

KF-GINS

[\[中\]](#) [\[EN\]](#)

An EKF-Based GNSS/INS Integrated Navigation System

Introduction

We open-source KF-GINS ¹, an EKF-based GNSS/INS integrated navigation system. KF-GINS implements the classical integrated navigation solution of GNSS positioning results and IMU data. The KF-GINS follows the course content of "Inertial Navigation Principles and GNSS/INS Integrated Navigation" by Prof. Xiaoji Niu and Dr. Qijin Chen, as a supporting resource for the course. The software is developed in C++ programming language, and CMake is used to manage the project.

The main features of KF-GINS are as follows:

- A GNSS/INS loosely-coupled integrated navigation algorithm based on the extended Kalman filter architecture (error state vector), including IMU error compensation, inertial navigation solution, Kalman filter, error feedback, etc.
- 21-dimensional error states are defined in the system, including position errors, velocity errors, attitude errors, IMU bias errors, and IMU scale factor errors
- The attitude errors adopt the Phi-angle model, and the velocity errors and position errors are defined in the navigation frame.
- The inertial navigation solution adopts the two-sample mechanization algorithm based on the linear assumption, which compensates for the second-order coning correction, the rotational motion, and the sculling motion.

Organization: [Integrated and Intelligent Navigation \(i2Nav\) Group](#), GNSS Research Center, Wuhan University.

Related Reference:

- 牛小骥, 陈起金, "[惯性导航原理与GNSS/INS组合导航课程讲义](#)", 武汉大学多源智能导航实验室, 2022
- 牛小骥, 陈起金, "[惯性导航原理与GNSS/INS组合导航课程视频](#)", 武汉大学多源智能导航实验室, 2022
- X. Niu, Q. Zhang, L. Gong, C. Liu, H. Zhang, C. Shi, J. Wang and M. Coleman (2014). "Development and evaluation of GNSS/INS data processing software for position and orientation systems." Survey Review 2014; 47(341), 87-98.
- 严恭敏, 翁浚, 捷联惯导算法与组合导航原理. 西北工业大学出版社, 2019.
- E.-H. Shin, "Estimation techniques for low-cost inertial navigation," Ph.D. dissertation, Dept. Geomatics Eng., Univ. of Calary, AB, Canada, 2005
- [Savage, P.G.](#), "Strapdown analytics, Part I", Maple Plain, MN: Strapdown Associates, 2000.
- Titterton David, JohnL. Weston, and John Weston, "Strapdown inertial navigation technology", The Institution of Electrical Engineers, 2004.
- P. D. Groves, "Principles of GNSS, Inertial, and Multi-sensor Integrated Navigation Systems", 2nd ed., vol. 39. Artech House, 2013.

If you use this software for your academic research, please give acknowledgment as follows and cite our [related document and papers](#)

- 1 English version: "The authors would like to acknowledge the team of Prof. Xiaoji Niu of the Integrated and Intelligent Navigation (i2Nav) group from GNSS Research Center of Wuhan University for providing the open-source KF-GINS software that was used in the paper."
- 2 中文模板: "本文作者感谢武汉大学卫星导航定位技术研究中心多源智能导航实验室(i2Nav)牛小骥教授团队开源的KF-GINS软件平台。"

Contacts:

- For any technique problem, you can send an email to Liqiang Wang (wlg@whu.edu.cn) or Dr. Hailiang Tang (thl@whu.edu.cn).
- For Chinese users, we also provide a QQ group (481173293) for discussion. You are requested to provide your organization and name to join the QQ group.

1 Program Compilation and Execution

1.1 Compilation environment

KF-GINS is managed by CMake and can be compiled on both Linux and Windows. We recommend you choose Linux environment first.

The configuration file is used as a parameter to run KF-GINS after successfully compiling it. To debug the program, it is also required to add the configuration file as a parameter.

1.2 Dependency libraries

KF-GINS requires another three libraries, which are Eigen3, abseil-cpp, and yaml-cpp. We have appended them as the thirdparty libraries in the **ThirdParty** directory.

1.3 Compile under Linux

We recommend you use g++ compiler of Ubuntu18.04 or Ubuntu20.04 to compile KF-GINS. You should install the build-essential libraries following the commands:

```
1 sudo apt-get install cmake
2 sudo apt-get install build-essential
```

After preparing your own compilation environment, you can clone the repository locally and compile KF-GINS as follows:

```
1 # Clone the repository
2 git clone https://github.com/i2Nav-WHU/KF-GINS.git ~/
3
4 # Build KF-GINS
5 cd ~/KF-GINS
6 mkdir build && cd build
7
8 cmake ../ -DCMAKE_BUILD_TYPE=Release
9 make -j8
10
11 # Run demo dataset
12 cd ~/KF-GINS
13 ./bin/KF-GINS ./dataset/kf-gins.yaml
14
15 # Wait until the program finish
```

1.4 Compile under Windows

The MSVC(Microsoft Visual C/C++) compiler is generally used in Windows. We recommend you to compile KF-GINS in the VSCode software.

