU acceleration unit is m/s^2, the gyro unit is rad/s. For MEMS IMU, before doing the TC, usually place the IMU in static condition for several minutes, and remove the bias.

```
%% G370 imu body frame B-L-D to F-R-D
IMUData_ = IMUData;
IMUData_(:,5:7) = IMUData_(:,5:7)./180*pi;
IMUData_(:,2:7) = IMUData_(:,2:7) - mean(IMUData_(100:3000,2:7)); %remove bias
IMUData(:,1) =  IMUData_(:,1)+0.06;
IMUData(:,2) = IMUData_(:,2);
IMUData(:,3) = IMUData_(:,3);
IMUData(:,4) = IMUData_(:,4) - 9.7978;
IMUData(:,5) = IMUData_(:,5);
IMUData(:,6) = IMUData_(:,6);
IMUData(:,7) = IMUData_(:,7);
```

Form the initial setting, it is easy to calculate the initial rotation matrix from body frame to ECEF.

```
% Determine Least-squares GNSS position solution
[old_est_L_b,old_est_lambda_b,old_est_h_b,old_est_v_eb_n]
=pv_ECEF_to_NED(old_est_r_eb_e,old_est_v_eb_e);
est_L_b = old_est_L_b;
est_lambda_b=old_est_lambda_b;
% Initialize estimated attitude solution
old_est_C_b_n=Euler_to_CTM(attitude_ini)';% To transformation matrix
[~,~,old_est_C_b_e] =
NED_to_ECEF(old_est_L_b,old_est_lambda_b,old_est_h_b,old_est_v_eb_n,old_est_C_b_n);
```

For better showing the result, usually store the result of GPS Time, IMU position converted to GNSS antenna, velocity, Euler angle and position in NEU:

```
% Generate output profile record
out_profile(1,1) = old_time;
out_profile(1,2:4)=(old_est_r_eb_e+old_est_C_b_e*L_ba_b)';
out_profile(1,5:7) = old_est_v_eb_e';
```

```
out_profile(1,8:10) = CTM_to_Euler(old_est_C_b_n')';
Rotation=[   -sin(est_L_b)*cos(est_lambda_b),-sin(est_L_b)*sin(est_lambda_b),cos(est_L_b);
   -sin(est_lambda_b),cos(est_lambda_b),0;
   cos(est_L_b)*cos(est_lambda_b),cos(est_L_b)*sin(est_lambda_b),sin(est_L_b)];
NEU=Rotation*(out_profile(1,2:4)-STA.STA(1).Coor(1:3))';
out_profile(1,11:13)=NEU';
```

Initialize Kalman filter P matrix and IMU bias states:

```
P_matrix = Initialize_TC_P_matrix(TC_KF_config);
```

Now, all setting and input information is ready, start the main loop for TC.

Read the IMU time and calculate the time interval.

```
% Input data from motion profile
time=IMU(epoch,1);
% Time interval
tor_i = time - old_time;
```

Remove the bias of acceleration and gyro.

```
meas_f_ib_b = meas_f_ib_b_ - est_IMU_bias(1:3);
meas_omega_ib_b = meas_omega_ib_b_ - est_IMU_bias(4:6);
```

Do mechanization equation, get the new position, velocity and rotation matrix of INS, the function detail has introduced in chapter 2.4.

```
[est_r_eb_e,est_v_eb_e,est_C_b_e] = Nav_equations_ECEF(tor_i,...
   old_est_r_eb_e,old_est_v_eb_e,old_est_C_b_e,meas_f_ib_b,meas_omega_ib_b);
```

Looking for a matching GNSS epoch, if the GNSS epoch is not exist, the program will directly output the mechanization equation result. If there are both pseudo range and pseudo range rate available, the Kalman filter will be available.

The GNSS data each column is as follows:

```
% GNSSObs       GNSS observation array after preprocessing
%  Column 1: epoch
%  Column 2: Obsweek (week)
%  Column 3: Obssec (s)
%  Column 4: PRN
%  Column 5: corrected pseudorange (m)
%  Column 6-8: Satellite position in ECEF(m)
%  Column 9: range rate(m/s)
%  Column 10: Rate of Satellite clock  (s/s)
%  Column 11-13: Satellite velocity in ECEF(m/s)
%  Column 14: Elevation angle (deg)
```

```
% Column 15: Signal Noise Ratio (SNR)
```

The corrected pseudorange means the pseudorange that has removed the ionospheric, tropospheric delays and relativistic correction. For our dataset, the clock jitter happens in each dataset, to repair the clock jitter, the corrected pseudorange also considered the clock error.

The GNSS data was calculated by our laboratory program, but not open source now, so here, using my signal point positioning program to explain. In the file named "Main_PP.m" and "Cal_Sat_Pos.m".

As for the satellite position in ECEF

1. the time of signal transmission relative to the ephemeris reference time

$$t_k = t_{st,a}^{s} - t_{oe} \tag{3-30}$$

Where:

$t_{st,a}^{s}$ is the time when the satellite transmits the signal

t_{oe} is the reference time

```
tk=tkr-t1;
```

2. The mean anomaly is

$$M = M_0 + (\bar{\omega}_{is} + \Delta n)\Delta t \tag{3-31}$$

Where:

M_0 is the angle of the reference point

$\bar{\omega}_{is}$ is the satellite angular velocity

Δn is the correction value of average angular velocity

```
Mk=tempNav.M0+n*tk;   % (rad/s)
```

3. The true anomaly is obtained using Kepler's equation:

$$E_k = M + e_0 \sin\ E_{k-1} \tag{3-32}$$

Where:

e_0 is eccentricity of elliptical orbit

k is the iteration

```
while dEk>1e-9
    Ek=Mk+e*sin(Ek0);
    dEk=abs(Ek-Ek0);
    Ek0=Ek;
end
```

4. The true anomaly is then obtained from the eccentric anomaly using

$$\begin{aligned} v &= \arctan_2\,(\sin\ v, \cos\ v) \\ &= \arctan_2\left[\left(\frac{\sqrt{1-e_o^2}\sin\ E}{1-e_o\cos\ E}\right), \left(\frac{\cos\ E-e_o}{1-e_o\cos\ E}\right)\right] \end{aligned} \tag{3-33}$$

```
vk=2*atan(tan(Ek/2)*sqrt((1+e)/(1-e)));  % (rad)
if(vk<0)
    vk=vk+2*pi;
```

```
end
```

5. The position in an orbital coordinate frame may be expressed in polar coordinates comprising the radius and argument of latitude.

$$\Phi = \omega + \nu \tag{3-34}$$

Where:

ω is the perigee angle

```
PHI_k=vk+tempNav.omega;
```

6. The orbital radius varies as a function of the eccentric anomaly, while harmonic perturbations are applied to both terms, giving

$$u_{os}^{o} = \Phi + C_{us}\sin 2\Phi + C_{uc}\cos 2\Phi \tag{3-35}$$

$$r_{os}^{o} = a(1 - e_o\cos\ E) + C_{rs}\sin\ 2\Phi + C_{rc}\cos\ 2\Phi \tag{3-36}$$

$$i_{os}^{o} = i_0 + i \cdot t_k + \delta i_k \tag{3-37}$$

Where:

C_{us} is the angle pitch cosine harmonic correction amplitude of ascending node

C_{uc} is the angle pitch sine harmonic correction amplitude of ascending node

a is the long axis of satellite orbit

C_{rs} is the orbit radius cosine harmonic correction amplitude

C_{rc} is the orbit radius cosine harmonic correction amplitude

i_0 is the orbital inclination at reference time

i is the rate of change of track inclination

```
duk=tempNav.cus*sin(2*PHI_k)+tempNav.cuc*cos(2*PHI_k);  % Argument of Lat correction
drk=tempNav.crs*sin(2*PHI_k)+tempNav.crc*cos(2*PHI_k);  % Radius correction
dik=tempNav.cis*sin(2*PHI_k)+tempNav.cic*cos(2*PHI_k);  % Inclination correction
uk=PHI_k+duk;                % Corr. arg of lat
r=A*(1-e*cos(Ek))+drk;       % Corrected radius
ik=tempNav.i0+dik+tempNav.idot*tk;  % Corrected inclination
```

7. The satellite position in an orbital frame is:

$$x_{os}^{o} = r_{os}^{o}\cos\ u_{os}^{o}, y_{os}^{o} = r_{os}^{o}\sin\ u_{os}^{o}, z_{os}^{o} = 0 \tag{3-38}$$

```
x=r*cos(uk);
y=r*sin(uk);
```

8. Right ascension of ascending node is

$$\Omega = \Omega_0 + (\dot{\Omega} - \dot{\Omega}_e)t_k + \dot{\Omega}_e t_{oe} \tag{3-39}$$

Where:

Ω_0 is the right ascension of ascending node at reference time

$\dot{\Omega}$ is the change rate of right ascension to time

$\dot{\Omega}_e$ is the earth's rotation rate

```
Omega=tempNav.Omega0+(tempNav.Omegadot-omgedote)*tk-omgedote*tempNav.toe;
```

9. Satellite position in ECEF is

$$r_{es}^{e} = \begin{pmatrix} x_{os}^{o}\cos\ \Omega - y_{os}^{0}\cos\ i\sin\ \Omega \\ x_{os}^{o}\sin\ \Omega + y_{os}^{0}\cos\ i\cos\ \Omega \\ y_{os}^{o}\sin\ i_{os}^{o} \end{pmatrix} \tag{3-40}$$

```
% ECEF coordinates
Xs=x*cos(Omega)-y*cos(ik)*sin(Omega);
Ys=x*sin(Omega)+y*cos(ik)*cos(Omega);
Zs=y*sin(ik);
```

As for the range rate:

The actual pseudo range rate (satellite speed minus receiver speed) subtracts the satellite clock drift.

$$\dot{\rho}^{m} = -\frac{D^{m}c}{L_1} \tag{3-41}$$

Where:

D^{m} is the satellite doppler measurement

L_1 is the GNSS L1 frequency

```
rate = -1.0 * ObsData(i).svObs(j).measurements(D1C) * c / F1;
LV_ = [LV_; rate + ( AV_(end,1) .* (Xsvel-ApproCoorV_(1)) + AV_(end,2) .* (Ysvel-ApproCoorV_(2))
+ AV_(end,3) .* (Zsvel-ApproCoorV_(3)))];
```

As for the range rate:

The satellite clock calibration parameters were called af1 and af2. The af1 plus af2 of the navigation file is multiplied by the difference between the observation time and the ephemeris reference time.

```
rate_clock = tempNav.af1 + tempNav.af2 * t;
```

As for the satellite velocity:

1. Derivation of near point angle to time

$$\dot{E} = \frac{\bar{\omega}_{is}+\Delta_n}{1-e_o\cos\ E} \tag{3-42}$$

```
ek=mk/(1-tempNav.ecc*cos(Ek));
```

2. Derivation of angular distance of ascending intersection

$$\dot{\Phi} = \frac{(1+e_s\cos\ v_k)\dot{E}_k\sin\ E_k}{(1-e_s\cos\ E_k)\sin\ v_k} = \frac{\sqrt{1-e_s^2}\dot{E}_k}{1-e_s\cos\ E_k} \tag{3-43}$$

```
Wk=sqrt(1-tempNav.ecc^2)*ek/(1-tempNav.ecc*cos(Ek));
```

3. Derivation of perturbation correction term

$$\dot{u_{os}^{o}} = 2\dot{\Phi}(C_{us}\sin 2\Phi - C_{uc}\cos 2\Phi) \tag{3-44}$$

$$\dot{r_{os}^{o}} = 2\dot{\Phi}(C_{rs}\sin 2\Phi - C_{rc}\cos 2\Phi) \tag{3-45}$$

$$\dot{\iota_{os}^{o}} = 2\dot{\Phi}(C_{is}\sin 2\Phi - C_{ic}\cos 2\Phi) \tag{3-46}$$

```
Uuk=2*Wk*(tempNav.cus*cos(2*PHI_k)-tempNav.cuc*sin(2*PHI_k));
  Rrk=2*Wk*(tempNav.crs*cos(2*PHI_k)-tempNav.crc*sin(2*PHI_k));
  Iik=2*Wk*(tempNav.cis*cos(2*PHI_k)-tempNav.cic*sin(2*PHI_k));
```

4. Derivation of right ascension of ascending node angle distance, satellite vector diameter length, orbit inclination and ascending node

$$\dot{u}_k = \dot{\Phi}_k + \delta\dot{u}_k \tag{3-47}$$

$$\dot{r}_k = a_s e_s \dot{E}_k \sin\ E_k + \delta\dot{r}_k \tag{3-48}$$

$$i_k = i + \delta i_k \tag{3-49}$$

$$\dot{\Omega}_k = \dot{\Omega} - \dot{\Omega}_e \tag{3-50}$$

```
Uk=Wk+Uuk;
  Rk=(tempNav.roota^2)*tempNav.ecc*ek*sin(Ek)+Rrk;
  Ik=tempNav.idot+Iik;
  WK=tempNav.Omegadot-7.2921151467e-5;
```

5. Derivation of satellite position in orbit plane at signal transmitting time

$$\dot{x}_{os}^{o} = \dot{r}_{os}^{\circ}\cos\ u_{os}^{\circ} - r_{os}^{o}\dot{u}_{os}^{o}\sin\ u_{os}^{o} \tag{3-51}$$

$$\dot{y}_{os}^{o} = \dot{r}_{os}^{o}\sin\ u_{os}^{o} + r_{os}^{o}\dot{i}_{os}^{o}\cos\ u_{os}^{o} \tag{3-52}$$

```
  Xxk=Rk*cos(uk)-r*Uk*sin(uk);
  Yyk=Rk*sin(uk)+r*Uk*cos(uk);
```

6. The satellite velocity in ECEF is

$$\mathbf{v}_{es}^{e} = \begin{pmatrix} \dot{x}_{os}^{\infty}\cos\ \Omega - j_{os}^{o}\cos\ i\sin\ \Omega + i y_{os}^{o}\sin\ i\sin\ \Omega \\ \dot{x}_{os}^{o}\sin\ \Omega + \dot{y}_{os}^{o}\cos\ i\cos\ \Omega - i y_{os}^{o}\sin\ i\cos\ \Omega \\ \dot{y}_{cs}^{o}\sin\ i + i y_{os}^{o}\cos\ i \end{pmatrix} + \left(\omega_{ie} - \dot{\Omega}_d\right)\begin{pmatrix} x_{os}^{o}\sin\ \Omega + y_{os}^{o}\cos\ i\cos\ \Omega \\ -x_{os}^{o}\cos\ \Omega + y_{os}^{o}\cos\ i\sin\ \Omega \\ 0 \end{pmatrix} \tag{3-53}$$

```
Xsvel=-Ys*WK-(Yyk*cos(ik)-Zs*Ik)*sin(Omega)+Xxk*cos(Omega);
  Ysvel=Xs*WK+(Yyk*cos(ik)-Zs*Ik)*cos(Omega)+Xxk*sin(Omega);
  Zsvel=Yyk*sin(ik)+y*Ik*cos(ik);
```

As for the satellite elevation angle

$$\theta_{nu}^{as} = -\arcsin\left(u_{as,D}^{n}\right) \tag{3-54}$$

Where:

$u_{as,D}^{n}$ is the down vector in sight

```
ele=asind(Sat_xyz(3)/sqrt(sum(Sat_xyz.^2)));
```

As for the SNR, it can be obtained easily from observation file S1C.

```
ObsData(i).svObs(j).measurements(S1C)
```

In the program, this line means find the GNSS information in the same time and remove those satellites don't want to use.

```
GNSSObs(:,3) = round(GNSSObs(:,3),2);%Take two decimal places
index_GNSS=find(round(GNSSObs(:,3),1)==round(time,1)&~ismember(GNSSObs(:,4),GNSS_config.omit));
```

If the number of the available satellite is less than 1, the KF measurement is not enough, output the mechanization equation result.

```
if no_GNSS_meas < 1
   disp(['Time' num2str(time) 'GNSS Missing observation'])
else
```

There is a system time difference between GPS and BDS, if the GNSS raw data didn't consider the time difference, it necessary to do compensate here.

```
index_BDS=ismember(GNSSObs(index_GNSS,4),GNSS_config.BDSPRNList);
if ~isempty(index_BDS)
     GNSSObs(index_GNSS(index_BDS),5)=GNSSObs(index_GNSS(index_BDS),5)- ...
GNSS_config.ISBBDS_GPS;
end
```

3. Remove the satellites exceeded threshold

Next step is to remove those satellites that don't meet the threshold setting. This part of the code corresponds to my program "RemoveOutlierGNSSObs.m".

Calculate the estimated pseudo range, subtracting the distance from satellite to receiver antenna by pseudo range, as save the value to "RecClockPre".

```
RecClockPre=median(GNSSObs(:,5)-...
   sqrt(dot(GNSSObs(:,6:8)'-RecPosi*ones(1,no_GNSS_meas),GNSSObs(:,6:8)'-
RecPosi*ones(1,no_GNSS_meas)))');
```

The doppler frequency is as follows:

$$D^m = \frac{[(\mathbf{v}^m - \mathbf{v}) \cdot \mathbf{1}^m] L_1}{c} \tag{3-55}$$

Where:

$\mathbf{v}^m$ is the velocity of the mth satellite in the ECEF

$\mathbf{v}$ is the receiver velocity in the ECEF

L_1 is the satellite's transmission frequency

c is the speed of light

$\mathbf{1}^m$ is line of sight unit vector from the mth satellite to the GNSS receiver, which can be expressed as:

$$\mathbf{1}^m = \frac{[(x-x^m),(y-y^m),(z-z^m)]^T}{\sqrt{(x-x^m)^2+(y-y^m)^2+(z-z^m)^2}} = [1_x^m \quad 1_y^m \quad 1_z^m]^T \tag{3-56}$$

Where:

$[x, y, z]$ is the GNSS receiver position in ECEF

$[x^m, y^m, z^m]$ is the mth satellite position in ECEF

The pseudo range rate is:

$$\dot{\rho}^m = -\frac{D^m c}{L_1} = -[(\mathbf{v}^m - \mathbf{v}) \cdot \mathbf{1}^m] \tag{3-57}$$

The following program is as follows, and the result is saved in "RecClockRatePre".

```
u_as_e_T = zeros(no_GNSS_meas,3);
RecClockRateAll =zeros(no_GNSS_meas,1);
for i=1:no_GNSS_meas
   delta_r =  GNSSObs(i,6:8)' - RecPosi;
   range = sqrt(delta_r' * delta_r);
   u_as_e_T(i,1:3) = delta_r' / range;
   RecClockRateAll(i,1)= GNSSObs(i,9)  -u_as_e_T(i,1:3) * (GNSSObs(i,11:13)'- RecVelo);
end
RecClockRatePre=median(RecClockRateAll);
```

The bias of pseudo range and pseudo range rate can be obtained by subtracting the measured values of each satellite from the intermediate value.

```
RecClockBias=GNSSObs(:,5)-...
    sqrt(dot(GNSSObs(:,6:8)'-RecPosi*ones(1,no_GNSS_meas),GNSSObs(:,6:8)'-
RecPosi*ones(1,no_GNSS_meas)))'-RecClockPre;
RecClockRateBias= RecClockRateAll-RecClockRatePre;
```

Here are two ways to calculate the innovation.

The first method is if the GNSS pseudo range doesn't include the receiver clock error and clock error rate, using previous clock error and clock error rate to compare with the measurement value. If the innovation exceeds the innovation threshold, those satellites will not be used for TC.

