

Strapdown Inertial Navigation System Based on an IMU and a Geomagnetic Sensor

By **Joel Li** and **Van Yang**

Abstract

This article introduces a strapdown inertial navigation system (SINS) we built using Analog Device's inertial measurement unit (IMU) sensor [ADIS16470](#) and PNI's geomagnetic sensor RM3100. Some basic processes for SINS based on magnetics, angular rates, and gravity (MARG) are implemented, including electromagnetic compass calibration, an attitude and heading reference system (AHRS) that use an extended Kalman filter (EKF), and trace tracking. Loosely coupling sensor fusion by using the minimum squared error (MSE) method is also implemented. Both algorithm and experiment setup in each process step are shown. The result analysis and the method used to improve accuracy are discussed at the end of the article.

Introduction

With the market and technique growth of service robots, navigation has become a hot research and application point. Compared to vehicles, ships, or aircraft, service robots are small and low cost, so their navigation systems should be strapdown and low cost. Traditional, stable-platform navigation systems commonly involve separate accelerators and fibers or laser-based gyroscopes, with all the components mechanically and rigidly mounted on a stable platform that is isolated from the moving vehicle. This leads to the drawbacks of large size, poor reliability, and high cost. In contrast, in strapdown navigation systems, the inertial sensors are fastened directly to the vehicle's body, which means the sensors rotate together with the vehicle. This strapped down method eliminates the drawbacks of stable platform navigation. However, the accuracy of platform navigation is often higher than SINS. Platform navigation can often reach the strategic grade (0.0001°/hr gyroscope bias, 1 μ g accelerator bias) or military grade (0.005°/hr gyroscope bias, 30 μ g accelerator bias), while most SINS can only reach the navigation grade (0.01°/hr gyroscope bias, 50 μ g accelerator bias) or tactical grade (10°/hr gyroscope bias, 1 mg accelerator bias). For most service robot or AGV navigation applications, this is enough.

There are multiple navigation methods, including machine vision, GPS, UWB, lidar with SLAM, etc. While inertial navigation is always an important component in navigation, where an IMU is used. However, due to the limitation of this kind of sensor—such as bias error, cross-axis error, noise, and especially bias instability—inertial navigation often needs a partner sensor to give it a reference or calibration periodically, which is called sensor fusion here. There are many sensors to fuse with an IMU, such as cameras and odometers, but among these sensors, a geomagnetic sensor is a low cost way to get attitude together with an IMU.

In this article, we use ADI's IMU, ADIS16470, and a geomagnetic sensor to develop a platform and an algorithm to implement a strapdown inertial

navigation system. However, the sensor can only provide the attitude information. For the dead reckoning or distance measurement, we can only use the acceleration sensors in the IMU.

ADIS16470 IMU Introduction

ADI's ADIS16470 is a miniature MEMS IMU that integrates a 3-axial gyroscope and a 3-axial accelerometer. It supplies 8°/hr bias stability for the gyroscope and 13 μ g bias stability for the accelerometer, while its key parameters are factory calibrated. Also, ADIS16470's low cost price is attractive in the same level parts and is widely used by many customers. In this article, we use a microcontroller to communicate with ADIS16470 with SPI interface.

Geomagnetic Sensor Introduction

A geomagnetic sensor is a sensor used to measure the geomagnetic field in the compass' body coordinates (that is, frame), which provides an absolute reference for the heading. Its x, y, and z component values are projected from the local geomagnetic field. There are two key drawbacks to this kind of sensor—one is that its accuracy and resolution cannot be high—for example, the commonly used compass sensor HMC5883L from Honeywell only has a 12-bit resolution. The other drawback is that the sensor is easily interfered with by its surrounding environment, since the geomagnetic field is very weak, from milligauss to 8 gauss.

Despite these drawbacks, it can still be used in many circumstances, such as open field, low EMI environments, etc. By loosely coupling the geomagnetic sensor to the IMU, we can use it in most environments.

In this article, we use a high performance compass sensor, RM3100 from PNI Sensor Corporation, that supplies a 24-bit resolution. PNI uses the active excitation method to improve the immunity from noise.