You should first install the MSVC compiler and VSCode software (including the necessary plug-ins, such as C/C++, C/C++ Extension Pack, CMake, and CMake Tools).

After preparing your own compilation environment, you can clone the repository locally and open the KF-GINS folder in VSCode:

- Set compiler: open the Command Palette (Ctrl+Shift+P) and type "CMake: Select a Kit", select the MSVC compiler
- Set compile parameter: type "CMake: Select Variant" in the Command Palette, select "Release"
- Configure CMake: type "CMake: Configure" in the Command Palette
- Compile Project: type "CMake: Build" in the Command Palette

Open a PowerShell or CMD terminal in the project directory and run the test dataset:

```
1 | .\bin\KF-GINS.exe .\dataset\kf-gins.yaml
2 | # The executable file may be generated in the '.\bin\Release' directory. Then
   | the command is:
3 | # .\bin\Release\KF-GINS.exe .\dataset\kf-gins.yaml
```

2 Use KF-GINS

2.1 Prerequisite Knowledge

Frame defination:

- IMU frame: the origin is the IMU center, the three axes point forward-right-down
- Navigation reference frame: the origin coincides with the IMU frame, the three axes point north-east-down

Navigation state:

- position: the geodetic coordinates of the IMU position in the Earth frame (latitude-longitude-ellipsoid height)
- velocity: the IMU velocity to the Earth projected in the navigation reference frame (north speed-east speed-down speed)
- attitude: IMU attitude to navigation frame (quaternion, direction cosine matrix, or euler angles. euler angles are defined as yaw-pitch-roll, ZYX rotation order)

IMU noise model:

- IMU measurement noises are modeled as the Gaussian white noise
- IMU bias errors are modeled as the first-order Gauss-Markov process
- IMU scale factor errors are modeled as the first-order Gauss-Markov process

For more details on the algorithm, please refer to [惯性导航原理与GNSS/INS组合导航课程讲义](#).

2.2 Data format

- The IMU text file format is defined as:

Columns	Data description	Units
1	GNSS seconds of week	<i>s</i>
2~4	X-Y-Z axes incremental angles	<i>rad</i>
5~7	X-Y-Z axes incremental velocity	<i>m/s</i>

- The GNSS position text file format is defined as:

Columns	Data description	Units
1	GNSS seconds of week	<i>s</i>
2	latitude	<i>deg</i>
3	longitude	<i>deg</i>
4	ellipsoid altitude	<i>m</i>
5~7	position STD (north-east-down)	<i>m</i>

- The text file format of the navigation result and the ground-truth is defined as:

Columns	Data description	Units
1	GNSS week	-
2	GNSS seconds of week	<i>s</i>
3	latitude	<i>deg</i>
4	longitude	<i>deg</i>
5	ellipsoid altitude	<i>m</i>
6~8	3-D velocity (north-east-down)	<i>m/s</i>
9~11	attitude angles (roll-pitch-yaw)	<i>deg</i>

- The IMU error binary file format(all double data) is defined as:

Columns	Data description	Units
1	GNSS seconds of week	<i>s</i>
2~4	X-Y-Z axes gyroscope biases	<i>deg/h</i>
5~7	X-Y-Z axes accelerometer biases	<i>mGal</i>
8~10	X-Y-Z axes gyroscope scale factors	<i>ppm</i>
11~13	X-Y-Z axes accelerometer scale factors	<i>ppm</i>

- The state STD binary file format(all double data) is defined as:

Columns	Data description	Units
1	GNSS seconds of week	<i>s</i>
2~4	3-D position STD (north-east-down)	<i>m</i>
5~7	3-D velocity STD (north-east-down)	<i>m/s</i>
8~10	3-D attitude STD (roll-pitch-yaw)	<i>deg</i>
11~13	X-Y-Z axes gyroscope bias STD	<i>deg/h</i>
14~16	X-Y-Z axes accelerometer bias STD	<i>mGal</i>
17~19	X-Y-Z axes gyroscope scale factor STD	<i>ppm</i>
20~22	X-Y-Z axes accelerometer scale factor STD	<i>ppm</i>

2.3 Initial align

KF-GINS only supports initial alignment given all initial states currently. The initial states need to be set in the configuration file (kf-gins.yaml) before executing the program.

3 Datasets

3.1 Demo dataset

We offer a demo dataset with the configuration file, which is located in the **dataset** directory.

3.2 awesome-gins-datasets

Users can find our open-sourced datasets at [awesome-gins-datasets](#).

3.3 Your own dataset

The data formats used in KF-GINS are the same as the formats defined at [awesome-gins-datasets](#). You can follow the formats to prepare your own datasets.

4 License

The source code is released under GPLv3 license.

We are still working on improving the code reliability. For any technical issues, please contact Liqiang Wang (wlq@whu.edu.cn) or Hailiang Tang (thl@whu.edu.cn), or open an issue at this repository.

For commercial usage, please contact Prof. Xiaoji Niu (xjniu@whu.edu.cn).

1. We also open-sourced [OB_GINS](#), an optimization-based GNSS/INS integrated navigation system. [↪](#)