The second method is, if the GNSS data include the receiver clock error and clock error rate, the receiver clock bias should be a small value. If the bias larger than the threshold, the corresponding satellites will be removed.

```
if RecClockPreprocOptions==0
   RecClockUpdate=est_clock_previous(1)+est_clock_previous(2)*tor_s;
   Innovation=[GNSSObs(:,5)-...
      sqrt(dot(GNSSObs(:,6:8)'-RecPosi*ones(1,no_GNSS_meas),GNSSObs(:,6:8)'-
RecPosi*ones(1,no_GNSS_meas)))'-RecClockUpdate*ones(no_GNSS_meas,1);...
      RecClockRateAll-est_clock_previous(2)*ones(no_GNSS_meas,1)];
   index_elimi=abs (Innovation(1:no_GNSS_meas))>InnovationThreshold;
else
   Innovation=[RecClockBias;RecClockRateBias];
   index_elimi=abs (RecClockBias)>RecClockBiasThreshold;
end
```

Finally, remove those satellites that don't meet the requirements of satellite mask angle and Signal Noise Ratio (SNR), and return the list of removed satellites.

```
index_lowele=GNSSObs(:,14)<GNSS_config.mask_angle;
index_lowSignalStrenth=GNSSObs(:,15)<GNSS_config.mask_SignalStrenth;
index_elimi=index_lowele|index_lowSignalStrenth|index_elimi;
```

Go back to the "Tightly_coupled_INS_GNSS.m", remove the abnormal satellites and recalute the number of satellites.

```
index_GNSS(index_elimi)=[];%Removing abnormal satellites
no_GNSS_meas=length(index_GNSS); %Recalculate the number of satellites
```

After removing the abnormal satellites, if the number of available satellites is less than one, output the mechanization equation result.

```
if no_GNSS_meas < 1
   disp(['Time' num2str(time) 'GNSSObs removed in RemoveOutlierGNSSObs section'])
else
```

If there are too many satellites available, the KF matrix will be very big and calculation will be very slow. So, it is necessary to set a threshold, and Traverse all satellite combinations to find the minimum position dilution of precision (PDOP)[31]. This part code is in my program "PickSubsetOfGNSS.m".

```
if strcmpi(method, 'Optimal')
   AllCase=nchoosek(1:SatNum,IntendedSatNum);
   PDOPAllCase=zeros(length(AllCase),1);
   for CaseIndex=1:length(AllCase)
      EliminationMask=true(SatNum,1);%reset
      EliminationMask(AllCase(CaseIndex,:))=false;
      LOS=ones(sum(~EliminationMask),1)*RecPosi'-SatPosi(~EliminationMask,1:3);
      for i=1:sum(~EliminationMask)
         LOS(i,1:3)=LOS(i,1:3)/norm(LOS(i,1:3));
      end
      PDOPAllCase(CaseIndex)=sqrt(trace(inv(LOS'*LOS)));
   end
   [PDOP,IndexMinPDOP]=min(PDOPAllCase);
   EliminationMask=true(SatNum,1);%reset
   EliminationMask(AllCase(IndexMinPDOP,:))=false;
   RunningTime=toc;
   return
else
   disp(['GNSS Star selection algorithm:' method 'Undefined'])
end
```

```
RunningTime=toc;
LOS=ones(sum(~EliminationMask),1)*RecPosi'-SatPosi(~EliminationMask,1:3);
for i=1:sum(~EliminationMask)
   LOS(i,1:3)=LOS(i,1:3)/norm(LOS(i,1:3));
end
PDOP=sqrt(trace(inv(LOS'*LOS)));
Return
```

If the GNSS clock error repaired, forced use of clock error preprocessing strategy in the whole process, else, set the pervious receiver clock error as 0.

```
if TC_KF_config.RecClockPreprocOptions==2%Forced use of clock error preprocessing strategy in
the whole process
      Clock_Reset_Flag=1;%1 indicates that the GNSS data has been repaired by clock jump, the
status is updated, and the clock difference is set to zero
      GNSS_measurements(:,1)=GNSS_measurements(:,1)-RecClockPre;
else
      RecClockPre=0;
      Clock_Reset_Flag=0;%1 indicates that the GNSS data has been repaired by clock jump, and the
clock drift of the status update clock is set to zero, the default is 0
end
```

The TC GNSS/INS part is in “TC_KF_Epoch.m”. The Kalman filter method has introduced in the chapter 3.1, and the calculation process is basically the same. Compare with LC, the estimate value is added with clock error and clock error rate, the GNSS measurements replaced form position and velocity to pseudo range and pseudo range rate, the observation matrix is defined by the number of measurements.

(1) Determine transition matrix, this part has introduced in the end of chapter 3.3. Compare with LC 15 by 15 matrix, added the clock error and clock error rate.

```
Phi_matrix = eye(17);
Phi_matrix(1:3,1:3) = Phi_matrix(1:3,1:3) - Omega_ie * tor_s;
Phi_matrix(1:3,13:15) = est_C_b_e_old * tor_s;
Phi_matrix(4:6,1:3) = -tor_s * Skew_symmetric(est_C_b_e_old * meas_f_ib_b);
Phi_matrix(4:6,4:6) = Phi_matrix(4:6,4:6) - 2 * Omega_ie * tor_s;
geocentric_radius = R_0 / sqrt(1 - (e * sin(est_L_b_old))^2) *...
   sqrt(cos(est_L_b_old)^2 + (1 - e^2)^2 * sin(est_L_b_old)^2);
Phi_matrix(4:6,7:9) = -tor_s * 2 * Gravity_ECEF(est_r_eb_e_old) /...
   geocentric_radius * est_r_eb_e_old' / sqrt (est_r_eb_e_old' *...
   est_r_eb_e_old);
Phi_matrix(4:6,10:12) = est_C_b_e_old * tor_s;
Phi_matrix(7:9,4:6) = eye(3) * tor_s;
```

```
Phi_matrix(16,17) = tor_s;
```

(2) Determine approximate system noise covariance matrix, this matrix is the IMU bias, random walk receiver clock frequency drift and receiver clock measurement noise.

```
Q_prime_matrix = zeros(17);
Q_prime_matrix(1:3,1:3) = eye(3) * TC_KF_config.gyro_noise_PSD * tor_s;
Q_prime_matrix(4:6,4:6) = eye(3) * TC_KF_config.accel_noise_PSD * tor_s;
Q_prime_matrix(10:12,10:12) = eye(3) * TC_KF_config.accel_bias_PSD * tor_s;
Q_prime_matrix(13:15,13:15) = eye(3) * TC_KF_config.gyro_bias_PSD * tor_s;
Q_prime_matrix(16,16) = TC_KF_config.clock_phase_PSD * tor_s;% unit is (m^2/s^3)
Q_prime_matrix(17,17) = TC_KF_config.clock_freq_PSD * tor_s; %unit is (m)
```

(3) Propagate state estimates, INS states are zero due to closed-loop correction. Compare with LC, TC includes receiver clock error and receiver clock error rate. If the clock error has been repaired, the estimated value should be set as zero, else using the old clock error and clock error rate.

```
x_est_propagated(1:15,1) = 0;
if Clock_Reset_Flag==1%GNSS clock skip repair identification, for the repaired clock, the residual
clock error that has not been repaired is 0
   x_est_propagated(16,1)=0;
   x_est_propagated(17,1)=0;
else
   x_est_propagated(16,1) = est_clock_old(1) + est_clock_old(2) * tor_s;
   x_est_propagated(17,1) = est_clock_old(2);
end
```

(4) Propagate state estimation error covariance matrix, this step is same as LC.

```
P_matrix_propagated = Phi_matrix * (P_matrix_old + 0.5 * Q_prime_matrix) *...
   Phi_matrix' + 0.5 * Q_prime_matrix;
```

(5) Set up observation matrix. In order to correspond to each satellite, the matrix is written as follows. For predict pseudo range using

$$\hat{\boldsymbol{\rho}}_{a,C,k}^{j-} = \sqrt{\left[\hat{\boldsymbol{r}}_{ej}^{e}\left(\tilde{t}_{st,a,k}^{j}\right) - \hat{\boldsymbol{r}}_{ea,k}^{e-}\right]^{\mathrm{T}}\left[\hat{\boldsymbol{r}}_{ej}^{e}\left(\tilde{t}_{st,a,k}^{j}\right) - \hat{r}_{ea,k}^{e-}\right]} + \delta\hat{\boldsymbol{\rho}}_{c,k}^{a-} + \delta\boldsymbol{\rho}_{ie}^{j} \tag{3-58}$$

The corresponding program is as follows:

```
delta_r =  GNSS_measurements(j,3:5)' - est_r_ea_e_old;
range = sqrt(delta_r' * delta_r);
pred_meas(j,1) = range + x_est_propagated(16);
% Predict line of sight
u_as_e_T(j,1:3) = delta_r' / range;
```

The predict pseudo range rate is:

$$\dot{r}_{as} = \boldsymbol{u}_{as}^{e^{\mathrm{T}}}\left(\boldsymbol{v}_{es}^{e}\left(t_{st,a}^{s}\right) - \boldsymbol{v}_{ea}^{e}\left(t_{sa,a}^{s}\right)\right) + \delta\dot{\rho}_{ie,a}^{s} \tag{3-59}$$

Where:

$\delta\dot{\rho}_{ie,a}^{s}$ is the Sagnac correction, it needs accurate position. In program, it is not including.

```
range_rate = u_as_e_T(j,1:3) * (GNSS_measurements(j,6:8)'-est_v_ea_e_old);
pred_meas(j,2) = range_rate + x_est_propagated(17);
```

The measurement matrix is

$$\boldsymbol{H}_{G,k}^{\gamma} \approx \begin{pmatrix} 0_{1,3} & 0_{1,3} & u_{a1}^{\gamma\mathrm{T}} & 0_{1,3} & 0_{1,3} & 1 & 0 \\ 0_{1,3} & 0_{1,3} & u_{a2}^{\gamma\mathrm{T}} & 0_{1,3} & 0_{1,3} & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0_{1,3} & 0_{1,3} & u_{am}^{\gamma\mathrm{T}} & 0_{1,3} & 0_{1,3} & 1 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0_{1,3} & u_{a1}^{\gamma\mathrm{T}} & 0_{1,3} & 0_{1,3} & 0_{1,3} & 0 & 1 \\ 0_{1,3} & u_{a2}^{\gamma\mathrm{T}} & 0_{1,3} & 0_{1,3} & 0_{1,3} & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0_{1,3} & u_{am}^{\gamma\mathrm{T}} & 0_{1,3} & 0_{1,3} & 0_{1,3} & 0 & 1 \end{pmatrix}_{x=\hat{x}_k} \quad ,\gamma \in i,e \tag{3-60}$$

The program is as follows:

```
H_matrix(1:no_meas,7:9) = u_as_e_T(1:no_meas,1:3);
H_matrix(1:no_meas,16) = ones(no_meas,1);

H_matrix((no_meas + 1):(2 * no_meas),4:6) = u_as_e_T(1:no_meas,1:3);
H_matrix((no_meas + 1):(2 * no_meas),17) = ones(no_meas,1);
```

(6) Each satellite signal measurement noise is defined by the elevation angle. Here, if the elevation angle is less than 30 degrees, the noise will also be expanded.

```
index_LowEle=GNSS_measurements(:,9)<30;
ElevationGain=ones(no_meas,1);
ElevationGain(index_LowEle)=1./(2*sind(GNSS_measurements(index_LowEle,9)));
```

The TC measurement noise covariance matrix is as follows:

```
R_matrix(1:no_meas,1:no_meas) = diag(ElevationGain)*...
   TC_KF_config.pseudo_range_SD^2;

R_matrix((no_meas + 1):2*no_meas,(no_meas + 1):2*no_meas) =...
   diag(ElevationGain.*ElevationGain) * TC_KF_config.range_rate_SD^2;
```

(7) Calculate Kalman gain. This step is same as LC.

```
K_matrix = P_matrix_propagated * H_matrix' * inv(H_matrix *...
P_matrix_propagated * H_matrix' + R_matrix);
```

(8) Formulate measurement innovations are as follows:

$$\begin{cases} \delta z_{\rho,k}^{-} = \left(\tilde{\rho}_{a,C}^{1} - \hat{\rho}_{a,C}^{1-}, \tilde{\rho}_{a,C}^{2} - \hat{\rho}_{a,C}^{2-}, \cdots \tilde{\rho}_{a,C}^{m} - \hat{\rho}_{a,C}^{m-}\right)_k \\ \delta z_{r,k}^{-} = \left(\tilde{\rho}_{a,C}^{1} - \hat{\rho}_{a,C}^{1-}, \tilde{\rho}_{a,C}^{2} - \hat{\rho}_{a,C}^{2-}, \cdots \tilde{\rho}_{a,C}^{m} - \hat{\rho}_{a,C}^{m-}\right)_k \end{cases} \tag{3-61}$$

```
delta_z(1:no_meas,1) = GNSS_measurements(1:no_meas,1) -...
   pred_meas(1:no_meas,1);
delta_z((no_meas + 1):(2 * no_meas),1) = GNSS_measurements(1:no_meas,2) -...
   pred_meas(1:no_meas,2);
```

(9) Update state estimates. Compare with LC, the method is same, but the estimate matrix, Kalman gain matrix and measurement innovation matrix is expended to 17 by 17.

```
x_est_new = x_est_propagated + K_matrix * delta_z;
```

(10) Update state estimation error covariance

```
P_matrix_new = (eye(15) - K_matrix * H_matrix) * P_matrix_propagated;
```

(11) Close loop correction, correct attitude, velocity, position, update the IMU bias, receiver clock error and receiver clock error rate.

```
est_C_b_e_new = (eye(3) - Skew_symmetric(x_est_new(1:3))) * est_C_b_e_old;
est_v_eb_e_new = est_v_eb_e_old - x_est_new(4:6);
est_r_eb_e_new = est_r_eb_e_old - x_est_new(7:9);
est_IMU_bias_new = est_IMU_bias_old + x_est_new(10:15);
est_clock_new = x_est_new(16:17)';
```

After TC, the GPS Time, position, velocity, attitude in Euler, and position in ENU results can be stored.

```
out_profile(epoch,1) = time;
   out_profile(epoch,2:4)=(est_r_eb_e+est_C_b_e*L_ba_b)';%Position of antenna
   out_profile(epoch,5:7) =
(est_v_eb_e+est_C_b_e*( Skew_symmetric(meas_omega_ib_b)*L_ba_b))';%Antenna velocity(m/s)
   out_profile(epoch,8:10) = CTM_to_Euler(est_C_b_n')' .* 180/pi; %Roll-Pitch-Yaw(degree)
   Rotation=[   -sin(est_L_b)*cos(est_lambda_b),- ...
sin(est_L_b)*sin(est_lambda_b),cos(est_L_b);
       -sin(est_lambda_b),cos(est_lambda_b),0;
       cos(est_L_b)*cos(est_lambda_b),cos(est_L_b)*sin(est_lambda_b),sin(est_L_b)];
   NEU=Rotation*(out_profile(epoch,2:4)-STA.STA(1).Coor(1:3))';
   out_profile(epoch,11:13)=[NEU(2),NEU(1),NEU(3)]'; %Position in ENU
```

Reset the old values for next loop update.

```
old_time = time;
old_est_r_eb_e = est_r_eb_e;
old_est_v_eb_e = est_v_eb_e;
```

```
old_est_C_b_e = est_C_b_e;
```

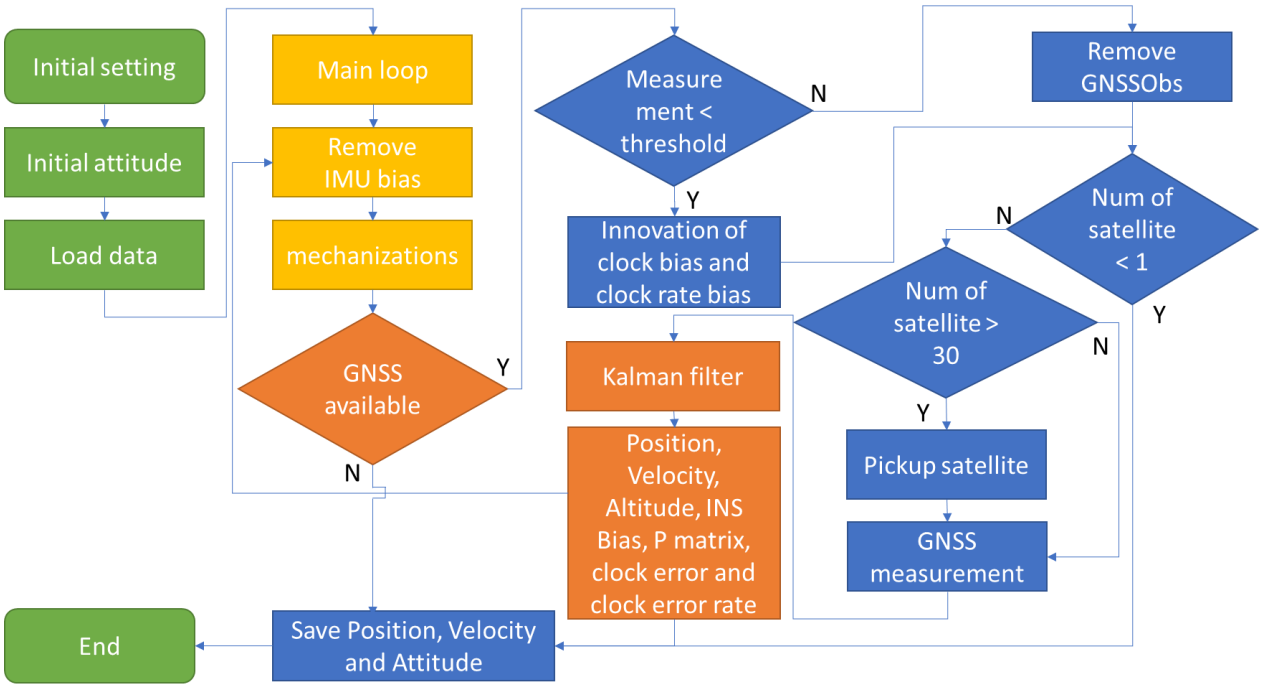


Figure 3-4 Tightly coupled flow chart

3.6 Optimization of Integrated Navigation Algorithm

In this chapter, most of the parts are follow the Prof. Paul Groves' demo. It is a good example but cannot meet the requirements of the real data set. In the real data, as chapter 2.2 said, there are so many challenge environments for GNSS. GNSS measurement with huge noise will make the positioning result unstable. To solve this problem, there are two ways. One method is to optimize the integrated navigation algorithm, the other is multi sensor fusion for integration navigation. This chapter will mainly be talking about the optimization of algorithm, and chapter 4 will mainly introduce several multi-sensor fusion methods.

In chapter 3.5, this paper not only introduces the TC but also compares the difference between LC and TC. For the similar part, the difference is marked in yellow. It is easy for the reader to understand the relationship between these two different methods. It is obvious to see that when the GNSS measurements are changed from position and velocity to pseudo-range and pseudo rang rate, the KF changed the estimate values $\boldsymbol{x}$, estimate noise covariance matrix $\boldsymbol{Q}$, measurement values δz, measurement noise covariance matrix $\boldsymbol{R}$, and observation matrix $\boldsymbol{H}$. When the measurement values changed, the KF just needs to change this matrix, and the other parts keep them as same.

3.6.1 Optimization Algorithm for GNSS/INS Loosely Coupled

As for the GNSS position and velocity, using RTK position and GNSS doppler velocity for example.

(1) Optimization in $\boldsymbol{R}$

In the output positioning file, rtklib gives us the quality of the position. "Q = 0" means the position in this epoch is none, "Q = 1" means the position is fix, "Q = 2" means the position is float. Most of the time, the fix position's noise is very small, usually the error is within 0.1 meter.

When the position is float, it means the GNSS position quality in this epoch is poor and error is huge. The rtklib also provide the number of satellites and stander deviation of each direction (x/y/z for ECEF)[32, 33]. In KF, when the RTK position is fix, the measurement noise is low, and when the quality is float, the noise is high. It is hard to know the accurate measurement noise when RTK is float, so here expand 50 times by experience.

```
if FIXFlag == 1 %RTK Fix
   R_matrix(1:3,1:3) = eye(3) * LC_KF_config.pos_meas_SD^2 * 1;
   R_matrix(1:3,4:6) = zeros(3);
   R_matrix(4:6,1:3) = zeros(3);
   R_matrix(4:6,4:6) = eye(3) * LC_KF_config.vel_meas_SD^2 * 1;
else %RTK Float
   R_matrix(1:3,1:3) = eye(3) * LC_KF_config.pos_meas_SD^2 .* 50;
   R_matrix(1:3,4:6) = zeros(3);
   R_matrix(4:6,1:3) = zeros(3);
   R_matrix(4:6,4:6) = eye(3) * LC_KF_config.vel_meas_SD^2;
end
```

(2) Optimization in δz

In the real GNSS data, it is common to see that GNSS velocity epochs are more than GNSS position epochs. If we only using position and velocity KF LC, it will waste some available measurement. For MEMS IMU, the error is very huge, so it is necessary to keep the INS error update. Change the position and velocity to only velocity, this part of program is in my program "LC_VEL_KF_Epoch"[34], the steps are as follow:

1. Change the $\boldsymbol{H}$ to only using velocity

The measurement matrix changed to 3 values, so the observation matrix should also be changed. The estimate value of velocity is in the second, so in the $\boldsymbol{H}$ matrix, the unit matrix is in the second part.

$$\boldsymbol{H}_{G,k}^{n} = (0_3 \quad -\boldsymbol{I}_3 \quad 0_3 \quad 0_3 \quad 0_3) \tag{3-62}$$

```
H_matrix = zeros(3,15);
H_matrix(1:3,4:6) = -eye(3);
```

2. Change the δz to only using velocity

Compare with position and velocity KF update, now only using velocity for update. The δz is reduced as follows:

$$\delta z_{G,k}^{e-} = \left(\hat{\boldsymbol{v}}_{eaG}^{e} - \hat{\boldsymbol{v}}_{eb}^{e} - \hat{\boldsymbol{C}}_{b}^{e}\left(\hat{\boldsymbol{\omega}}_{ib}^{b} \wedge \boldsymbol{L}_{ba}^{b}\right) + \boldsymbol{\Omega}_{ie}^{e}\hat{\boldsymbol{C}}_{b}^{e}\boldsymbol{L}_{ba}^{b}\right)_k \tag{3-63}$$

```
delta_z(1:3,1) = v_eb_e -est_v_eb_e_old;
```

Just change $\boldsymbol{H}$ and δz, the other KF steps are same as usual. In this way, we can keep velocity update and remove the INS bias. It should be noted that due to the cumulative error of only velocity update Kalman filter, this method is only suitable for vehicles passing through areas with poor GNSS signal in a short time. For some data set without GNSS Doppler velocity, the method is also available to give the idea of changing the position and velocity update to only position update.

3.6.2 Optimization Algorithm for GNSS/INS Tightly Coupled

Different with LC only has position and velocity for measurements, TC number of measurements is depending on the number of available satellites. It means the GNSS/INS coupled has more choices analyze the quality of the GNSS and does some accept and reject.