Calibration of Compass Sensor

Before using a compass sensor, it needs to be calibrated to eliminate two main errors. One is offset error, which is originally caused by the sensor's and the circuit's offset error. The other is scale error. Both errors are easily interfered with by the surrounding magnetic environment. For example, if there is an x-axis direction external magnetic field applied to the sensor, it will give out an external x-axis offset error. Simultaneously, the x-axis scale will also be different to the y- and z-axis.

The method commonly used to calibrate a magnetic sensor is to rotate the sensor as a circle in an x-y plane and then draw out the data. The geomagnetic field strength at a place is a constant value, so the data drawn should be a circle; however, we will, in fact, see an oval, which means we need to move and rescale the oval to a circle centered at zero.

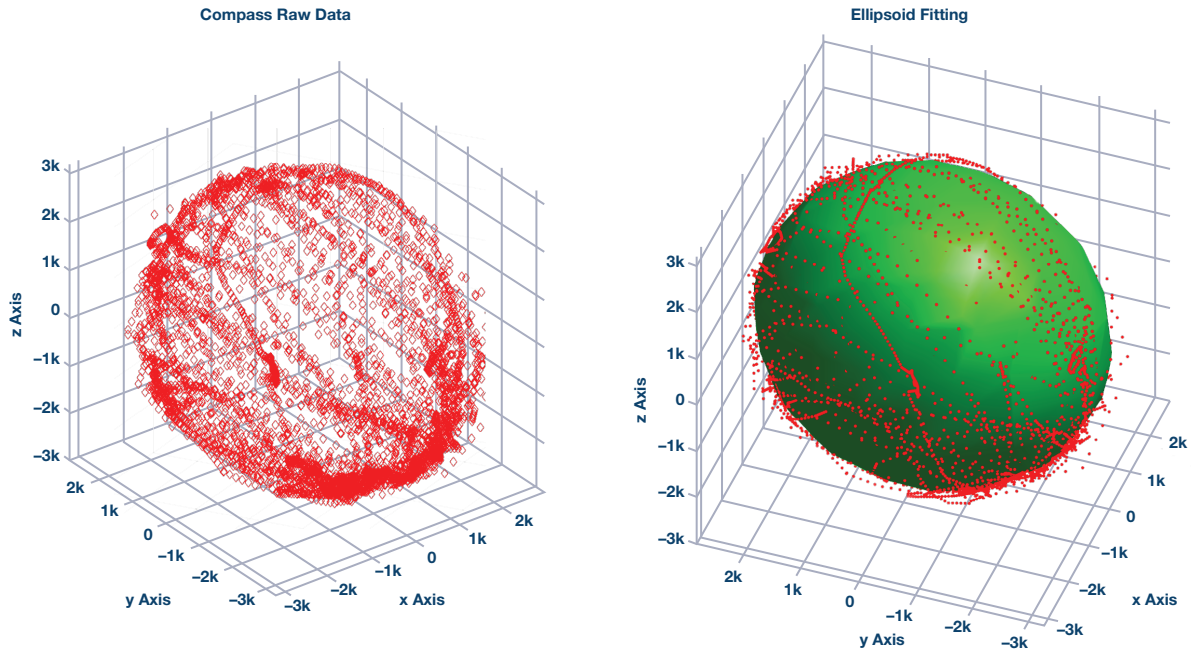


Figure 1. The raw compass data distribution (left) and compass data after the usage of the ellipsoid fitting (right).

The method of 2D calibration mentioned above has some drawbacks and requires an accelerator to measure its inclination. We use a 3D spherical fitting method to calibrate the compass sensor. First, we need to rotate the sensor to every direction in x-y-z space and draw its values in a 3D coordinate. Then we need to fit the data as an ellipsoid using the minimum squared error (MSE) method.