1. IAE(Innovation-based adaptive estimation) KF

Firstly, using IAE -KF as example, the MATLAB code is in the file named "IAE_KF_Epoch.m"[22, 35]. Compare with classical KF, IAE-KF redefine the weight of measurement noise covariances $\boldsymbol{R}$.

In the IAE-KF, the innovation-based floating factor γ_{ii} is:

$$\gamma_{ii} = \begin{cases} 1, \left|\tilde{v}_{k,i}\right| \le k_0 \\ \frac{\left|\tilde{v}_{k,i}\right|}{k_0} \times \left(\frac{k_1 - k_0}{k_1 - \left|\tilde{v}_{k,i}\right|}\right), k_0 < \left|\tilde{v}_{k,i}\right| \le k_1 \\ \infty, \left|v_{k,i}\right| > k_1 \end{cases} \tag{3-64}$$

```
if strcmp(TC_KF_config.KFMethod,'IAE-KF')
   IAE.WeightFunctionCategory=3;%3 means to use a three-stage weight function
   IAE.K0=1.5;%Three segment weight function standardized innovation piecewise point K0
   IAE.K1=5.0;%Three segment weight function standardized innovation piecewise point K1
end
```

Where k_0 and k_1 are two constants, usually chosen as 1.5-3.2 and 4.5-8.5, respectively $\tilde{v}_{k,i}$ is the i-th normalized measurement innovation at epoch k, which can be written as:

$$\tilde{v}_{k,i} = \frac{V_{k,i}}{\sqrt{\left(\boldsymbol{H}_k \boldsymbol{P}_{k,k-1} \boldsymbol{H}_k^T + \boldsymbol{R}_k\right)_{i,i}}} \tag{3-65}$$

Set a threshold with three sections, $|\tilde{v}_{k,i}| \in [0, k_0) \cup [k_0, k_1) \cup [k_1, +\infty)$, if the $\tilde{v}_{k,i}$ is larger than the largest threshold, redefine the measurement noise as a very huge value. If the $\tilde{v}_{k,i}$ is smaller than the minimum threshold, don't change the threshold. If the $\tilde{v}_{k,i}$ is between the largest threshold and the minimum threshold, set a value $\bar{P}_{ii}$ to calculate the gain.

$$\bar{P}_{ii} = \frac{(k_1 - k_0)^2}{k_0} * \frac{\left|\tilde{v}_{k,i}\right|}{\left(k_1 - \left|\tilde{v}_{k,i}\right|\right)^2} \tag{3-66}$$

In the TC, the $\tilde{v}_{k,i}$ is δz.

```
% 6. Calculate Standardized Innovation

StdInno=delta_z./sqrt(diag(H_matrix *P_matrix_propagated * H_matrix' + R_matrix));

% Inno record
IAE.Inno=delta_z;
IAE.StdInno=StdInno;
if IAE.WeightFunctionCategory==3
   index_outlier=find(abs(StdInno>IAE.K1));
   if ~isempty(index_outlier)
```

```
    MGain(index_outlier,index_outlier)=1e4*eye(length(index_outlier));
  end
  index_buffer=find(abs(StdInno)>IAE.K0&abs(StdInno)<=IAE.K1);
  if ~isempty(index_buffer)
    MGain(index_buffer,index_buffer)=(IAE.K1-IAE.K0)^2/IAE.K0*...
      diag(abs(StdInno(index_buffer))./((IAE.K1*ones(length(index_buffer),1)-
abs(StdInno(index_buffer)))...
      .*(IAE.K1*ones(length(index_buffer),1)-abs(StdInno(index_buffer)))));
  end
end
```

Redefine the $\boldsymbol{R}$ matrix:

$$\boldsymbol{R} = \bar{P}_{ii} * \boldsymbol{R} \tag{3-67}$$

```
R_matrix=MGain*R_matrix;
```

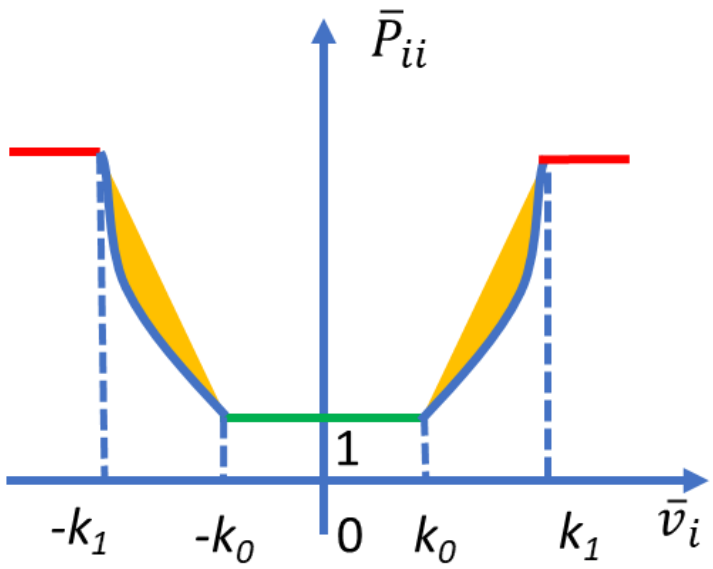


Figure 3-5 Three sections IAE-KF

2. M(M-estimator)-LS(least square) KF

To get a reliable result, an equivalent weight based on robust M-estimator is applied in the process of self-tuning Kalman filtering; robust self-tuning Kalman filtering is then proposed in this paper. The difference between IAE KF and M-LS KF in the normalized measurement innovation.

The residual is as follows:

$$\boldsymbol{V}_k = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)\boldsymbol{Z}_k \tag{3-68}$$

The unit weight variance is as follows:

$$\hat{\sigma}_0^2(\boldsymbol{k}) = \boldsymbol{Z}_k^{\mathrm{T}}\left(\boldsymbol{R}_k + \boldsymbol{H}_k \boldsymbol{P}_k^{-} \boldsymbol{H}_k^{\mathrm{T}}\right)^{-1} \boldsymbol{Z}_k / \boldsymbol{n}_k \tag{3-69}$$

Where:

$\boldsymbol{n}_k$ is the number of the satellites.

The covariance of the residual is:

$$\boldsymbol{Q}_{vv} = \boldsymbol{R}_k - \boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^{\mathrm{T}} \tag{3-70}$$

The stander residual is as follows:

$$\tilde{v}_{k,i} = \frac{\boldsymbol{v}_k}{\hat{\boldsymbol{\sigma}}_{\boldsymbol{0}}(\boldsymbol{k}) * \sqrt{\boldsymbol{Q}_{vv}}} \tag{3-71}$$

Next step is same as IAE KF, divide the stander residual to three sections, and expend the measurements noise matrix. The corresponding source code is in the file named “MLS_KF_Epoch.m”.

```
Resi=(eye(length(delta_z))-H_matrix*K_matrix)*delta_z;
UnitWeightVar=(delta_z'/(R_matrix+H_matrix*P_matrix_propagated*H_matrix'))*delta_z/(length(delta_z));
CovarianceOfResi=R_matrix-H_matrix*P_matrix_new*H_matrix';
StdResi=Resi/sqrt(UnitWeightVar)./sqrt(diag(CovarianceOfResi));
```

4. MULTI-SENSOR FUSION VEHICLE ASSIST

GNSS signal is easy to receive interference and deception, and its positioning accuracy is low in the urban canyon environment. It is necessary to prepare an alternative scheme to replace GNSS as measurement information when GNSS is not available. As chapter 3.6 introduced, the idea is to add position, velocity, and attitude measurements, keep the KF working continues and remove the gyro bias and acceleration bias.

4.1 Vehicle Motion Model

According to the kinematic model of the vehicle, the vehicle on the ground is limited by two constraints, and the vehicle speed, in body frame, is zero in the horizontal and vertical directions[36]. The observation noise is white noise. According to the actual vehicle and road conditions, the observed noise matrix is constructed as follows:

$$\begin{aligned} v_y^b &\approx 0 \\ v_z^b &\approx 0 \end{aligned} \tag{4-1}$$

In the ENU coordination, the vertical velocity is zero.

$$v_U^n \approx 0 \tag{4-2}$$

4.2 Wheel Speed Sensor

Vehicle wheel speed sensor (WSS) is a kind of sensor which can calculate the vehicle speed along the driving direction (X-axis) in the body frame by measuring the wheel speed. Wheel speed sensor is one of the most important sensors in vehicle sensors. At present, most of the vehicle navigation products on the market use the wheel speed signal to calculate the mileage[37, 38].

Generally speaking, the installation position of wheel speed sensor is different. Some wheel speed sensors are directly installed on the wheels, and many are installed on the transmission and reducer. However, although the installation position is different, the principle is to measure the wheel speed, and then calculate the longitudinal axis speed of the vehicle through the wheel rolling radius. When the wheel speed measurement is more accurate, the radius of the wheel becomes the main error source of calculating the speed. The wheel rolling radius is affected by tire wear, vehicle speed, tire load, road condition and external temperature[38, 39].

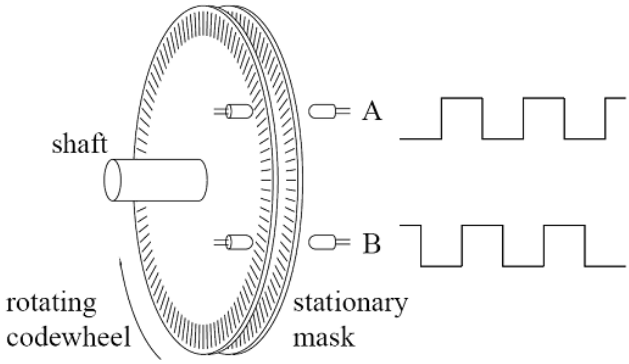


Figure 4-1 Wheel speed sensor working principle

In some integrated navigation systems and papers, wheel speedometer is used as speed measurement, and the scale factor of wheel speedometer is added to the estimator of Kalman filter. In this

paper, GNSS Doppler velocity is used as velocity observation. When GNSS velocity is not available, wheel speedometer information is used as velocity observation. In a short period of time, the scale factor of wheel speedometer does not change much, so the estimation of Kalman filter is the same as that in the previous chapter.

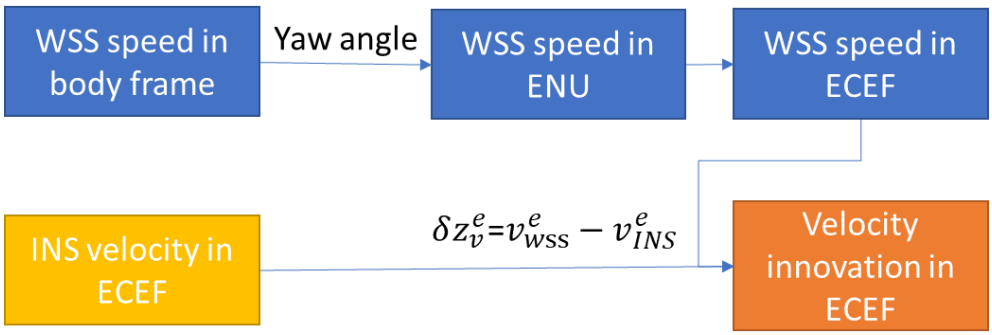


Figure 4-2 Barometer height and RTK height in ENU

When the current heading is known, the velocity vector of the wheel speedometer in the body coordinate system can be converted to the local plane rectangular coordinate system, namely the ENU coordinate system.

$$\begin{cases} v_E^n = \sin(heading)\boldsymbol{v_{WSS}^b} \\ v_N^n = \cos(\text{heading})\boldsymbol{v_{WSS}^b} \end{cases} \tag{4-3}$$

4.3 Barometer

Based on the vehicle motion model, gross errors of GNSS can be observed in the vertical direction. When GNSS is available, the barometer can be used to calculate the stable height, and the GNSS error can be obtained by comparing with the height of GNSS in the local horizontal coordinate system. When using RTK, this method can effectively avoid the influence of the error fixe solution and the long-time float solution on the positioning accuracy[40].

Without considering the temperature, barometric formula is as follows:

$$P = P_0 \cdot \left[1 - \frac{L \cdot h}{T_0}\right]^{\frac{Mg}{R^* L}} \tag{4-4}$$

Where:

P_0 is the Sea level standard atmospheric pressure, about 1013.25 hPa

h is the altitude

L is the rate of temperature decline, and for dry air, it is about $0.0065 K/m$

T_0 is the standard sea level temperature. The formula is to convert the degree of Celsius to Kelvin (thermodynamic temperature scale)

g is the acceleration of gravity on the earth surface, about 9.8 m/s^2

M is the mass of a substance with a molar mass, i.e., the amount of substance, about $0.0289644 kg/mol$

R is the universal gas constant, about 8.31447 $J/(mol \cdot K)$

So $\frac{Mg}{R^* L}$ is equal to 5.256, and it is easy to see that altitude is:

$$h = 153.8 \cdot (t_0 + 273.2)\left(1 - \left(\frac{p}{p_0}\right)^{0.1902}\right) \tag{4-5}$$

Here, using 2021/06/09 data for example. The actual air pressure and temperature almost unchanged after 40 minutes of driving in Tokyo. Due to the MEMS IMU's temperature is the internal temperature, and IMU temperature is affected by the sunshine, wind and heat from the car, compare with RTK result, the height from IMU is not stable and has some noises.

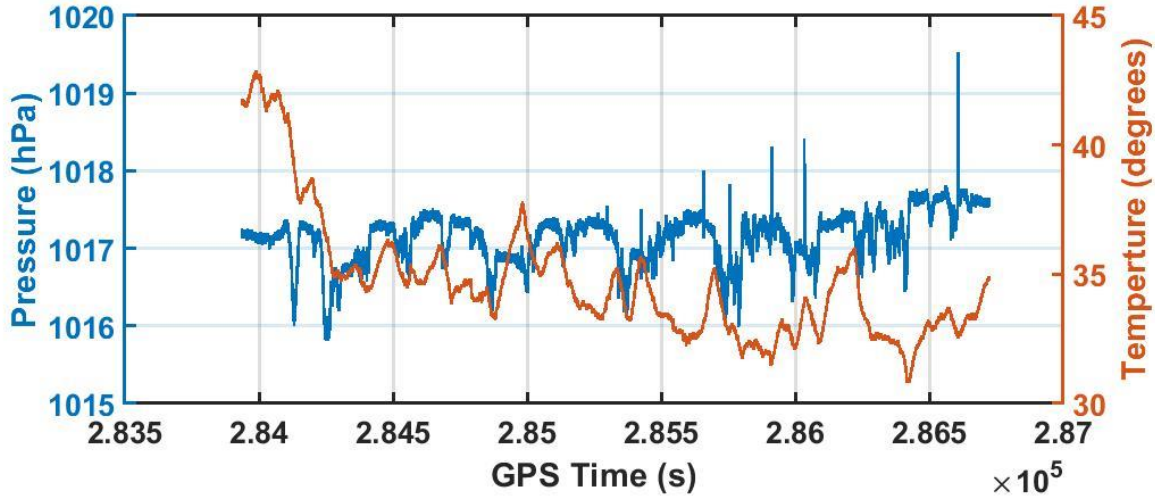


Figure 4-3 Estelle IMU temperature and barometer

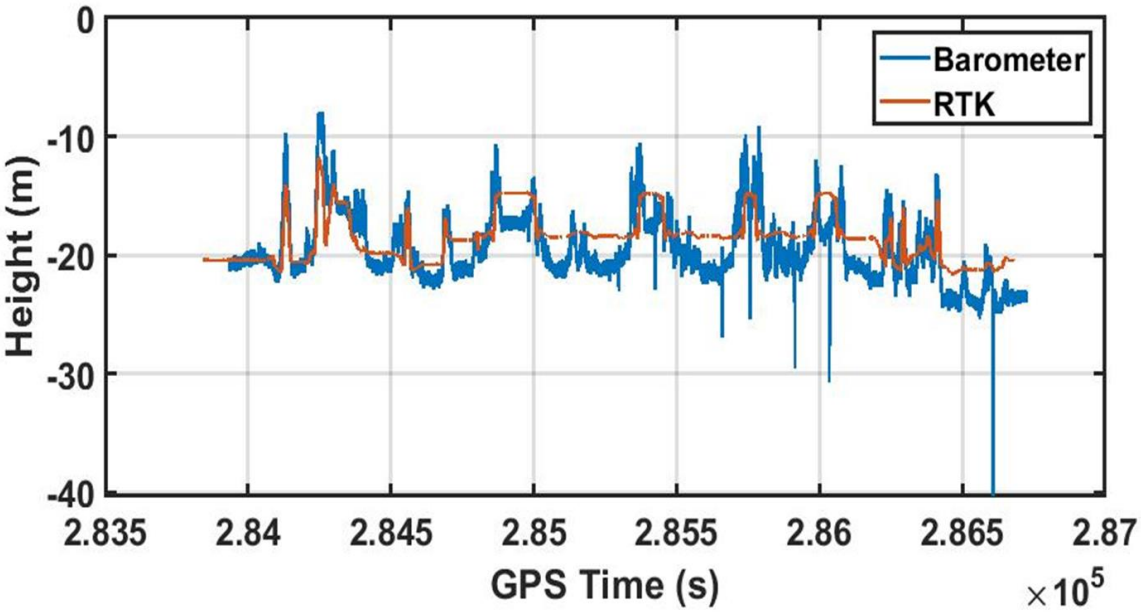


Figure 4-4 Barometer height and RTK height in ENU

4.4 Zero Angular Rate Update

For vehicles on the road, when the vehicle is stationary, the output angular rate of the gyroscope should be equal to zero, which is equivalent to obtaining the measurement information of the gyroscope. It can be used as Kalman filter observation into Kalman filter. 3 axis zero angular rate update is useful only when the gyro error is much larger than the angular interference. Therefore, 3 Zero angular rate update is

sometimes only used for azimuth axis, and it is not common to use it for high-precision gyro. When the Kalman filter do the zero angular rate update, the innovation is as follows:

$$\delta z_{ZA,k}^{-} = -\widehat{\omega}_{ib,k}^{b} \tag{4-6}$$

The measurement matrix is as follows:

$$\boldsymbol{H}_{ZA,k} = (0_3 \quad 0_3 \quad 0_3 \quad 0_3 \quad -I_3 \quad 0) \tag{4-7}$$

4.5 GNSS Compass

GNSS compass means using double GNSS antennas positioning, and get the heading from back antenna to front antenna. From chapter 4.1, It is easy to know that the vehicle motion model's roll, pitch can be set as zeros. the GNSS compass can provide the yaw. The attitude in ENU coordinate can be obtained. This paper calculates the innovation in ECEF coordinate, so it is necessary to convert the attitude innovation from ENU to ECEF.

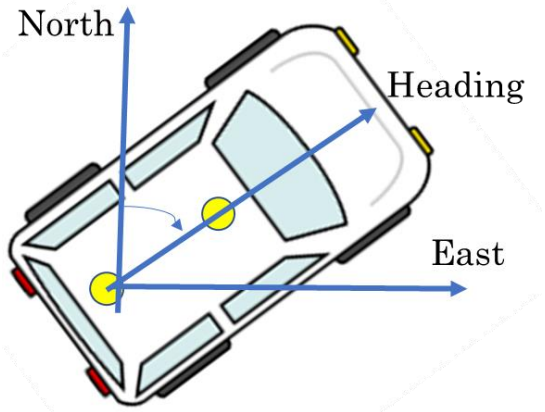


Figure 4-5 GNSS compass

Here, firstly, converts the attitude matrix to Euler matrix (roll-pitch-yaw). Next, get the innovation. Finally, convert the innovation to ECEF.

the attitude innovation is as follows:

$$\delta z_{G,k}^{e-} = \widehat{\boldsymbol{\psi}}_{eaG}^{e} - \widehat{\boldsymbol{\psi}}_{eb}^{e} \tag{4-8}$$

The measurement matrix is as follows:

$$\boldsymbol{H}_{\psi,k} = (-I_3 \quad 0_3 \quad 0_3 \quad 0_3 \quad 0_3) \tag{4-9}$$

The source code in "TC_KF_Epoch.m" is as follows:

```
old_Euler=CTM_to_Euler(est_C_b_n_old');
delta_heading_n=DoubleGNSSHeadingEpoch(1,1)-TC_KF_config.HeadingBias-old_Euler(3);
%The course measured by two antennas is subtracted from the course calculated by recursion
delta_z(no_meas*2+1:no_meas*2+3,1)=HeadingNToEulerE(delta_heading_n,est_L_b_old,est_lambda_
b_old);
```

5. EXPERIMENT AND RESULT

To test the GNSS/INS program, here using different data set for test. It should be noted that the equipment used in each experiment has not changed. The position of IMU is fixed behind the car bumper, but the position of GNSS antenna is not fixed. The GNSS receiver using Ublox-F9P, base station is by Tokyo university of marine science and technology, the IMU is by Epson G370, the wheel speed sensor, temperature, air pressure is obtained from Estelle. If poslv available, using poslv position as the true value for comparison, if not, using F9P-RTK result for comparison. For better analysis and compare the result, using Google Earth and QGIS for plot.

Table 5-1 List of the equipment

Equipment	Output	Frequency	Note
Ublox-F9P	GNSS information	5 Hz	Integrated with Estelle, synchronize the clock of sensors
Epson G370	Gyro and acceleration	50 Hz	Be used in Tsukishima 1st test and Marunouchi 2nd test
Estelle	Wheel speed sensor, temperature, air pressure	50 Hz	Be used in Tsukishima 2nd test and Marunouchi 1st test
Poslv	Position	200 Hz	

The LC GNSS setting is:

Elevation mask: 15 degrees;

SNR mask: 35 dBHz;

Frequency: L1 + L2;

Satellite system: GPS GALILEO QZSS BDS;

Integer ambiguity res: Instantaneous;

The TC setting is:

Elevation mask: 15 degrees;

SNR mask: 35 dBHz;

Frequency: L1;

Satellite system: GPS GALILEO QZSS BDS;

Clock jitter: repaired;

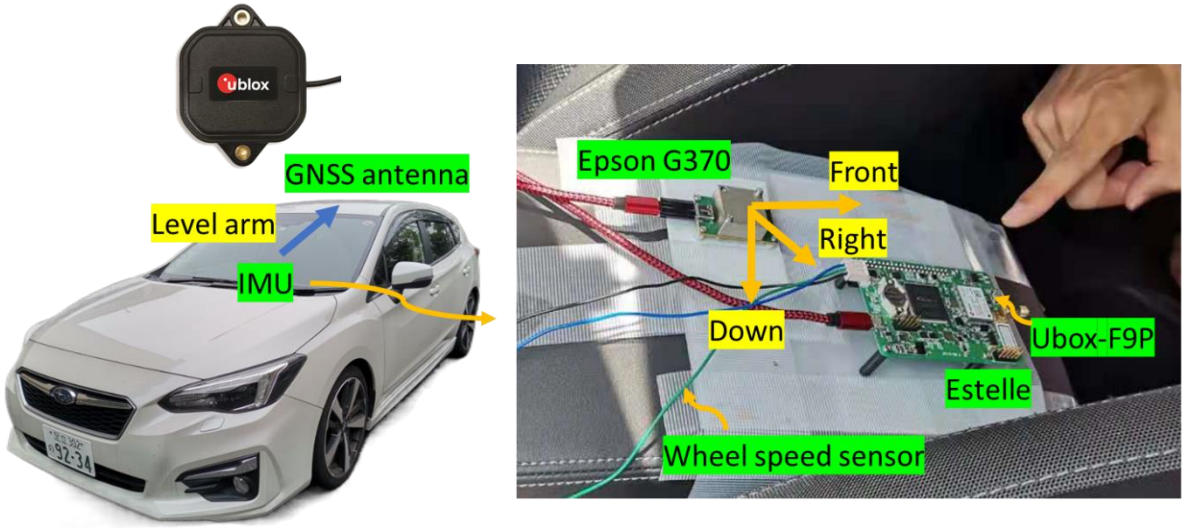


Figure 5-1 Setup mounted on board of a car

Firstly, using data of 2021/04/12, this data was obtained from Tsukishima, Kotoku, Tokyo. For RTK in Tsukishima, as the figure 5-2 shows, there are several challenge places. When the car goes through the bridge, the RTK signal is blocked, at that time, only doppler velocity available.

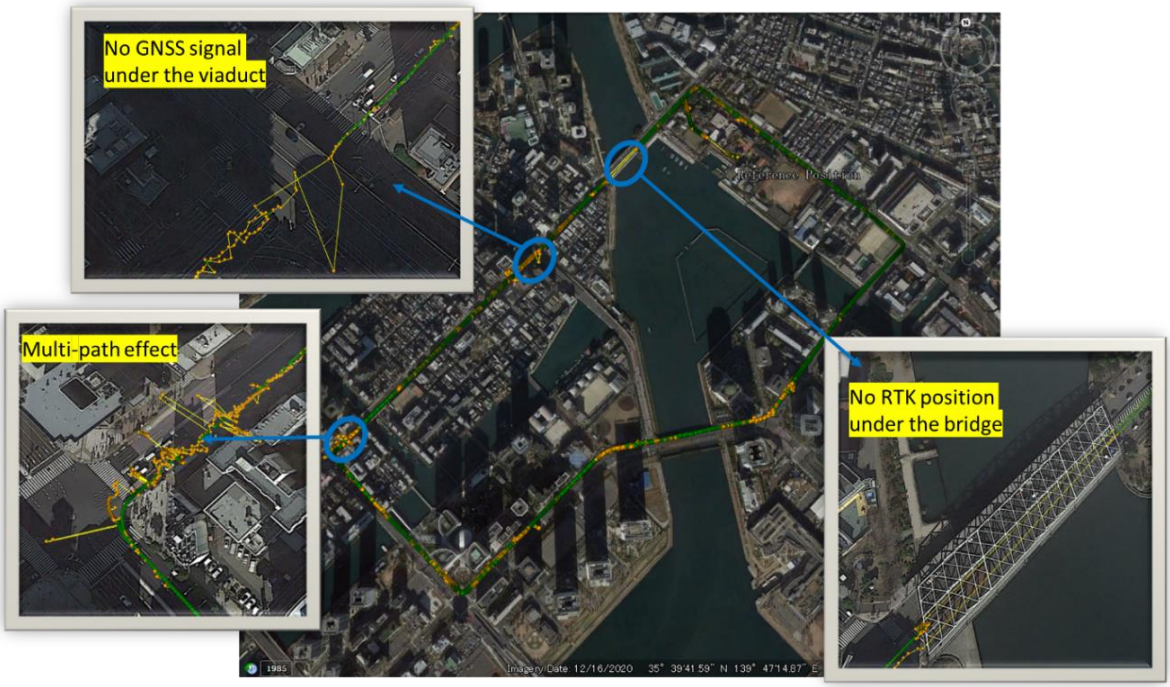


Figure 5-2 RTK result in Tsukishima

5.1 Compare the with velocity and without velocity update

The chapter 3.6.1 has introduced that if the GNSS/INS program only has position and velocity update, when the car under the bridge, the position result is only from mechanization equation. For MEMS IMU, the error will be very huge. so, I did a test about using only velocity and without using only velocity update. Considering that this special situation occurs in a small number of road sections, this section only shows the positioning when the car goes through the bridge.

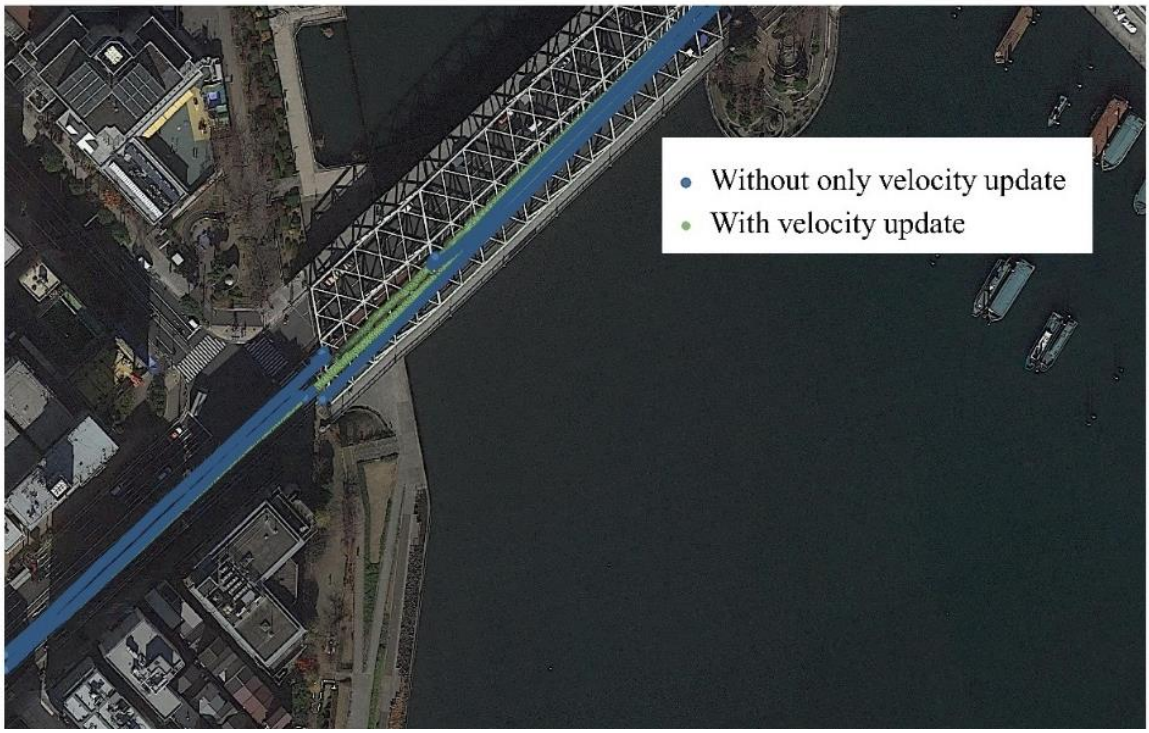
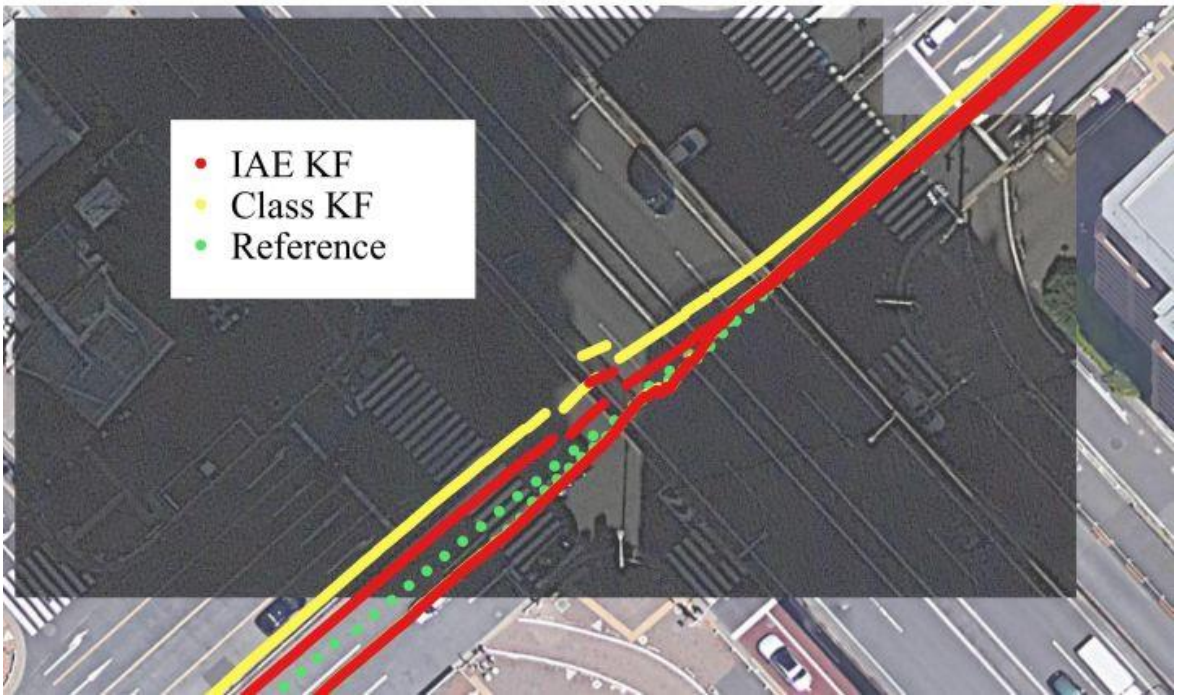


Figure 5-3 Comparison before and after adding only velocity update

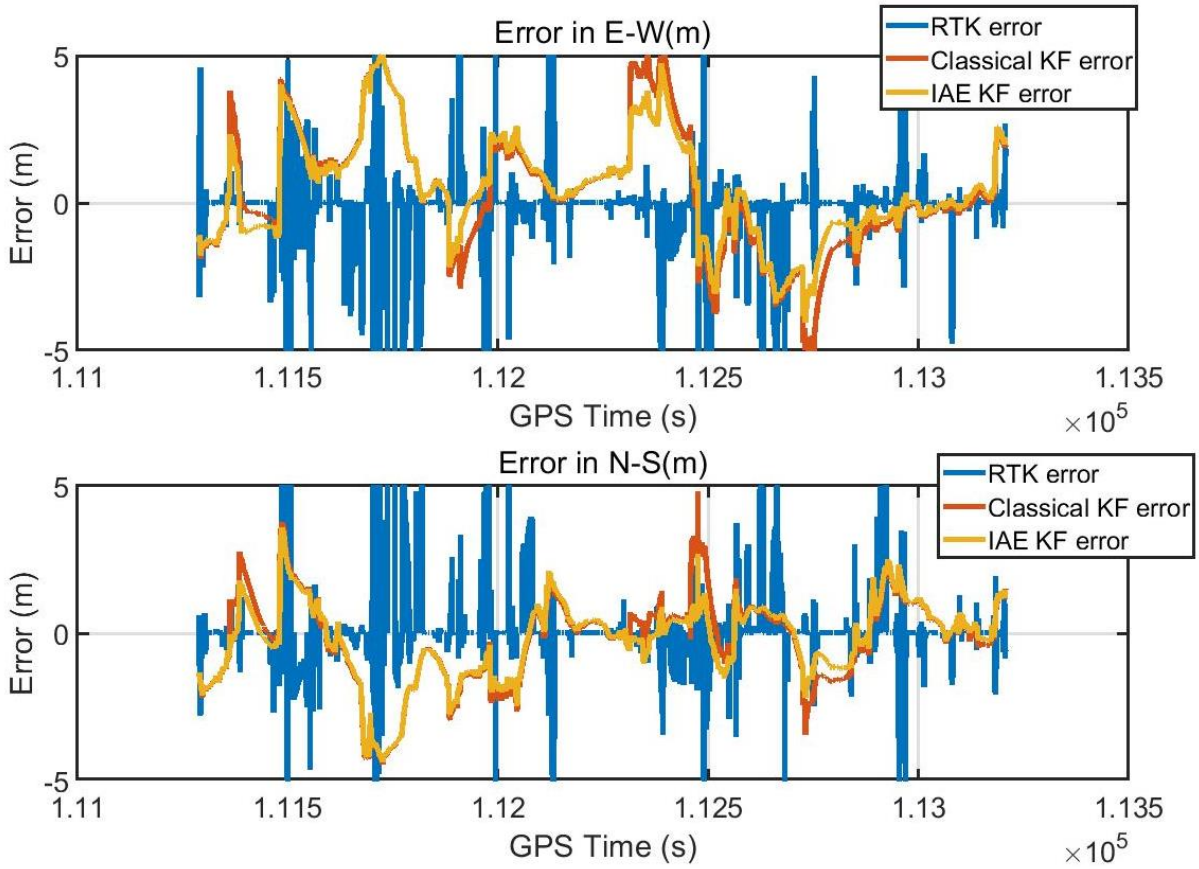
It is easy to see that, the blue line is only position and velocity update KF, the green line is added the velocity update. When there is no RTK position available, the INS bias will not update, the position error will be accumulated. After using the velocity update, the GNSS information is fully used, and the result shows that the heading error and distance error is reduced.

5.2 Compare the Classical KF and IAE-KF

In chapter 3.6.2 has introduced the IAE-KF. IAE-KF using the standardization innovation to update the reweight of each satellite measurement noise covariance. Here, using Tsukishima data for test, and set the Ublox F9P-RTK data as reference. From figure 5-4, we can see that although both Classical KF TC result and IAE-KF TC result has error, the IAE-KF result is smaller. In table 5-1, shows the maxim error, mean error and the 95% confidence interval in horizontal. In later experiments, the IAE-KF will be always used.



(a)



(b)

Figure 5-4 Compare Classical KF and IAE-KF

In (a) is the track under the viaduct. In (b) measurements over time: error in the East direction (left) and error

in the North direction (right).

Table 5-1 Error of Classical KF and IAE-KF in horizontal

UNIT (M)	MAX-E	MAX-N	MEAN-E	MEAN-N	95%-E	95%-N
IAE-KF	5.1099	4.3755	0.4866	0.2565	4.3426	3.8570
CLASSICAL KF	6.5025	4.8021	0.3887	0.2160	4.7789	3.8949

5.3 Compare the GNSS/INS LC and GNSS/INS TC

There are many data set for tests, different data sets have different urban scenarios. And this section will present the metrics used to assess the GNSS/INS LC and TC performances. As the figure 5-5 shows that, these datasets are obtained in downtown Tokyo.

Three portions were selected, as considered more relevant for the objective of the tests:

1. In Tsukishima, some places the GNSS signal is poor, but the environment isn't always challengeable;

2. In Marunouchi, there are many tall buildings and it is hard to receive continuous GNSS signals. Also, it has one of the largest rail way station, there are many viaducts, limiting the number of the satellites in view. It is a typical urban canyon environment;

The positioning performance was assessed through specific metric metrics. For all these datasets, analyzed the horizontal position error in terms of its mean, standard deviation and 95th percentile. And compared the yaw angle of the attitude.

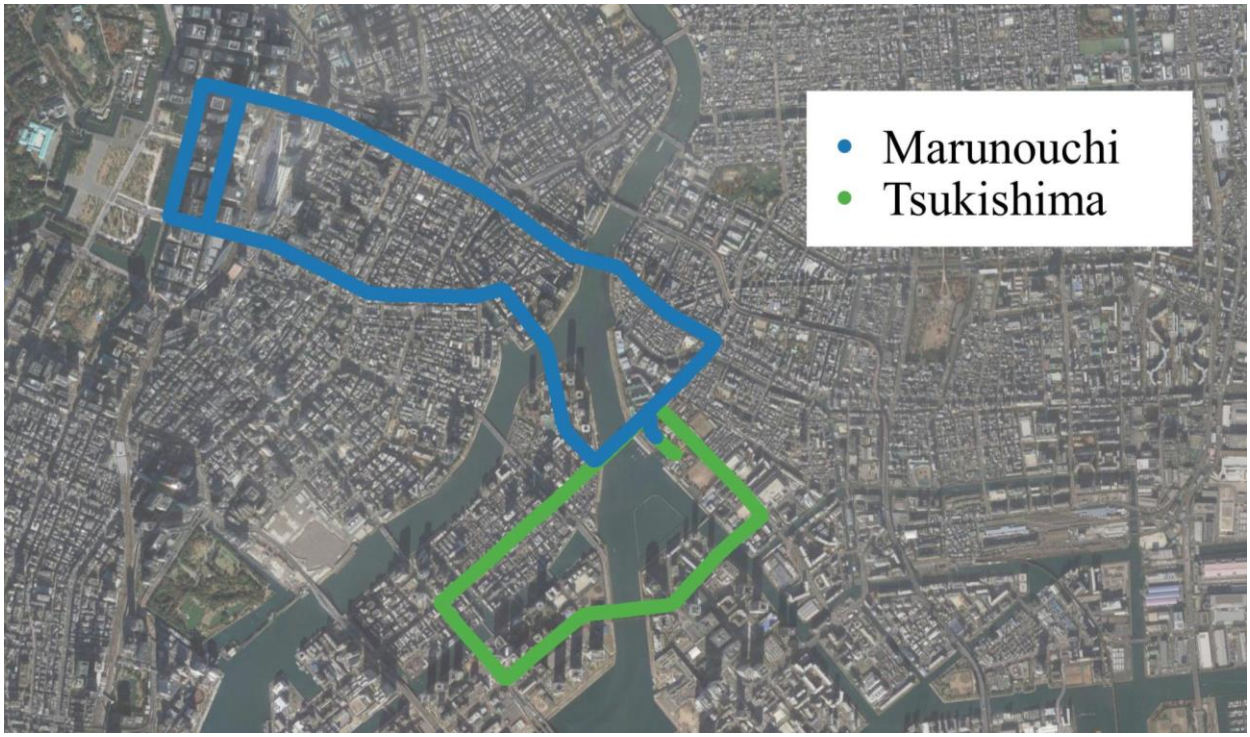


Figure 5-5 Trajectory followed during the experimental tests in Tokyo

5.3.1 Tsukishima Dataset

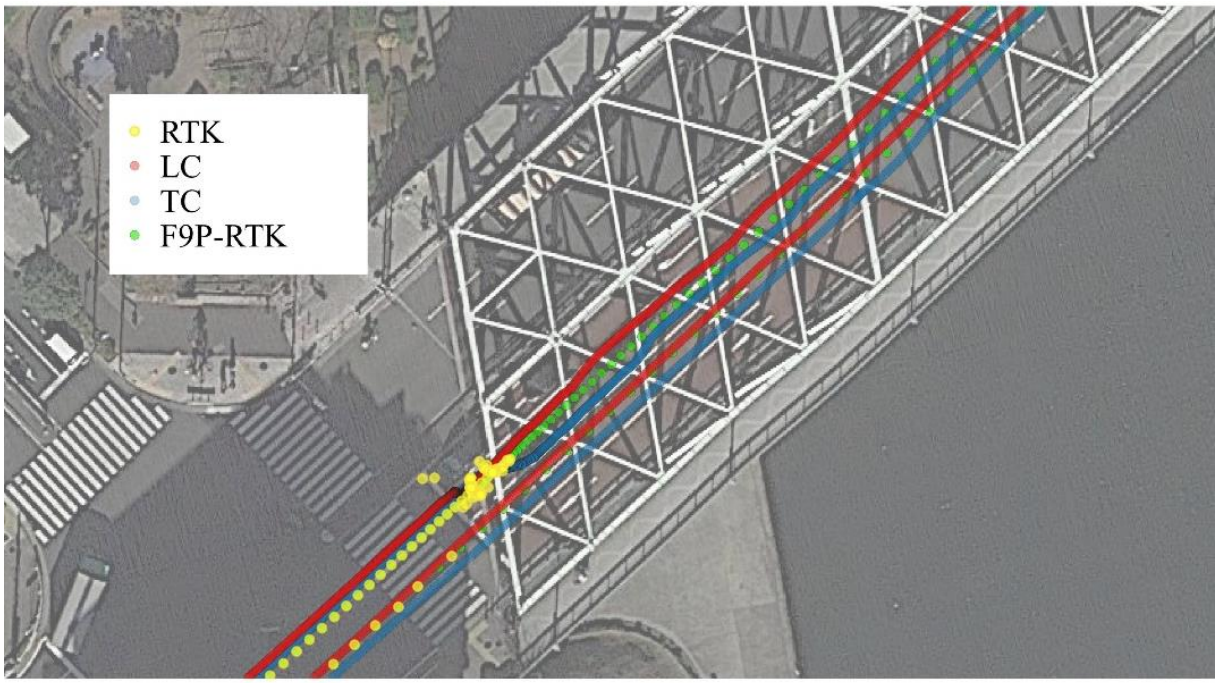
The 1st data set is in Tsukishima, Kotoku, Tokyo. As figure 5-2 and 5-4 shows that there is a viaduct and has multi-path effect in the crossroads. To better compare the LC and TC, I marked 4 places.