An ellipsoid equation can be expressed as

$$aX^2 + bY^2 + cZ^2 + 2fXY + 2gXZ + 2hYZ + 2pX + 2qY + 2rZ + d = 0 \quad (1)$$

where X, Y, and Z are the geomagnetic components of the compass output in its three directions. Fitting these values to an ellipsoid means we need to get a best value set of the coefficients. We define the coefficients as:

$$\sigma = [a \ b \ c \ f \ g \ h \ p \ q \ r \ d]^T$$

when fitting, we define vector as:

$$\beta = [X^2 \ Y^2 \ Z^2 \ 2XY \ 2XZ \ 2YZ \ 2X \ 2Y \ 2Z]^T$$

so we need to calculate a best σ , and use Equation 2 to find the minimum value:

$$\min (\sigma^T \beta^T \beta \sigma) \quad (2)$$

This way we can get the fitting result shown in Figure 1.

To calibrate the sensor, we need to stretch and move the fitted ellipsoid to a sphere centered at zero. We use a matrix singular value decomposition (SVD) method to do this calibration. The postcalibration sphere is shown in Figure 2.^{1,2}

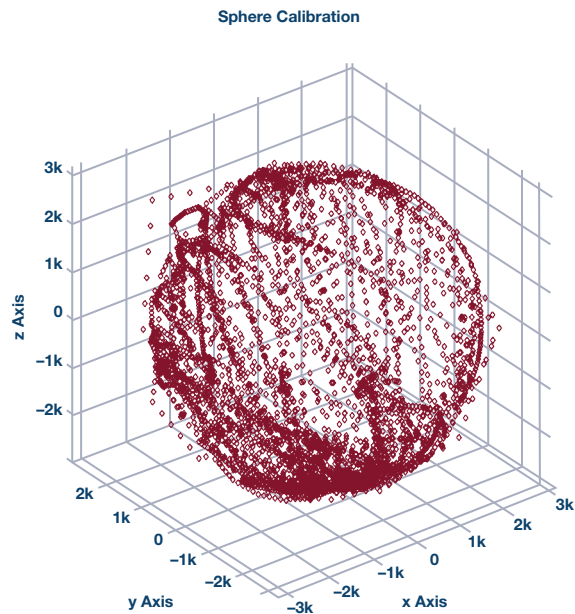


Figure 2. The compass data after sphere calibration with the SVD method.

After calibration, we can see the measured magnetic field strength (sphere radius) will keep to a nearly constant value, as shown in Figure 3.

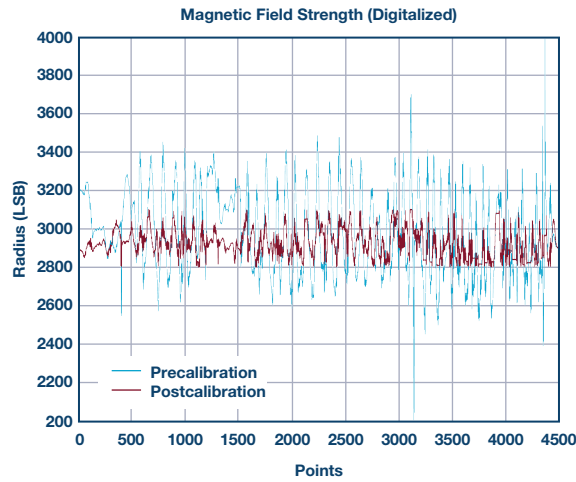


Figure 3. Magnetic field comparison between precalibration and postcalibration.

Attitude and Heading Reference System Using ADIS16470 and a Compass

An AHRS consists of sensors on three axes that provide attitude information, including roll, pitch, and yaw. AHRS is a concept from aircraft navigation. We use it to describe the orientation; that is, attitude.

Before introducing our method, it is necessary to explain the reason for fusion to find the attitude first. In fact, there are three kinds of sensors in our system now: gyroscope, accelerator, and compass.

The gyroscope gives out the angular rate of rotation around each axis. By integrating the angular rate, we can find the rotation angle. If we know the initial heading, we will always get the heading attitude. Because of integration, the gyroscope's instable bias will be accumulated, which will lead to an angle error. Besides, the Gaussian distributed noise from the gyroscope will be integrated to a Brownian moving process and lead to random walking error. Thus, it is difficult for us to use a gyroscope for a long time, and gyroscopes need to be calibrated periodically.

The accelerator provides the acceleration rate of movement in each axis direction. In static status, we can get the angle between each axis and gravity acceleration rate. Because the gravity acceleration is constant in direction and value, we can get the heading attitude relative to the direction of gravity. However, this method uses gravity acceleration as a reference, so the angle rotating around gravity acceleration cannot be solved.