- No. 1 is a metal bridge, the carrier phase is blocked and no RTK position;
- No. 2 is under a viaduct, the number of satellites is 0, both LC and TC will lost GNSS measurement and output the INS results;
- No.3 is cross-roads, the car is surrounded by tall buildings, the multi-path effect is very huge;
- No. 4 has some tall builds in the south, the GNSS error is also huge there.

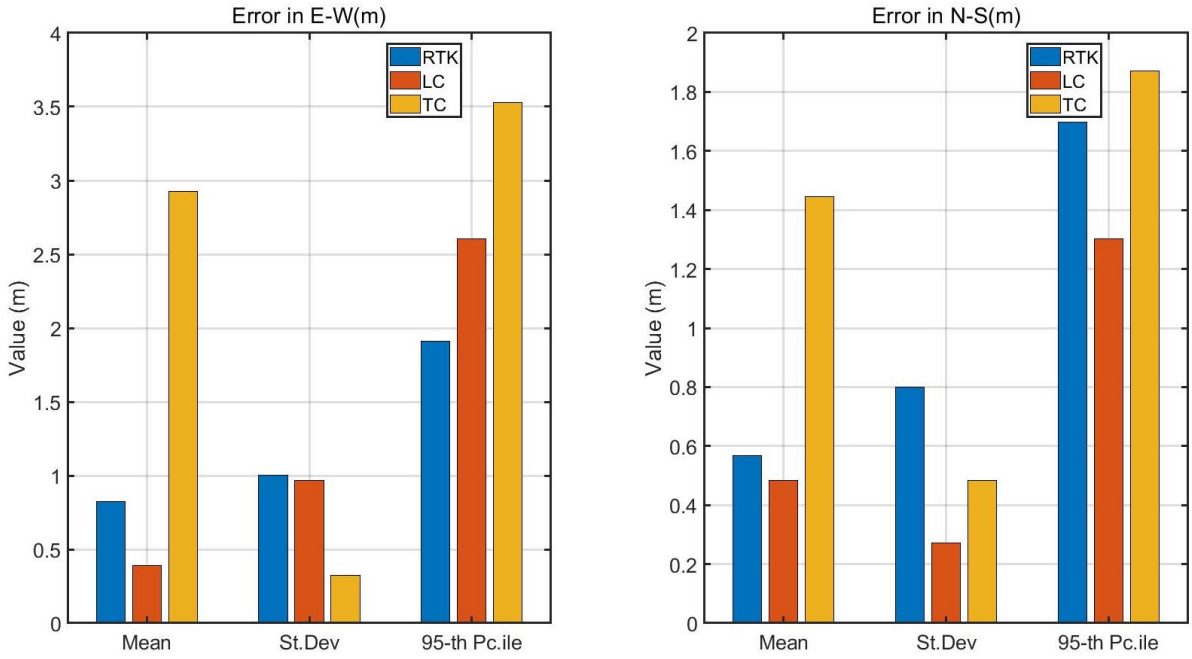
Figure 5-6 Tsukishima dataset track

1. In place No. 1

The place No. 1 is start from the car came to the bridge and end at the car went out of the bridge. The duration is about 1 minute.



(a)



(b)

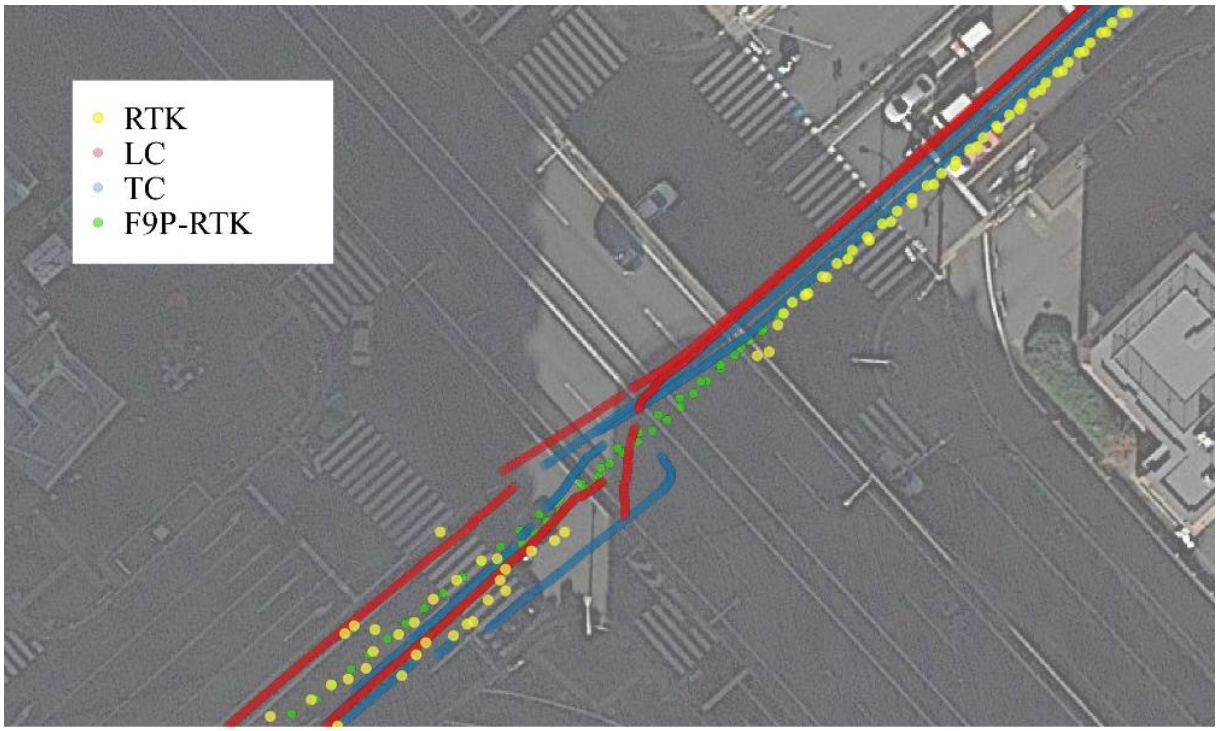
Figure 5-7 Tsukishima dataset track in No. 1 place

In (a) the LC, TC and F9P-RTK track were plot. In (b) measurements over time: error in the East direction (left) and error in the North direction (right).

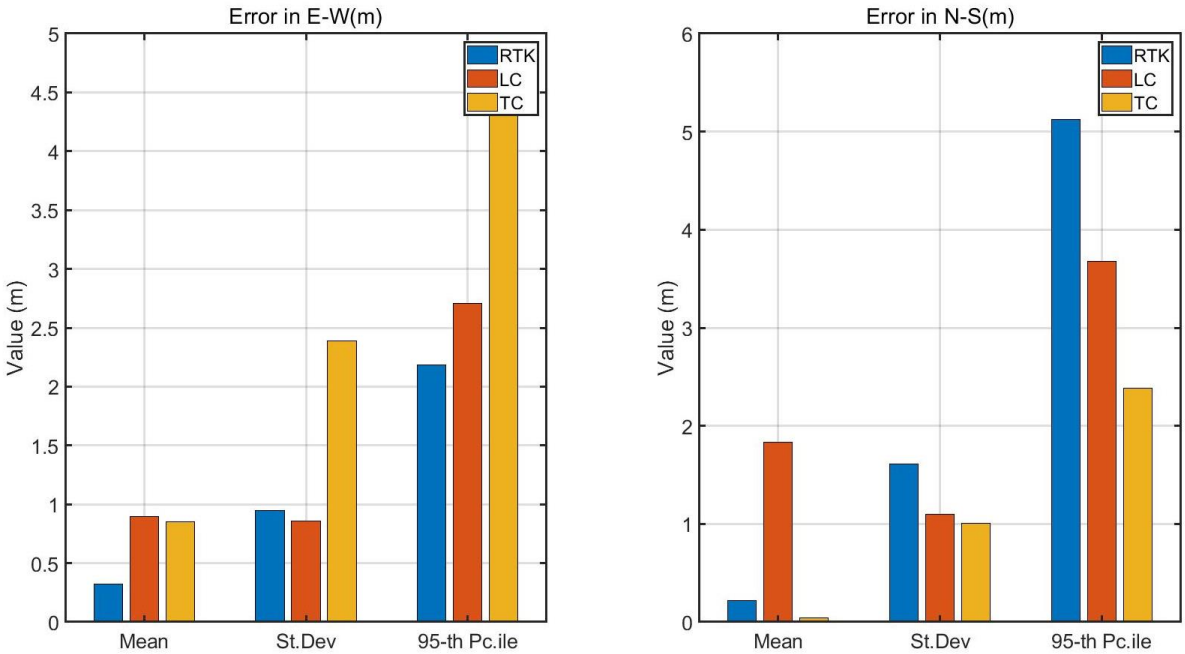
LC only uses Doppler velocity to update KF in this region, pseudo range and doppler frequency can still be received, the TC is not affected. It can be seen that the attitude deviation of LC occurs when there is no position observation.

2. In place No. 2

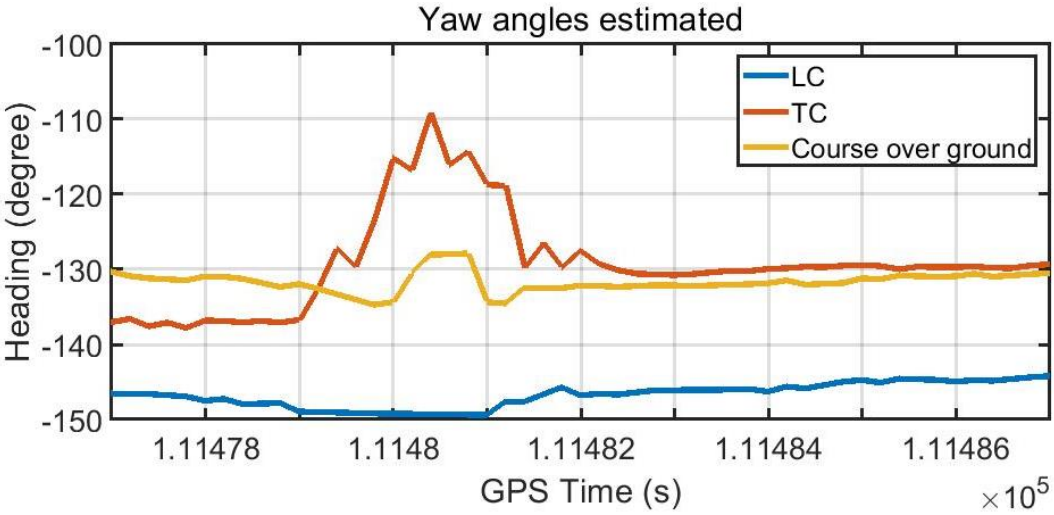
The place No. 2 start from the viaduct right zebra crossing and end at the left side crossing. The duration is about 10 seconds.


RTK
LC
TC
F9P-RTK

(a)


Error in E-W(m)
Value (m)
RTK
LC
TC
Mean
St.Dev
95-th Pc.ile
Error in N-S(m)
Value (m)
RTK
LC
TC
Mean
St.Dev
95-th Pc.ile

(b)



(c)

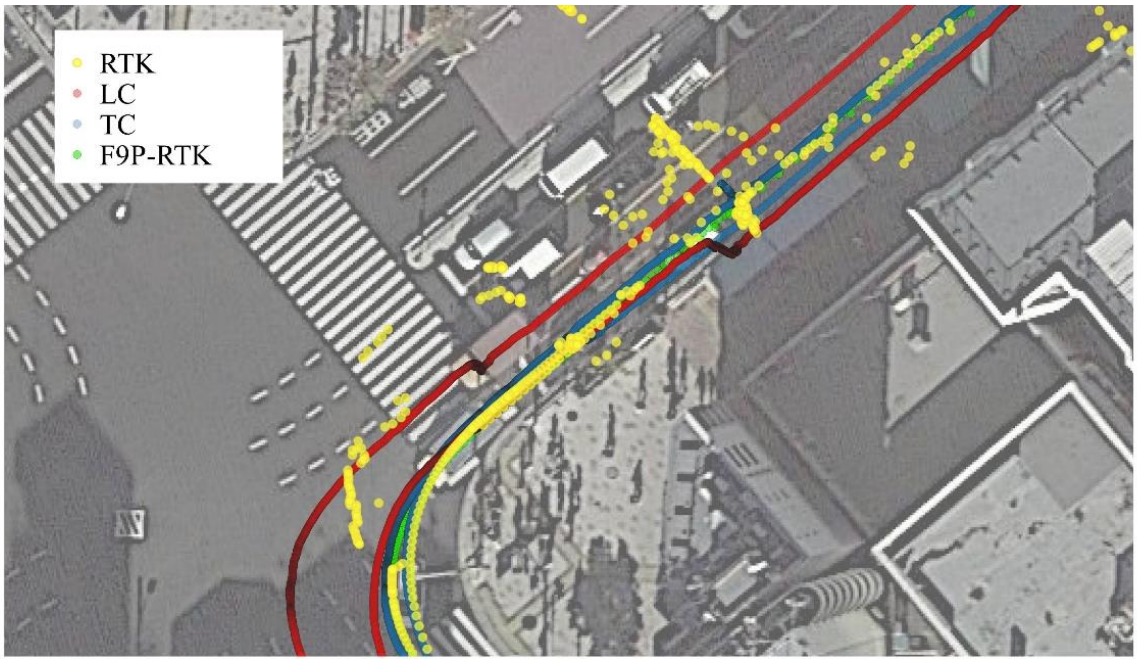
Figure 5-8 Yaw angles under the viaduct

In (a) the LC, TC and F9P-RTK track were plot. In (b) measurements over time: error in the East direction (left) and error in the North direction (right). In (c) LC yaw angle, TC yaw angle and Ublox course over ground.

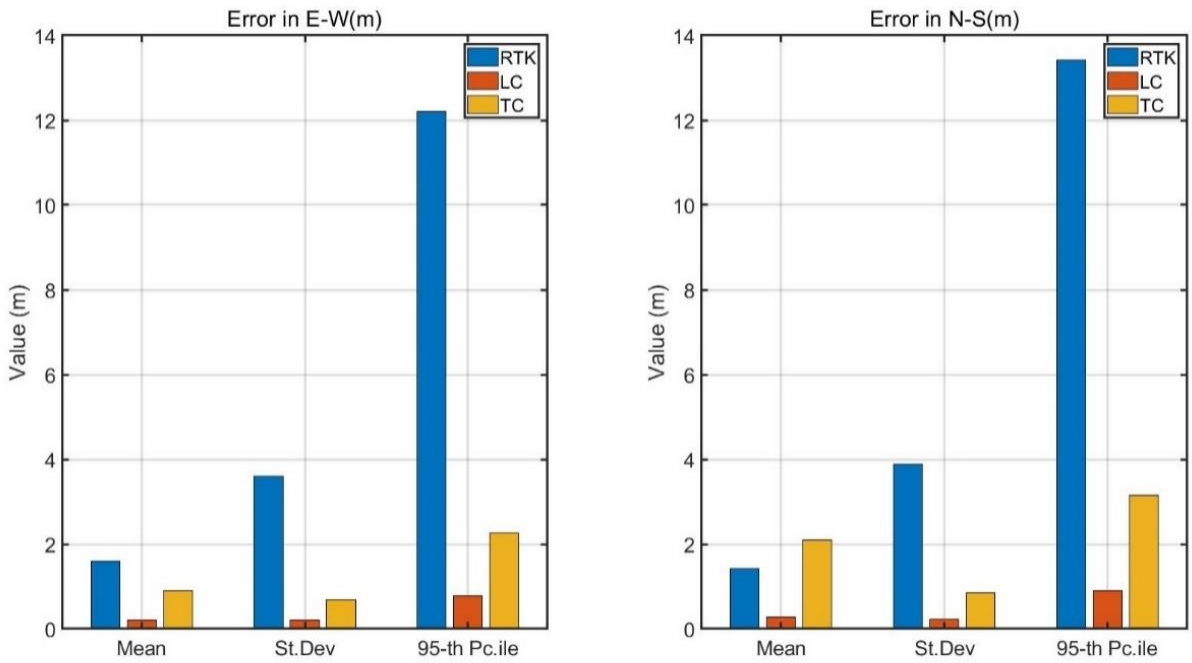
From the result, it can be seen that when there is no GNSS, both LC and TC output the INS results. The TC attitude is more stable than LC.

3. In place No. 3

The car parking in this is for about 3 minutes. And suffered from the GNSS multi-path effect.



(a)



(b)

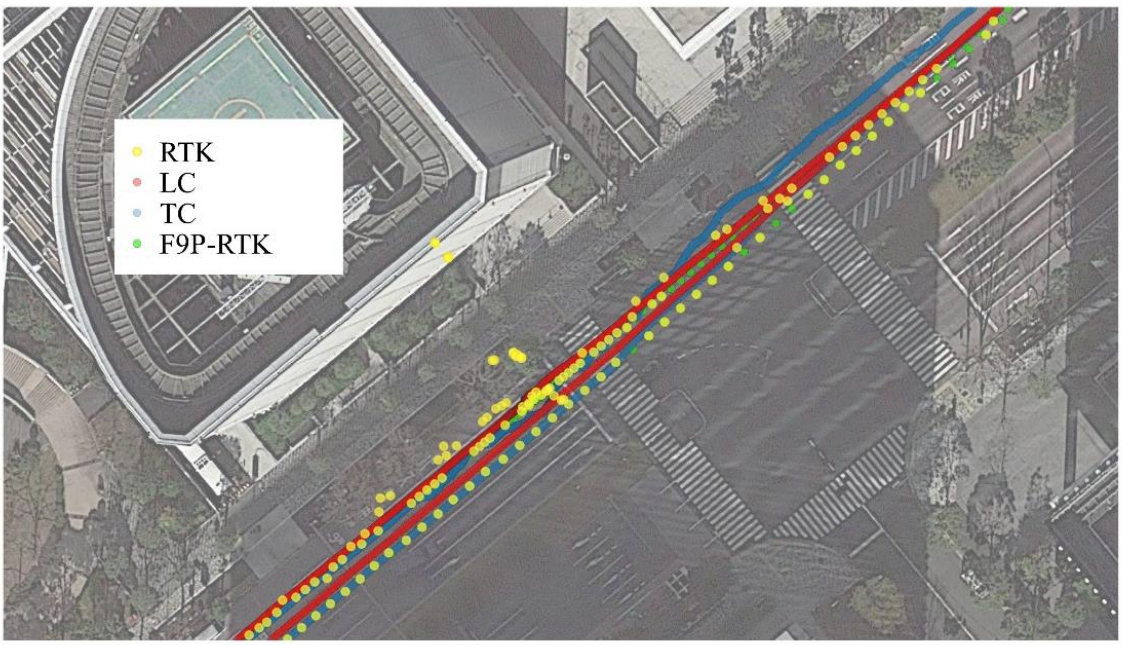
Figure 5-9 Tsukishima dataset track in No. 3 place

In (a) the LC, TC and F9P-RTK track were plot. In (b) measurements over time: error in the East direction (left) and error in the North direction (right).

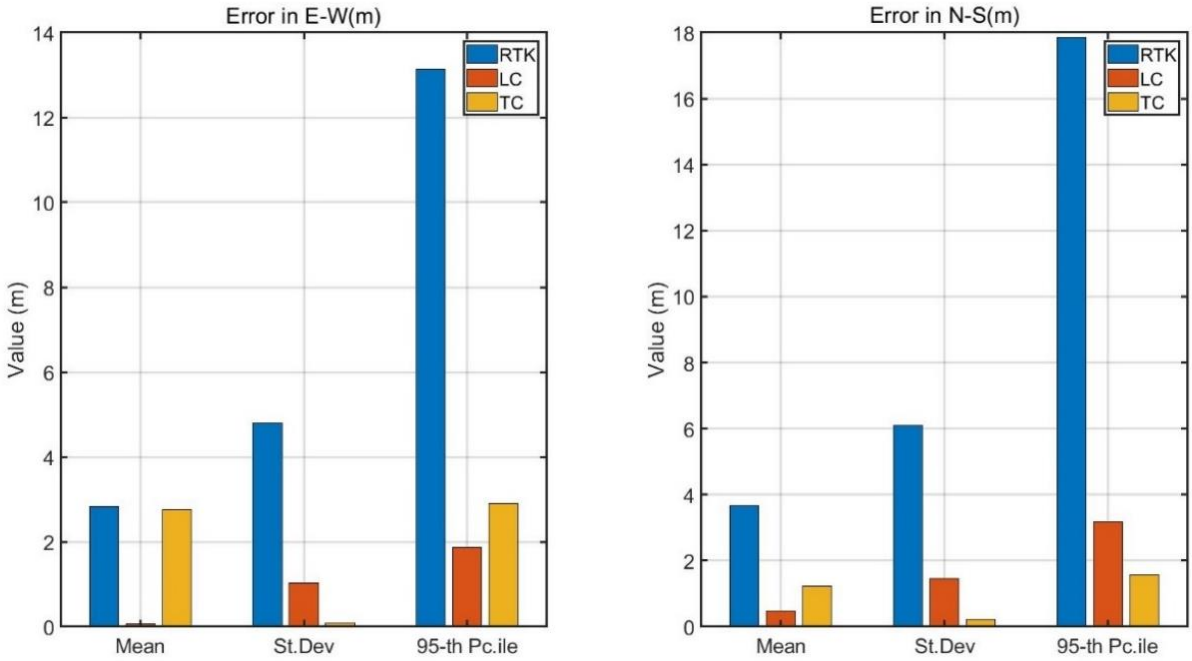
The car has been parking in this area for a while, and it is affected by multi-path. LC has velocity observation, which can eliminate the influence of multipath error on positioning. For TC, the pseudo range rate can also resist multipath error.

4. In place No. 4

The place No. 4 in the south of the Tsukishima, the area is same as the figure 5-10 (a) shows. The duration is about 40 seconds.



(a)



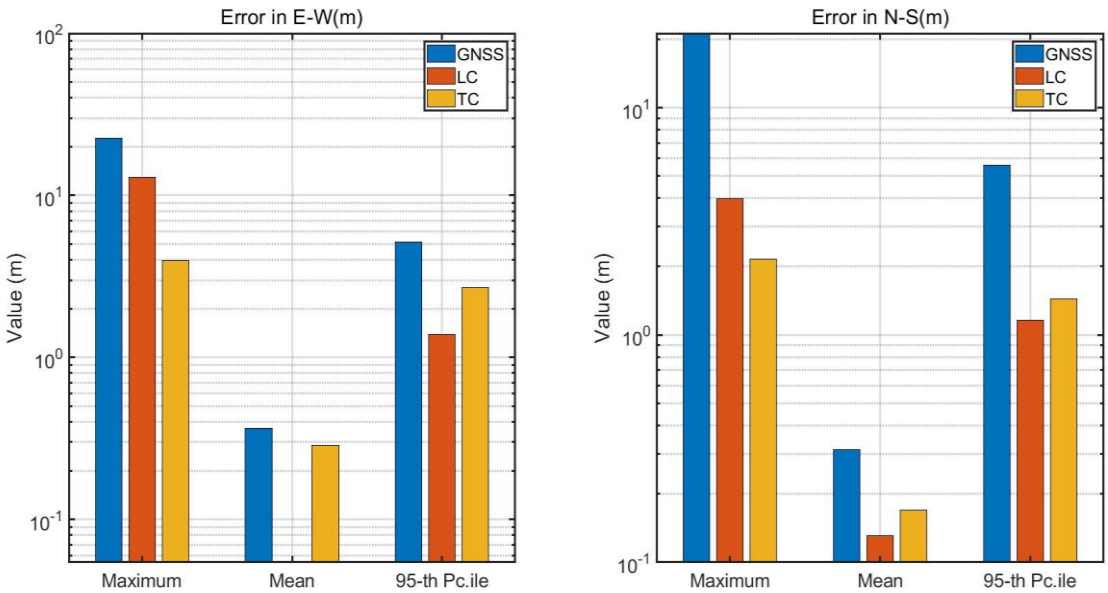
(b)

Figure 5-10 Tsukishima dataset track in No. 4 place

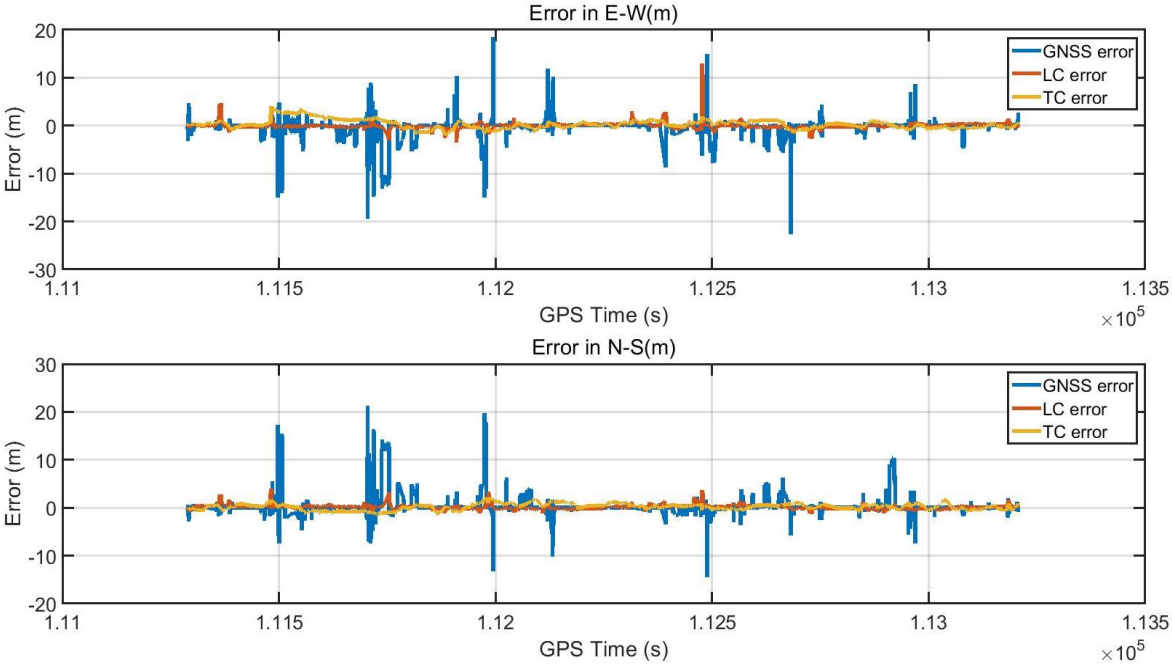
In (a) the LC, TC and F9P-RTK track were plot. In (b) measurements over time: error in the East direction (left) and error in the North direction (right).

The GNSS is affected by the tall buildings. There are errors in velocity and position measurements. However, TC can determine the measurement noise of each satellite according to the altitude angle of the visible satellite and reduce the error.

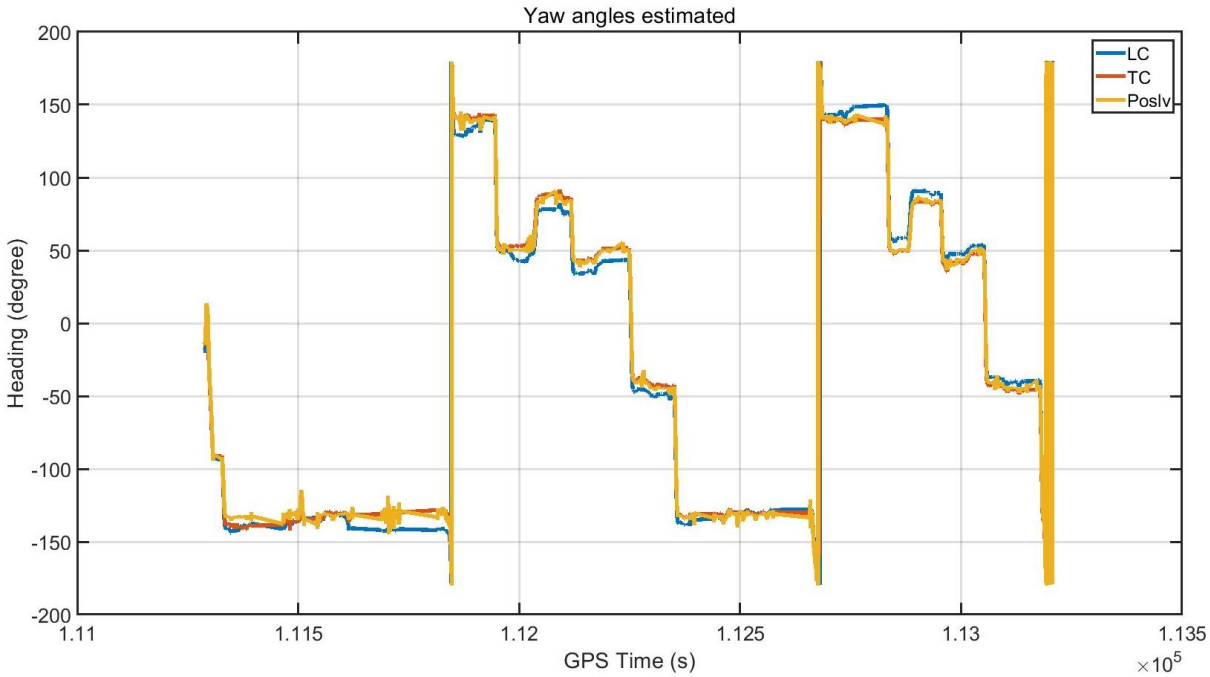
For the 20210412 data, FIX rate is about 57.1%:



(a)



(b)



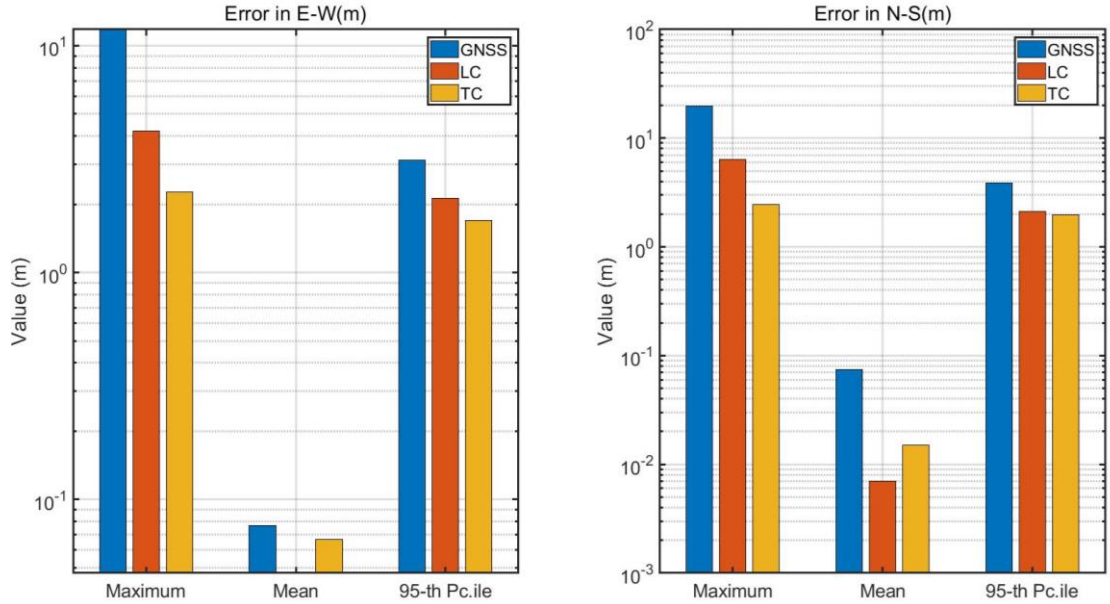
(c)

Figure 5-11 Tsukishima 20210412 error

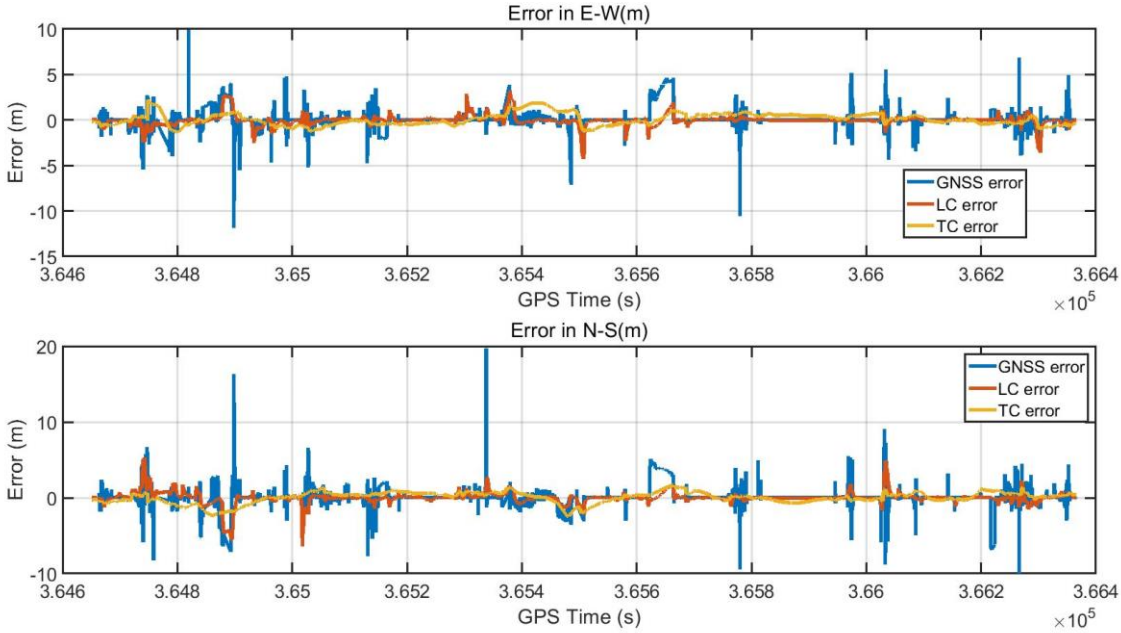
Horizontal positioning errors in 0302 1st dataset. In (a) measurements over time: error in the East direction (left) and error in the North direction (right). In (b) metrics associated to the error in the East direction (above) and the error in the North direction (below). In (c) LC yaw angle, TC yaw angle and Ublox course over ground.

The 20201105 data RTK FIX rate is about 81%. The GNSS signal quality is good and satellites are

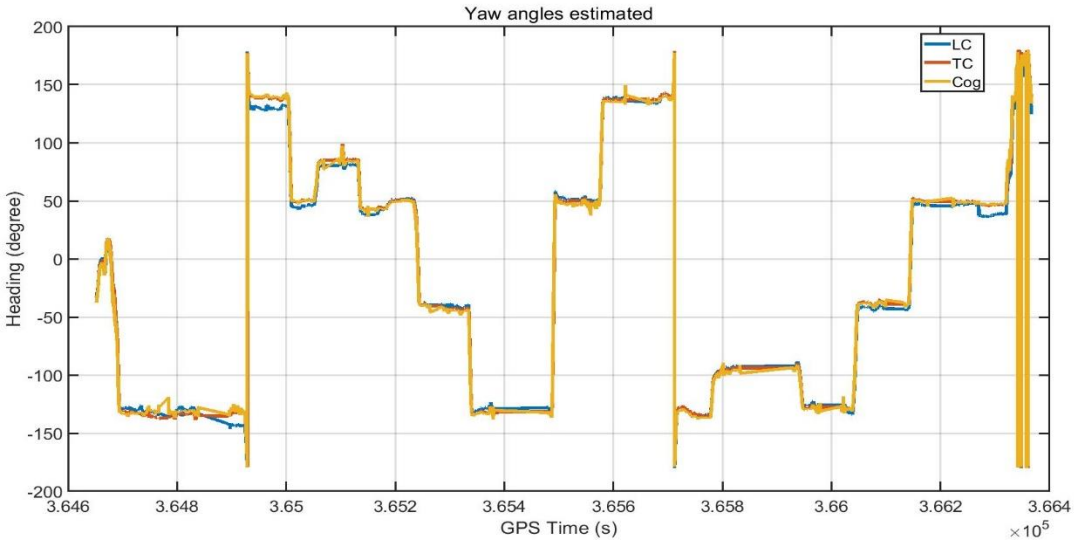
enough for use. The error of the RTK, LC and TC are smaller than the previous dataset result.



(a)



(b)



(c)

Figure 5-12 Tsukishima 20201105 error

Horizontal positioning errors in 0302 1st dataset. In (a) measurements over time: error in the East direction (left) and error in the North direction (right). In (b) metrics associated to the error in the East direction (above) and the error in the North direction (below). In (c) LC yaw angle, TC yaw angle and Ublox course over ground.

Table 5-2 Error of Tsukishima

	UNIT (M)	MAX-E	MAX-N	MEAN-E	MEAN-N	95%-E	95%-N
1ST	GNSS	22.6471	21.2237	0.3653	0.3127	5.1651	5.5614
	TC	3.9694	2.1577	0.2866	0.1700	2.6873	1.4457
	LC	12.9120	3.9714	0.0544	0.1307	1.3885	1.1613
2ND	GNSS	11.7968	19.6548	0.0767	0.0739	3.1382	3.8893
	TC	2.2633	2.4547	0.0669	0.0151	1.6963	1.9741
	LC	4.2219	6.3466	0.0474	0.0070	2.1314	2.1126

In Tsukishima, the maxim error is taken placed in the place 2. The TC attitude estimation is better than LC, so, when there is no GNSS measurements for KF update, the INS error form TC is smaller than LC. That is why the LC maxim error is larger than TC. In 20201105 data, the GNSS available satellites are quite good, and the car go through the GNSS challenge places quickly, RTK and LC mean error is smaller than TC. Compare with these two data, the 1105 data using Estelle IMU, this IMU noise is larger than G370, so the mean error is larger than 0412 data. TC doesn't have any advantages in such kind of condition. In the normal urban environment, the maxim error of RTK is smaller, so the LC is better and TC.