The compass provides the value in each axis projected from the geomagnetic field. We can derive the angular value from the relation between each axis and the geomagnetic field direction, which is also a constant vector. As mentioned in the previous section, because of poor immunity to the external magnetic field, the compass needs a low interference environment.

From this explanation, we can see it is difficult to rely on only one sensor to find the attitude, and that we need to use two or three sensors in combination and fuse the information. This article involves accelerators, gyroscopes, and geomagnetic compasses to find the attitude. This kind of fusion is called the magnetic, angular rate, and gravity (MARG) system.

Extended Kalman Filter Design and Sensor Fusion

There are multiple methods to fuse IMU and compass data together, such as complementary filters, statistics ARMA filters, Kalman filters, and more. We use an extended Kalman filter in this article.

First, we need to introduce some definitions used in this article.

Coordinate Definition

The heading, or orientation, is the relation between two coordinates (that is, frames). One coordinate is always changing, the other remains constant. For a coordinate definition method, we use navigation coordinates and body coordinates. As opposed to north-east-down (NED) or geographic methods, we define the initial body coordinate value measured as the navigation coordinate, which is a constant coordinate afterward. The mapping (projecting) matrix from the body coordinate to the navigation coordinate is defined as

$$C_b^n$$

Attitude Definition

Different from the Euler angles or direction cosine matrix (DCM) matrix, we use quaternions here, defined as

$$q$$

which is commonly used in navigation to avoid argument.³

Attitude Updating Using Kalman Filter

The kinematics equation, that is, the state transition equation, we use in this article is a deviation format, which is not linear, so we need to use an EKF with first-order linear approximation to the deviation equation. For EKF designs, we define

$$x = [\omega \ q]^T$$

a 1×7 vector as a state variable, where

$$\omega = [\omega_x \ \omega_y \ \omega_z]^T$$

is the angular rate, and

$$q = [a \ b \ c \ d]^T$$

is the attitude quaternion.

$$z = [\omega \ q]^T$$

a 1×7 vector, is the observation variable, which has the same component as the state variable.

$$A = \left[\frac{1}{T} (\omega_k - \omega_{k-1}) \quad \frac{1}{2} q_{k-1} \omega_{k-1} \right]$$

a 7×7 matrix, is the state transition matrix, where the first part in A is the digitalized differential equation of the angular rate, and the second part is the digitalized quaternion updating equation, which is derived from the kinematics equation.

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

a 7×7 matrix, is the observation matrix.

$$P = E[e^T e]$$

is the error covariance matrix, a 7×7 matrix, where

e

is the error vector estimated, \hat{x} , from the real, x . We set the initial error to a relatively small value in testing. It will converge to a small value automatically.

R, Q

are set as the noise covariance of state transition noise and observation noise. We get their initial value,

R_0

and

Q_0

by measuring the square of ac rms value of the gyroscopes and the accelerators. While keeping the IMU and compass in static status, we set

$$R_0 = Q_0$$

With this definition, our Kalman filter will be accomplished with the following five steps:

Step 1: calculate the Kalman gain, K , using Equation 3:

$$K_k = P_k^- H^T (HP_k^- H^T + R_k) \quad (3)$$

Step 2: calculate the error covariance matrix, P :

$$P_k = (I - K_k H) P_k^- \quad (4)$$

Step 3: output the estimated status, \hat{x} :

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k \hat{x}_k^-) \quad (5)$$

Step 4: project the status, \hat{x} :

$$\hat{x}_{k+1}^- = A \hat{x}_k \quad (6)$$

Step 5: project the error covariance matrix, P :

$$P_{k+1}^- = A P_k A^T + Q_k \quad (7)$$

The process can be simply described as the block diagram in Figure 4.

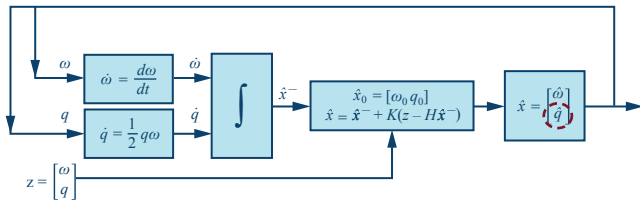


Figure 4. The Kalman filter flow chart for updating attitude.