5.3.2 Marunouchi Dataset

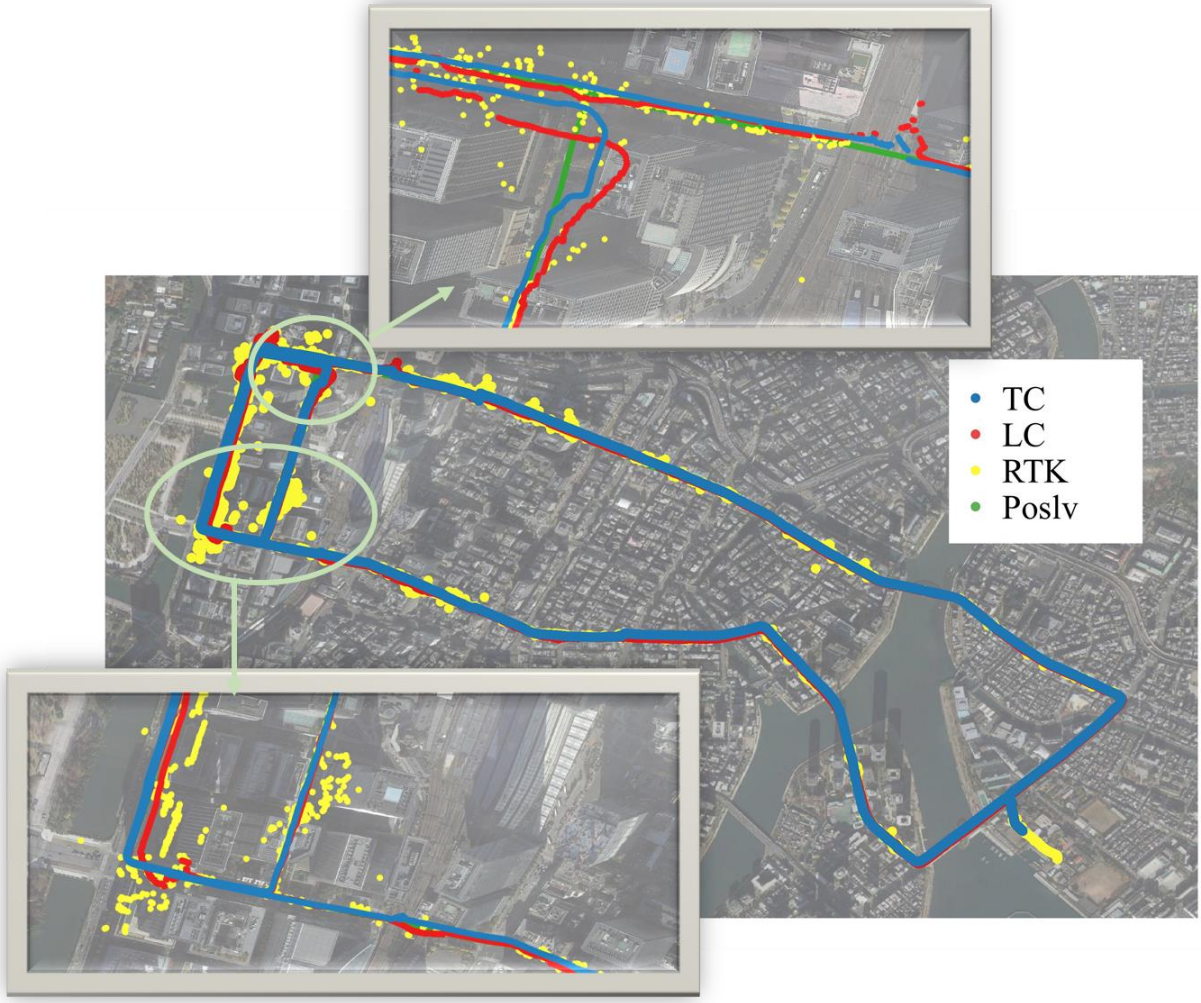
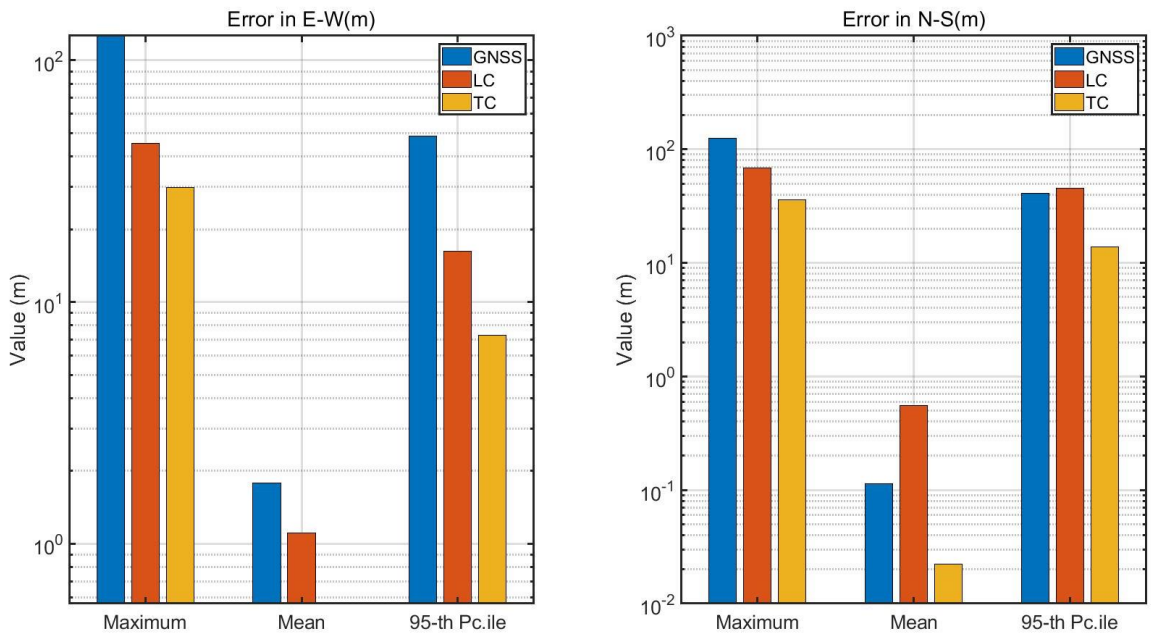
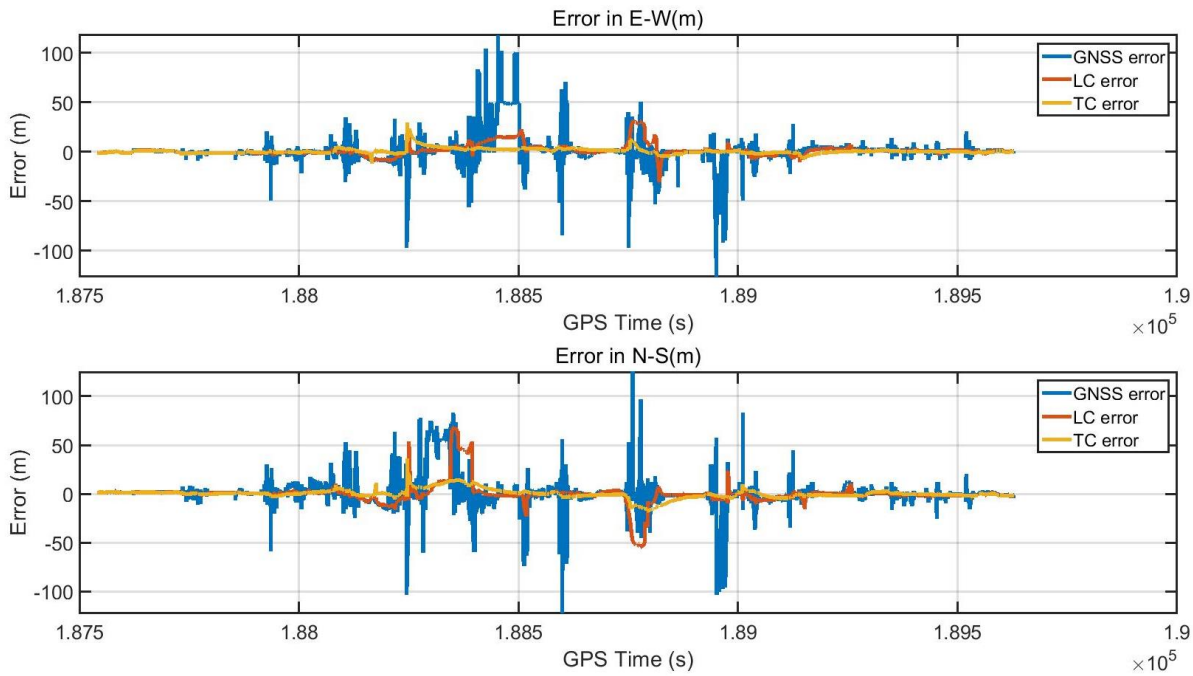


Figure 5-13 Marunouchi track of 0302 1st test

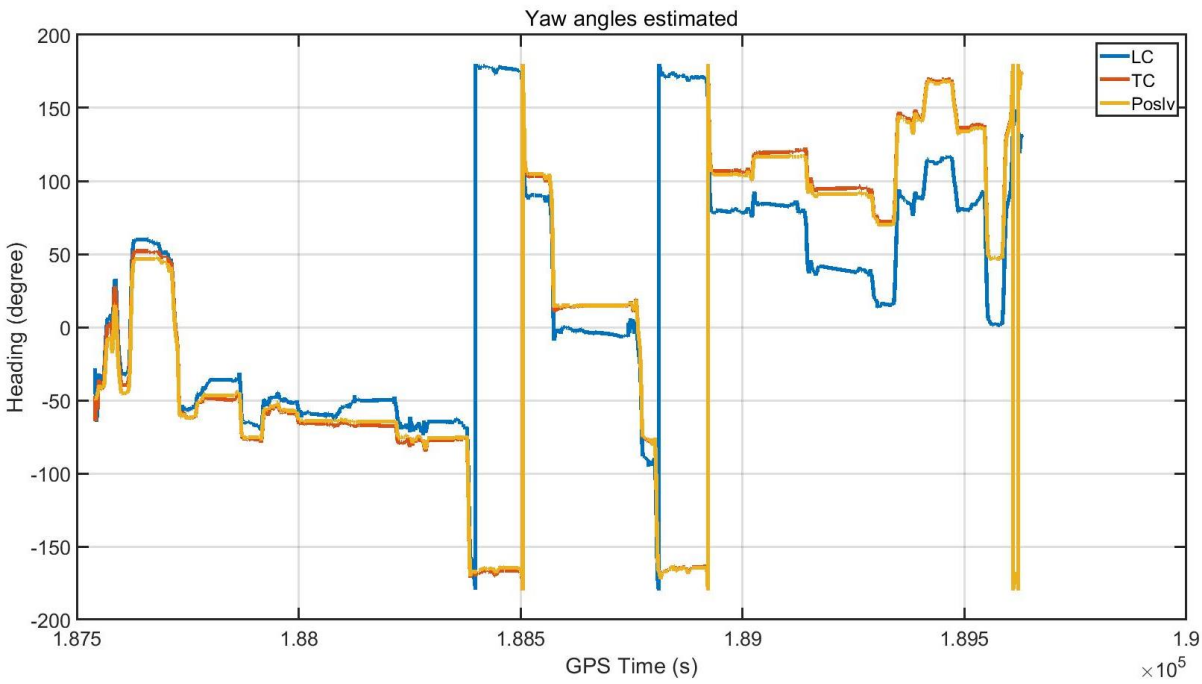
The car drive in the Marunouchi twice, start form university, went to the Tokyo station, turned around and went back. There are many tall buildings and viaducts in this area, the GNSS epoch is not continue and the error is huge. For the 1st dataset, the RTK FIX rate is about 59%. From figure 5-13, it is easy to see that TC has a large deviation on the left side of the image, and this error continues until the end of positioning.


Error in E-W(m)
Error in N-S(m)
GNSS
LC
TC
Value (m)
Maximum
Mean
95-th Pc.ile

(a)


Error in E-W(m)
Error in N-S(m)
GNSS error
LC error
TC error
Error (m)
GPS Time (s)
×10^5

(b)



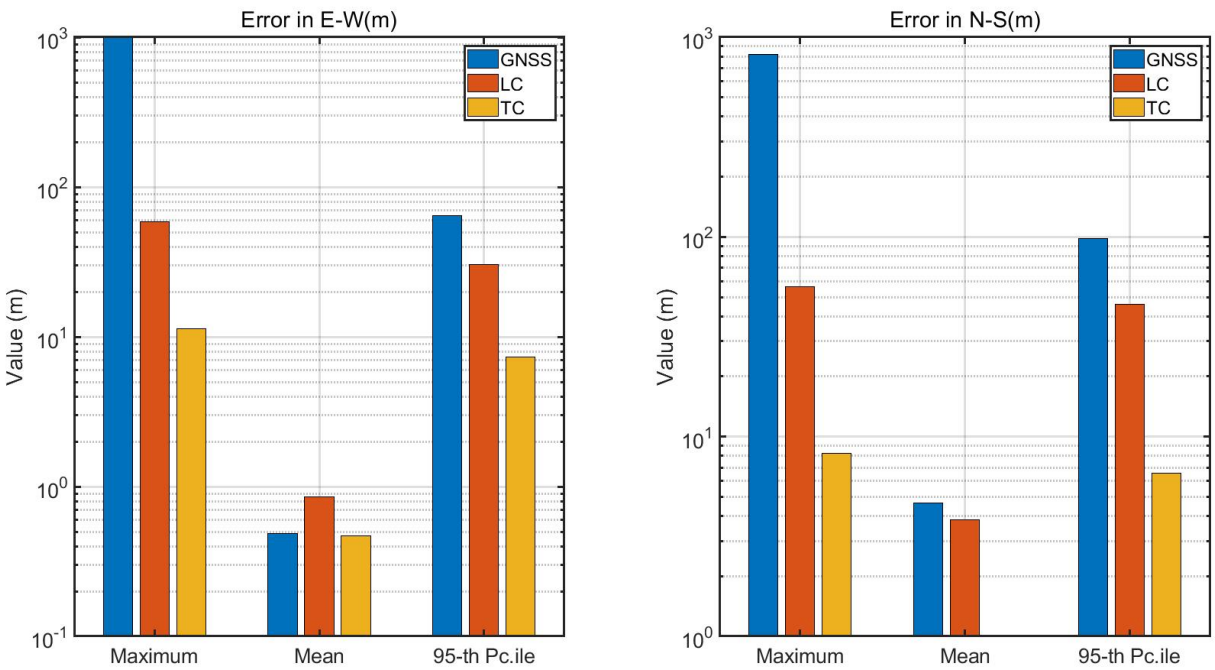
(c)

Figure 5-14 Marunouchi error of 0302 1st test

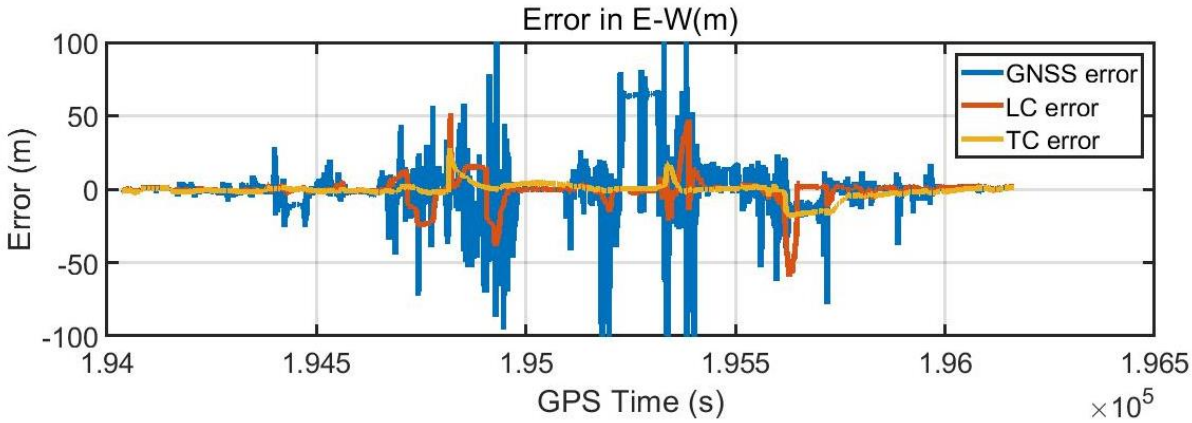
Horizontal positioning errors in 0302 1st dataset. In (a) measurements over time: error in the East direction (left) and error in the North direction (right). In (b) metrics associated to the error in the East direction (above) and the error in the North direction (below). In (c) LC, TC and Poslv yaw angles.

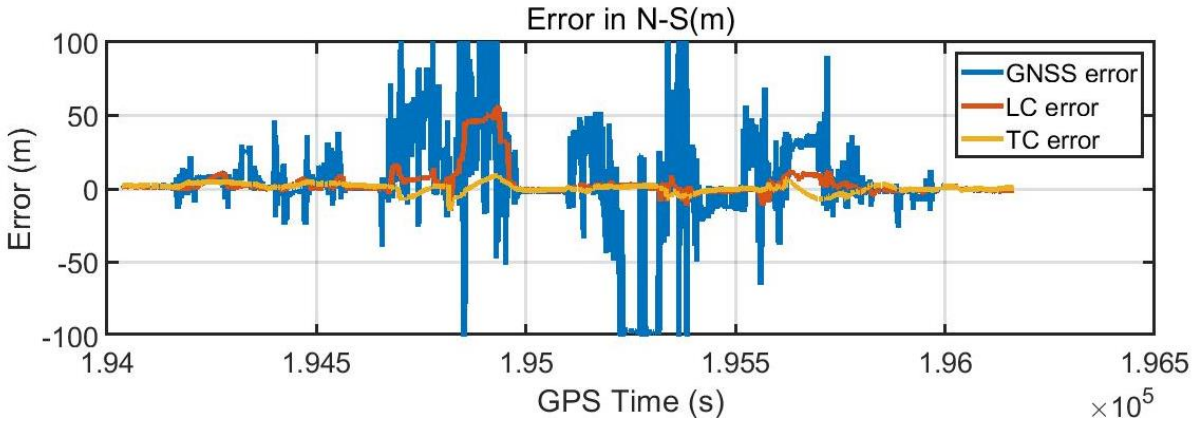
For the 2nd dataset, the RTK FIX rate is about 45.1%. GNSS quality is much worse than the 1st dataset. The error of TC has shifted as a whole. Although GNSS has a large error, the LC can still keep track.

Figure 5-15 Marunouchi track of 0302 2nd test

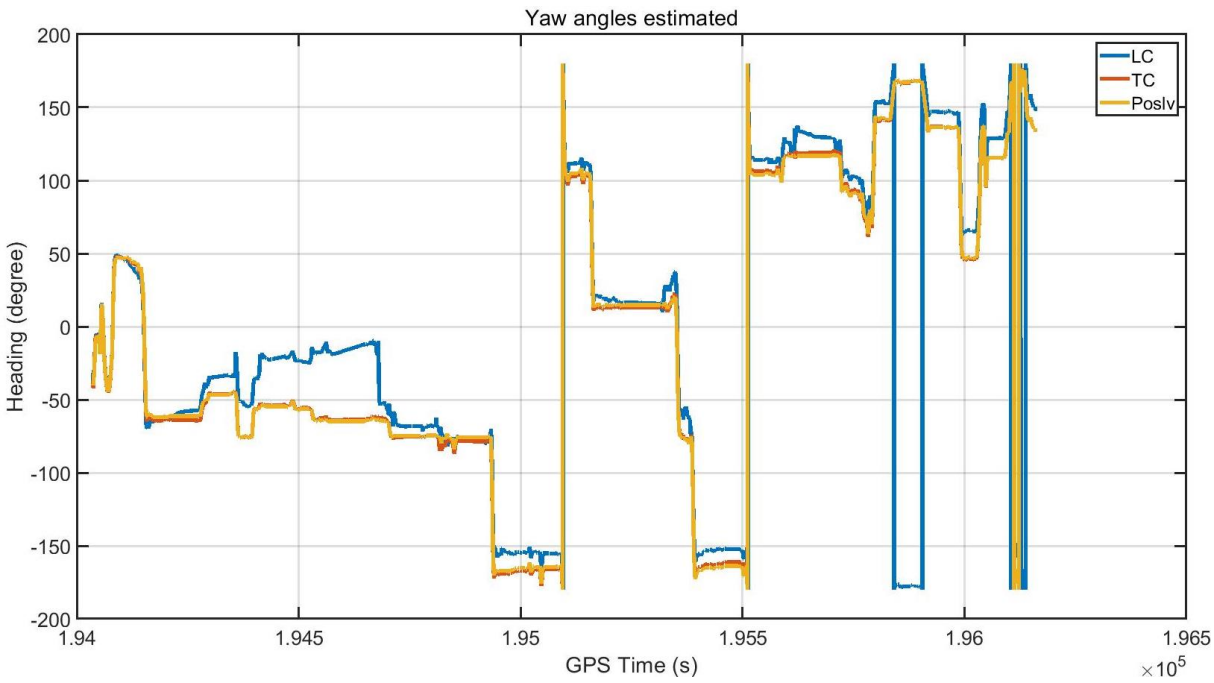


(a)





(b)



(c)

Figure 5-16 Marunouchi error of 0302 2nd test

Horizontal positioning errors in 0302 2nd dataset. In (a) measurements over time: error in the East direction (left) and error in the North direction (right). In (b) metrics associated to the error in the East direction (above) and the error in the North direction (below). In (c) LC, TC and Poslv yaw angles.

Table 5-3 Error of Marunouchi

	UNIT (M)	MAX-E	MAX-N	MEAN-E	MEAN-N	95%-E	95%-N
1ST	GNSS	126.1871	124.9765	1.7869	0.1131	48.5011	40.8056
	TC	29.6913	36.1436	0.5658	0.0224	7.2646	13.8202
	LC	45.1765	68.9323	1.1100	0.5589	16.2394	45.2281
2ND	GNSS	1003.8537	820.0854	0.4863	4.6508	64.6970	97.8188
	TC	11.3592	8.2782	0.4679	1.0031	7.3340	6.5890
	LC	59.0915	56.4772	0.8582	3.8423	30.5077	45.9571

Because of the gross error in GNSS, the average error is lager. For LC, it is easy effected by GNSS position, MEMS IMU without remove the bias intime, the positioning error will accumulate very fast. When the receiver goes out of the urban canyon environment, the error will reduce fast. For TC, it will not be affected by GNSS error easily, but when the error increased, it will take a long time to recognize its estimate error, and reduce the positioning error. According to the twice tests, it is easy to draw the conclusion that, TC is better than LC in urban canyon environment.

5.3 Multi-Sensor Fusion

As chapter 4 shows that, there are some sensors that can replace the GNSS measurements. The vehicle motion model can give the velocity information in vertical. The WSS and yaw angle can give the velocity in horizontal. This information is important for continuous estimation error. As table 5.1 shows, Estelle has lots of sensors available. This chapter will briefly introduce the multi-sensor fusion in GNSS/INS.

Before using the vehicle motion model, it is necessary to convert the navigation frame from ECEF to ENU. This paper using base station position as coordinate origin, and the position unit is meter. Compare with figure 3-3, the difference is that this chapter adds a WSS+YAW Kalman filter.

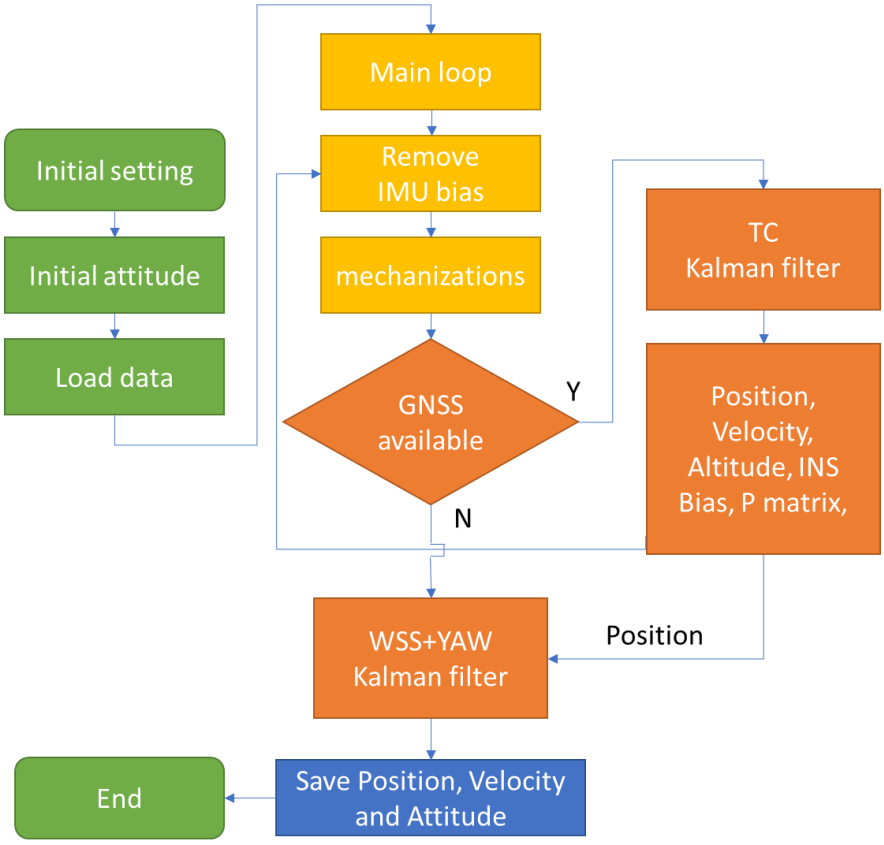


Figure 5-19 Multi-sensor fusion flow chart

For the WSS+YAW, it has accumulated error, so here using Shinonome dataset. The car went through the Shuto Expressway twice, totally 1 minute, and the multi-path error of GNSS is very huge. The F9P-RTK's solution here is not Fix solution, and it is known that the car went straight without changing the vehicle lane. According to the figure 5-20, we can know that the multi-sensor track is better than F9P-RTK.

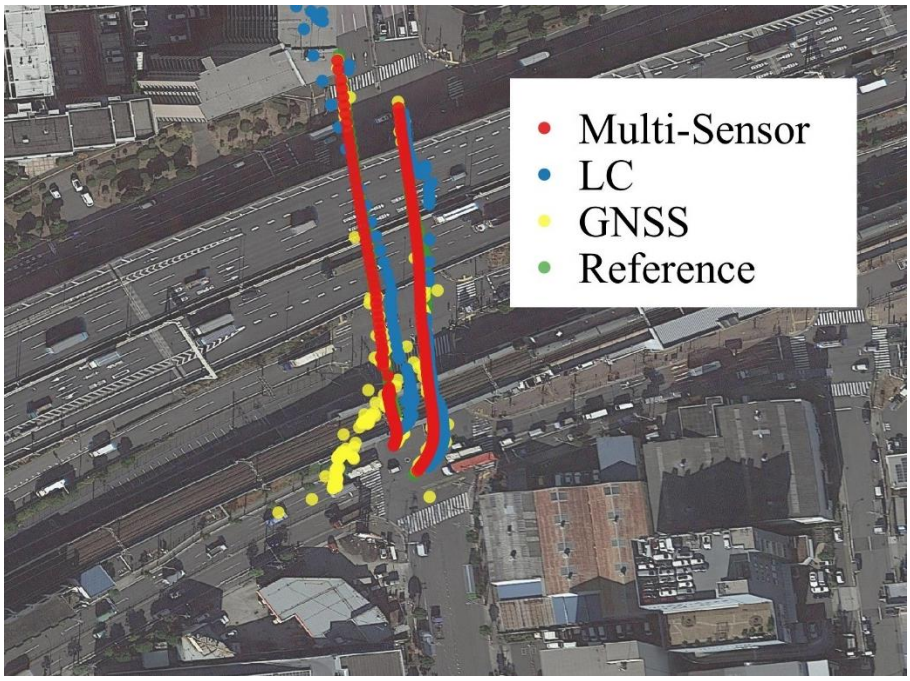
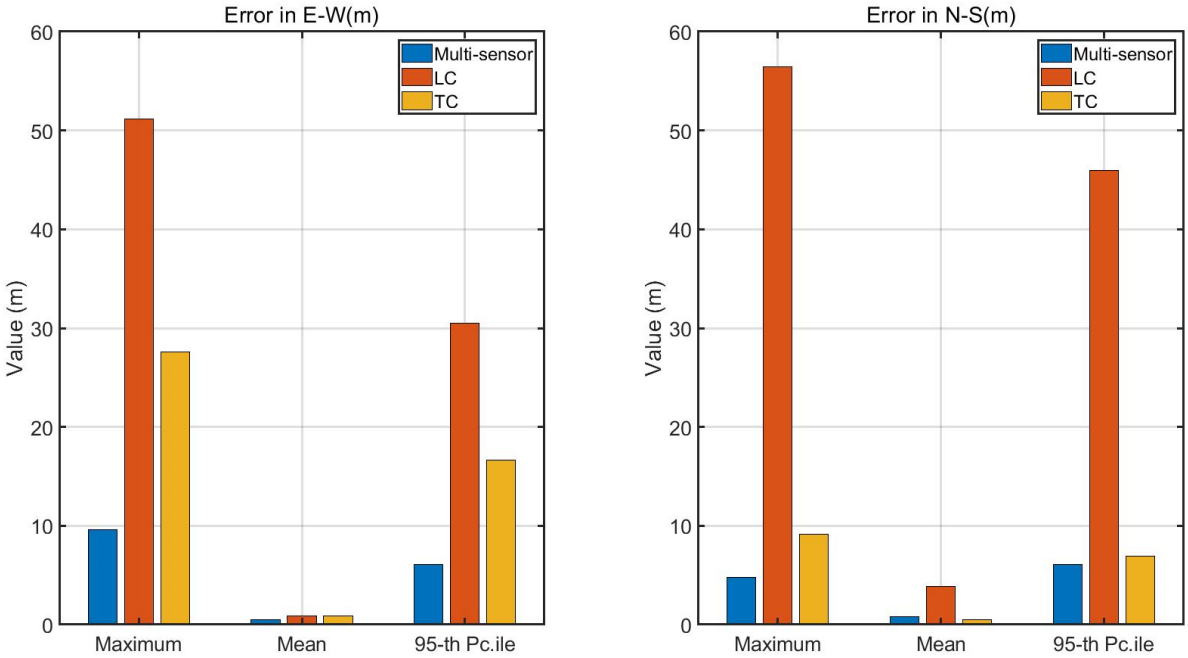
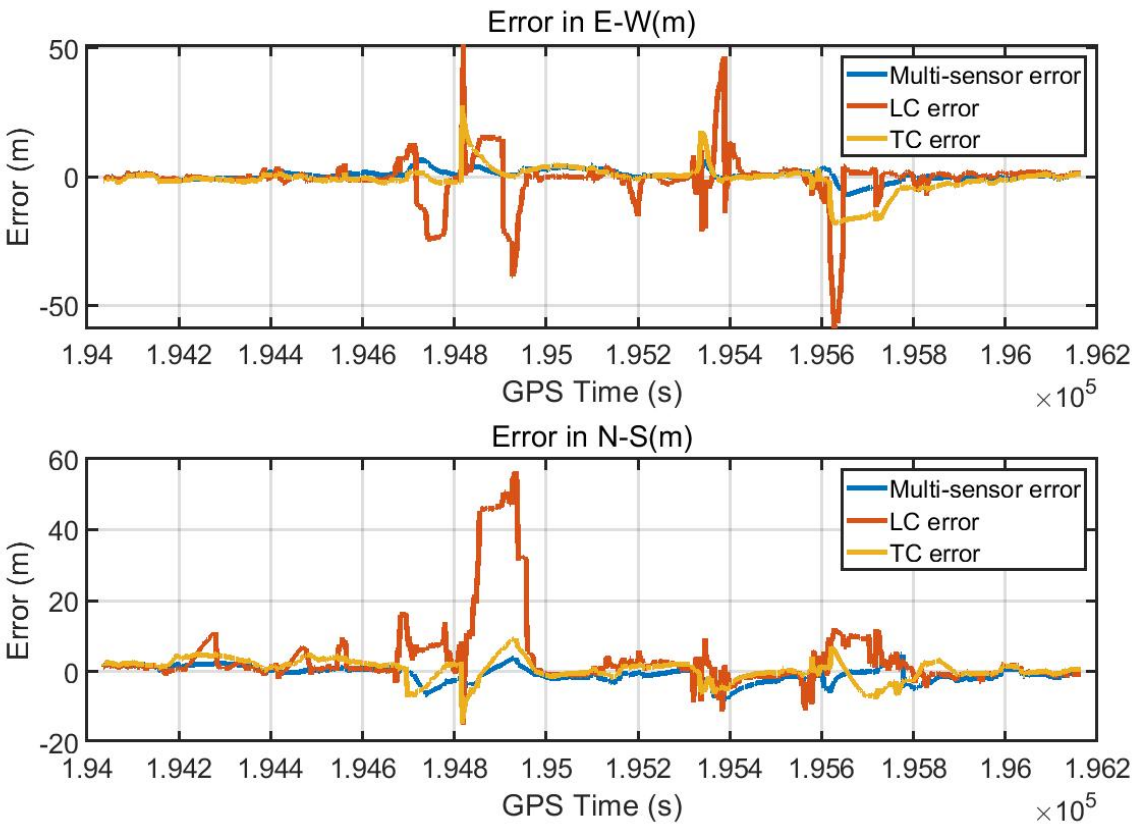


Figure 5-20 Multi-Sensor fusion result in the Shuto Expressway

This paper also did a test in the Marunouchi 2nd dataset, using the TC position result and WSS+YAW dead-reckoning result to do the KF. The results are as follow. According to the result, we can see that the multi-sensor fusion result's maximum error is reduced, and positioning accuracy is improved.



(a)



(b)

Figure 5-21 Multi-Sensor fusion result in the Marunouchi 2nd

Horizontal positioning errors in 0302 2nd dataset. In (a) measurements over time: error in the East direction (left) and error in the North direction (right). In (b) metrics associated to the error in the East direction (above) and the error in the North direction (below).

Table 5-4 Error of different methods

UNIT (M)	MAX-E	MAX-N	MEAN-E	MEAN-N	95%-E	95%-N
GNSS	1003.853	820.0854	0.4863	4.6508	64.6970	97.8188
LC	59.0915	56.4772	0.8582	3.8423	30.5077	45.9571
TC	11.3592	8.2782	0.4679	1.0031	7.3340	6.5890
MULTI-SENSOR	9.6177	8.0312	0.4673	0.8318	6.1176	6.1178

6. CONCLUSION

This paper focuses on the LC and TC for vehicle GNSS/INS integrated navigation, mainly introduces IMU error and GNSS error, INS mechanization equations, deduces and calibrates IMU installation error. By introducing KF, taking IMU attitude, velocity, position, acceleration bias and gyroscope bias as KF estimators, the error model of LC is derived. Based on LC, shows how to replace the KF measurements values from LC into TC. Comparing with the LC source code, the differences between LC and TC algorithms are intuitive shown. Next, this paper introduces some methods for optimize the LC and TC, such as adding the measurements information, or updating the measurement noise covariance. After serious of the vehicle data test and analyzed the results, the conclusions are as follows:

1. In open-sky conidiation, LC mean error and stander deviation are smaller than TC, maximum error is larger than TC;
2. In urban conidiation, LC error is morse than TC;
3. LC is easier affected by GNSS error than TC;
4. The estimate yaw angle form LC is worse than TC;
5. Multi-sensor fusion can reduce the maximum error effectively.

For this master thesis there are several shortages:

- The IMU bias affected by the temperature, but here is no temperature compensate;
- The initial attitude ROLL and PITCH are set as zeros, the attitude error is existing;
- The loosely coupled without anti-error Kalman filter, so the error is affected by the GNSS easily;
- The IMU bias and random walk were settled as a constant value, it is reasonable to do ALLAN variance each time.

For future research：

- The program was written by MATLAB, it is necessary to replace it by C/C++;
- Carrier phase will be included in TC to improve the positioning accuracy;
- Multi-sensor in tightly coupled, which include car speed or GNSS compass;
- Although there are many GNSS/INS sources code in GitHub, few of them has README document to explain their program in detail. This paper will make perfect and opensource to the GNSS/INS beginners.

Reference

[1] P. D. Groves, "Principles of GNSS, inertial, and multisensor integrated navigation systems, [Book review]," *IEEE Aerospace and Electronic Systems Magazine,* vol. 30, no. 2, pp. 26-27, 2015.

[2] O. Montenbruck *et al.*, "IGS-MGEX: preparing the ground for multi-constellation GNSS science," *Inside Gnss,* vol. 9, no. 1, pp. 42-49, 2014.

[3] R. Odolinski, P. J. Teunissen, and D. Odijk, "Combined bds, galileo, qzss and gps single-frequency rtk," *GPS solutions,* vol. 19, no. 1, pp. 151-163, 2015.

[4] Wikipedia. (2021, 2 May). *Satellite navigation*. Available: https://en.wikipedia.org/wiki/Satellite_navigation

[5] H. He, J. Li, Y. Yang, J. Xu, H. Guo, and A. Wang, "Performance assessment of single-and dual-frequency BeiDou/GPS single-epoch kinematic positioning," *GPS solutions,* vol. 18, no. 3, pp. 393-403, 2014.

[6] D. Li, R. Landry Jr, and P. Lavoie, "Low-cost MEMS sensor-based attitude determination system by integration of magnetometers and GPS: A real-data test and performance evaluation," in *Proceedings of IEEE/ION PLANS 2008*, 2008, pp. 1190-1198.

[7] W. Li and J. Wang, "Effective adaptive Kalman filter for MEMS-IMU/magnetometers integrated attitude and heading reference systems," *The Journal of Navigation,* vol. 66, no. 1, pp. 99-113, 2013.

[8] Y. Wu, C. Goodall, and N. El-Sheimy, "Self-calibration for IMU/odometer land navigation: Simulation and test results," in *Proceedings of the 2010 International Technical Meeting of The Institute of Navigation*, 2010, pp. 839-849.

[9] H. Lan, M. Elsheikh, W. Abdelfatah, A. Wahdan, and N. El-Sheimy, "Integrated RTK/INS navigation for precision agriculture," in *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, 2019, pp. 4076-4086.

[10] M. F. Abdel-Hafez, K. Saadeddin, and M. A. Jarrah, "Constrained low-cost GPS/INS filter with encoder bias estimation for ground vehicles′ applications," *Mechanical Systems and Signal Processing,* vol. 58, pp. 285-297, 2015.

[11] A. Noureldin, T. B. Karamat, M. D. Eberts, and A. El-Shafie, "Performance enhancement of MEMS-based INS/GPS integration for low-cost navigation applications," *IEEE Transactions on vehicular technology,* vol. 58, no. 3, pp. 1077-1096, 2008.

[12] C. Eling, L. Klingbeil, and H. Kuhlmann, "Real-time single-frequency GPS/MEMS-IMU attitude determination of lightweight UAVs," *Sensors,* vol. 15, no. 10, pp. 26212-26235, 2015.

[13] G. Welch and G. Bishop, "An introduction to the Kalman filter," 1995.

[14] L. Wang, G. Libert, and P. Manneback, "Kalman filter algorithm based on singular value decomposition," in *[1992] Proceedings of the 31st IEEE Conference on Decision and Control*, 1992, pp. 1224-1229: IEEE.

[15] G. J. Bierman, *Factorization methods for discrete sequential estimation*. Courier Corporation, 2006.

[16] H. Jaeger, *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach* (no. 01). GMD-Forschungszentrum Informationstechnik Bonn, 2002.

[17] K. Xiong, H. Zhang, and C. Chan, "Performance evaluation of UKF-based nonlinear filtering,"

Automatica, vol. 42, no. 2, pp. 261-270, 2006.

[18] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[19] G. Falco, M. Pini, and G. Marucco, "Loose and Tight GNSS/INS Integrations: Comparison of Performance Assessed in Real Urban Scenarios," *Sensors (Basel)*, vol. 17, no. 2, Jan 29 2017.

[20] M. Emerson Cavalheri. (2019). *gnssins*. Available: https://github.com/marcoamm/gnssins

[21] J. L. Su. (2019). *ignav*. Available: https://github.com/Erensu/ignav

[22] X. Liu. (2018). *TightlyCoupledINSGNSS*. Available: https://github.com/benzenemo/TightlyCoupledINSGNSS

[23] K. Chen. (2021). *GINav*. Available: https://github.com/kaichen686/GINav

[24] T. Takasu. (2006). *RTKLIB (2.4.3 b34 ed.)*. Available: http://www.rtklib.com/

[25] S. Gleason, D. Gebre-Egziabher, and D. G. Egziabher, "GNSS applications and methods," 2009.

[26] W. Wen, L.-T. Hsu, and G. Zhang, "Performance analysis of NDT-based graph SLAM for autonomous vehicle in diverse typical driving scenarios of Hong Kong," *Sensors*, vol. 18, no. 11, p. 3928, 2018.

[27] Wikipedia. (2021, 22 June). *Earth-centered inertial*. Available: https://en.wikipedia.org/wiki/Earth-centered_inertial

[28] D. Tedaldi, A. Pretto, and E. Menegatti, "A robust and easy to implement method for IMU calibration without external equipments," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 3042-3049: IEEE.

[29] MathWorks. (2021). *Calibrate Accelerometer*. Available: https://www.mathworks.com/help/supportpkg/arduinoio/ug/calibrate-sensors.html

[30] X. Yang. (2021). *nav_matlab*. Available: https://github.com/yandld/nav_matlab

[31] C.-W. Park and J. P. How, "Method and apparatus for selecting optimal satelittes in global positioning system," ed: Google Patents, 2004.

[32] T. Li *et al.*, "P 3-LOAM: PPP/LiDAR Loosely Coupled SLAM With Accurate Covariance Estimation and Robust RAIM in Urban Canyon Environment," *IEEE Sensors Journal*, vol. 21, no. 5, pp. 6660-6671, 2020.

[33] W. Stempfhuber and M. Buchholz, "A precise, low-cost RTK GNSS system for UAV applications," *Proc. of Unmanned Aerial Vehicle in Geomatics, ISPRS*, 2011.

[34] X. L. Guo. (2021). *iXR_GNSS-IMU_TightlyCouplingProgram*. Available: https://github.com/kakusang2020/iXR_GNSS-IMU_TightlyCouplingProgram

[35] Y. Yang, L. Song, and T. Xu, "Robust estimator for correlated observations based on bifactor equivalent weights," *Journal of geodesy*, vol. 76, no. 6, pp. 353-358, 2002.

[36] A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, "Vehicle trajectory prediction based on motion model and maneuver recognition," in *2013 IEEE/RSJ international conference on intelligent robots and systems*, 2013, pp. 4363-4369: IEEE.

[37] M. Ilyas, Y. Yang, Q. S. Qian, and R. Zhang, "Low-cost IMU/odometer/GPS integrated navigation aided with two antennae heading measurement for land vehicle application," in *2013 25th Chinese Control and Decision Conference (CCDC)*, 2013, pp. 4521-4526: IEEE.

[38] H. Liu, Z. Wang, S. Fang, and C. Li, "MEMS Based SINS/OD Filter for Land Vehicles' Applications," *Mathematical Problems in Engineering*, vol. 2017, pp. 1-9, 2017.

[39] vignesh. (2021). *Rotary Encoder Working Principle*. Available: https://instrumentationforum.com/t/rotary-encoder-working-principle/6781

[40] M. N. Berberan-Santos, E. N. Bodunov, and L. Pogliani, "On the barometric formula," *American Journal of Physics*, vol. 65, no. 5, pp. 404-412, 1997.

Acknowledgment

This is my first time studying abroad. Due to the COVID-19, I brought a lot of trouble to Prof. Kubo. Thank you very much for your tolerance and patient guidance. Thank you for taking the trouble to collect a lot of interesting data, a large number of data tests enrich my paper, but also verify the positioning accuracy of the algorithm. GNSS / INS is a subject that emphasizes the combination of knowledge and practice. It needs to be carefully prepared when collecting data. In processing data, it is to ensure that no new error is introduced in each step. I have learned a lot in the past two years. It's my pleasure to study in this laboratory

Thanks, Mr. Ozeki, for your warm answers and help in life. Thanks, Mr. Komatsu, for discussing problems with me. Some seemingly complex problems will become clear after communication. Thank you, Mr. Kobayashi, for your technical guidance. I'm glad to have you as an excellent schoolmate. Thanks to Yoshihara's care for me, the occasional chat eased a lot of my depression.

I would like to thank those engineers and teachers who are willing to share their code on the Internet. Code is the closest to the truth, without those meticulous formula derivation, detailed code comments and enthusiastic Q&A, I think it is difficult for me to get into this field I have never touched. Thanks to Mr. Takasu's rtklib, Prof. Paul Groves’ book and source code, Xiao Liu's GNSS/INS tightly coupled code, Tao Li's single point positioning code and enthusiastic communication, Xi Yang's rich INS toolbox and Yize Zhang's explains. My GNSS / INS level is limited, but without the help of the above predecessors, it is difficult for me to write this paper. Thank you very much again.

I haven't been home for nearly two years, and I miss my relatives in China very much. Thank you very much for your care and support these days. Thanks to my girlfriend for understanding me academically, allowing me to continue studying in my own research field with peace of mind.

Finally, I would like to thank Tokyo University of Marine Science and Technology and Shanghai Maritime University for giving me this opportunity to study abroad, which has given me a different life as a graduate student and the opportunity to study with outstanding students. Thanks to the subsidy of the research laboratory and the scholarship support provided by the Japanese government, I can study in the laboratory without worry about the cost of living.

There is no end to learning. I am willing to forge ahead in the future!