Sensor Fusion Based on MSE

In the previous section, the observation variable is

$$z = [\omega \ q]^T$$

wherein there is no information from the compass. Since ω is the angular rate, we can only use the quaternion to import compass data q . We used the MSE method to get q , the component in the observation variable.

We define the variables as follows:

m_b and a_b : the compass magnetic value and acceleration value in the body frame.

m_n and a_n : the compass magnetic value and acceleration value in the navigation frame.

m_{n0} and a_{n0} : the initial static compass magnetic value and acceleration value in the navigation frame.

C_b^n

is the attitude transformation matrix from the body frame to the navigation frame, expressed using quaternion, which can be written as

$C_b^n(q)$

which gives us the error, ϵ , between the initial value in the navigation frame and the value mapped to the navigation frame from the body frame in real time.

Based on the previous definition, the MSE method can be used to get a best

$C_b^n(q)$

by minimizing Equation 8:

$$f(q) = \epsilon^T \epsilon = \left(\begin{bmatrix} a_{n0} \\ m_{n0} \end{bmatrix} - C_b^n(q) \begin{bmatrix} a_b \\ m_b \end{bmatrix} \right)^T \left(\begin{bmatrix} a_{n0} \\ m_{n0} \end{bmatrix} - C_b^n(q) \begin{bmatrix} a_b \\ m_b \end{bmatrix} \right) \quad (8)$$

(4) By taking the derivation of $f(q)$ and making it equal zero,

$$\frac{df(q)}{dq} = 0 \quad (9)$$

(5) we will get a best q in variance meaning. We used the Gauss-Newton method to solve the nonlinear equation above with first-order gradient convergence.

(6) By combining the angular rate, we will get the observation variable,

$$z = [\omega \ q]^T$$

(7) which fused compass data and IMU data in the Kalman filter.

The process can be simply described by the block diagram in Figure 5.

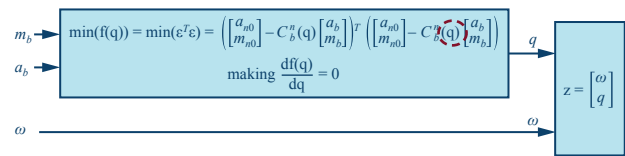


Figure 5. A sensor fusion block diagram using the MSE method.

Loose Coupling

As previously mentioned, we often encounter circumstances where compass sensors cannot be used. If the magnetic data is interfered with, the solved attitude accuracy will be worse than when using an IMU only. Therefore, we use loose coupling, which judges whether the magnetics sensor is available or not. When the magnetics sensor is not available, we will use only the IMU to find the attitude, and when the magnetics sensor is available, we will use the fusion algorithm to find the attitude, as shown in Figure 6.

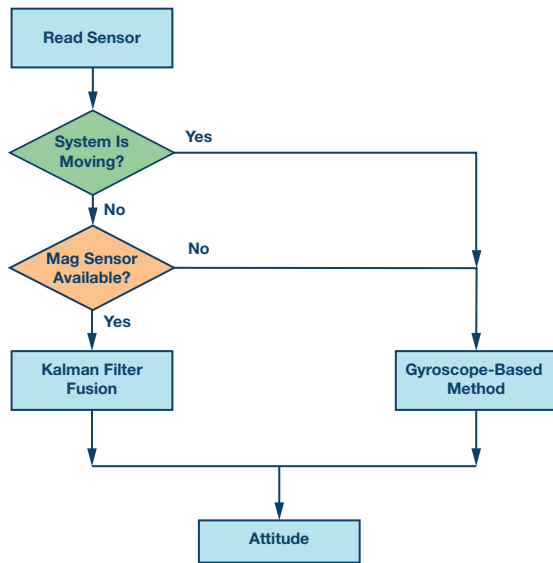
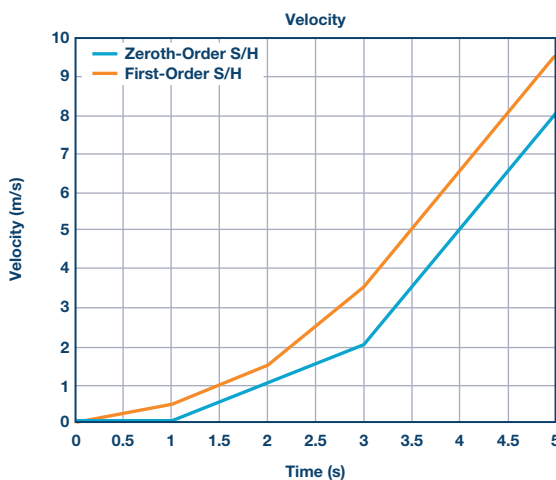


Figure 6. The attitude calculating flow chart.

After getting new data, or in a new attitude solving cycle (in some systems the sampling cycle is different from the attitude resolving cycle, but we are doing single sample cycle resolving here), we calculate the magnitude of acceleration. If it does not equal 1 *g*, we will not use the accelerator's output for the attitude calculation. Then we calculate the magnitude of the compass output and compare this with the initial value. If they do not equal each other, we will not use the geomagnetic sensor's data in this cycle. When both conditions are met, we will use the Kalman filter and perform MSE fusion.

Dead Reckoning (DR) Using ADIS16470

In navigation, dead reckoning is the process of calculating one's current position by using a previously determined position, and advancing that position based on known or estimated speeds or acceleration over a resolving cycle. The ADIS16470 accelerometer will be used here. Based on the attitude resolved in the last section, we get the strapdown system's moving orientation, then we need to calculate the distance in the orientation, and the position will finally be determined.



Dead Reckoning Method Introduction

Strapdown dead reckoning needs to track the position of an INS using a specific force equation, which is based on the acceleration measurement. Specific force equations can be simply described as Equation 10, Equation 11, and Equation 12:

$$a_e(t) = C_b^e a_b(t) \quad (10)$$

$$v_e(t + \delta t) = v_e(t) + \delta t \times (a_e(t + \delta t) - g_e) \quad (11)$$

$$s_e(t + \delta t) = s_e(t) + \delta t \times (v_e(t + \delta t)) \quad (12)$$

Where a_e is the acceleration in the Earth frame, a_b is the acceleration in the body frame, v_e is the velocity in the Earth frame, s_e is the distance in the Earth frame, g_e is the gravity acceleration in the Earth frame, which is [0 0 1] in *g* units. We need to emphasize that the Earth frame is different from the navigation frame—the Earth frame is NED oriented. This δt is the resolving cycle.

The first equation finds the acceleration projection from the IMU body frame to the Earth frame as the format

$$C_b^e$$

shows.

The second equation integrates or accumulates the acceleration into velocity; however, since the measured acceleration involves a gravity component, the gravity needs to be subtracted.

Similar to Equation 11, Equation 12 integrates velocity into the distance.

There are several problems in the traditional method.

- ▶ Accelerator outputs always have bias, which is combined with gravity, making it difficult to be subtracted in Equation 10, so the more accurate expression should be:

$$\begin{bmatrix} a_x^e \\ a_y^e \\ a_z^e \end{bmatrix} = C_b^e \left(\begin{bmatrix} a_x^b \\ a_y^b \\ a_z^b \end{bmatrix} - \begin{bmatrix} a_x^{bias} \\ a_y^{bias} \\ a_z^{bias} \end{bmatrix} \right) - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (13)$$

unless using some professional equipment, such as a dividing header.

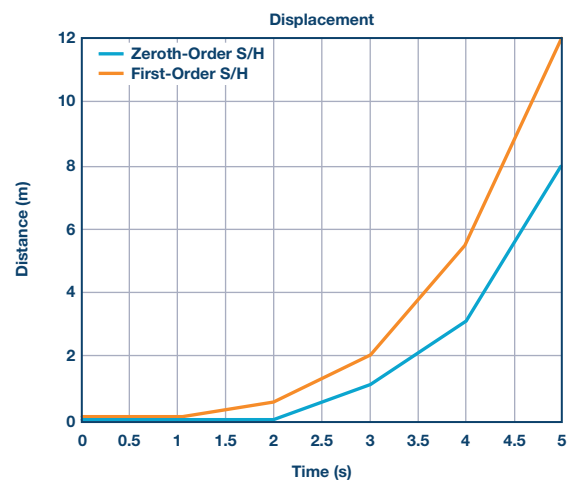


Figure 7. The comparison between zeroth-order and first-order integration methods in velocity calculation.

- ▶ The numerical integration method, the traditional method, commonly uses the zeroth-order holder method (the previous value) to do integration. However, for continuous movement, this will introduce significant error. For example, let's compare the following methods:

Method 1:

$$v_i = v_{i-1} + a_{i-1}t$$

(Zeroth-order holder)

Method 2:

$$v_i = v_{i-1} + a_{i-1}t + \frac{a_i - a_{i-1}}{2}$$

(linear interpolation)

With an acceleration of 0.5 m/s² across 5 seconds, the displacement will differ up to 4 m. The simulation result shows in Figure 7.

Based on the previous discussion, we modified two points of the traditional specific force equation for our application:

- ▶ We don't use the Earth coordinates as the navigation frame. Instead, as we did in finding the previous attitude, we use the initial attitude

$$\begin{bmatrix} a_{x0}^n & a_{y0}^n & a_{z0}^n \end{bmatrix}$$

as the navigation frame. In this way, both bias and gravity can be cancelled easily as in Equation 14:

$$\begin{bmatrix} a_x^n \\ a_y^n \\ a_z^n \end{bmatrix} = C_b^n \begin{bmatrix} a_x^b \\ a_y^b \\ a_z^b \end{bmatrix} - \begin{bmatrix} a_{x0}^n \\ a_{y0}^n \\ a_{z0}^n \end{bmatrix} \quad (14)$$

Though the bias and gravity components are contained in the initial attitude, this way we do not need to separate them to get each component, and we instead can directly subtract them.

- ▶ As compared between zeroth-order holder and first-order interpolation, we use the first-order to receive a more accurate integration result.

Kinematics Pattern and Zero Speed Updating Technology (ZUPT)

By using the IMU's initial value as the navigation frame, we can partially cancel the accelerator's initial bias impact. However, even if we can accurately measure the bias using the dividing header before we use a device, it is still difficult to cancel, unless we use another accurate sensor to periodically calibrate it. This is mainly caused by two parts: one is bias instability, which means the bias we measured before is not the actual bias now. The other is velocity random walk, which is integral to the acceleration. The previously mentioned undesired characteristics will make our calculated distance drift significantly. Even when we stop moving and remain static, the speed integrated from the acceleration still exists, and the distance will still increase.

To solve this problem, we need to find a way to reset the speed by using ZUPT technology. ZUPT technology closely relies on the application, so we need to get the kinematics characteristics of our system and application, then give some rules to our algorithm. The more kinematics patterns we find, the more accurate our result will be.

We apply our experiment by moving a swivel chair with the SINS system in place. Since our research is not limited to a specific application, we use the following kinematics assumptions:

- ▶ For dead reckoning, there is no z-axis moving in the navigation frame. This limit is only for dead reckoning, it is not used in attitude resolving. Obviously, we are moving the system in 2D space. This helps cancel the z-axis error.
- ▶ All turning happens after stopping. If turning happens when moving, attitude resolving will be interfered with since extra acceleration will be involved.
- ▶ The acceleration cannot remain unchanged for more than 500 ms if the system is moving. Velocity cannot stay unchanged for longer than 2 s. Since we are pushing or pulling a swivel chair, it is difficult to manually keep the force accurately unchanged for more than 500 ms, and it is difficult for a human to keep pushing a swivel chair with uniform speed for more than 2 s. In fact, we are using this rule to do ZUPT.
- ▶ The acceleration cannot be larger than ± 1 m/s². This rule is used for some noise filtering, which is based on our pulling or pushing force on the chair, which will not be large.

As we can see in Figure 8, when the system is moving in the X direction (after projected to the navigation frame), the Y direction will also generate acceleration, after integration, the Y direction speed will not be zero, which means even if we are only moving in the X direction, the dead reckoning system will still give us the Y component.

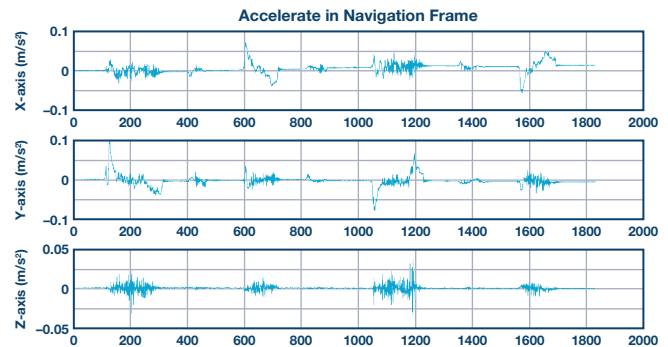


Figure 8. The acceleration in three directions in the navigation frame.

Based on the third kinematic assumption, we can use ZUPT to cancel this error. The integrated velocity after ZUPT is shown in Figure 9.

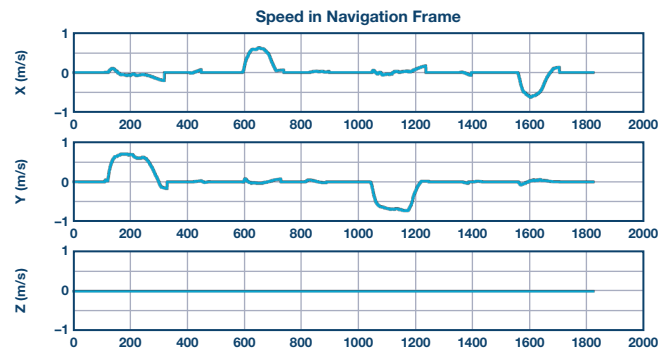


Figure 9. The velocity in three directions in the navigation frame.

Though we used the third assumption, as previously shown, the error still cannot be totally cancelled. The error cancellation depends on the threshold to the set zero acceleration and zero speed. However, most errors have been corrected.

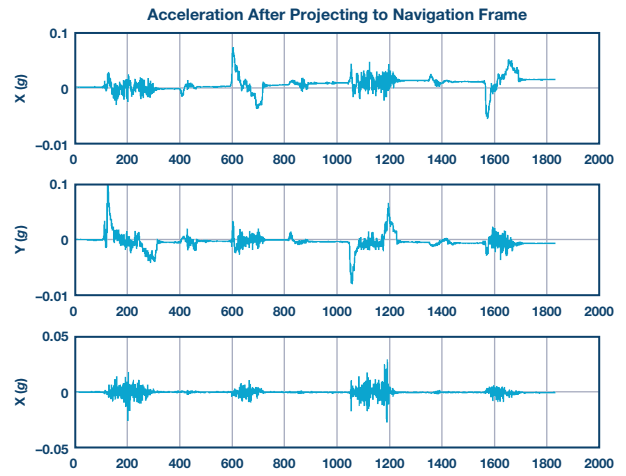
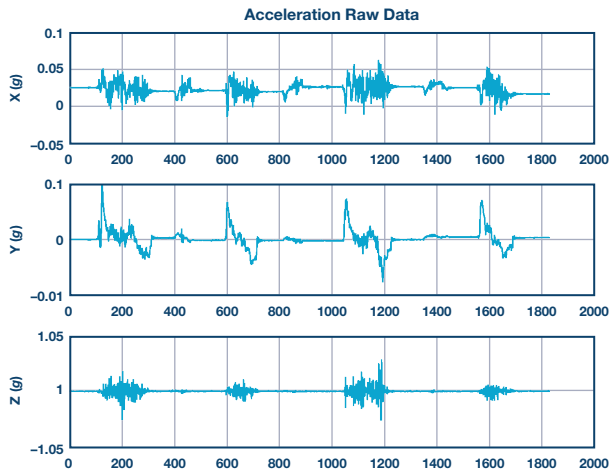


Figure 10. The acceleration in the body frame (left) and in the navigation frame (right).

Baseline Shifting Cancellation

Although ZUPT is used, the zero acceleration may still be unreachable at times. This results in two factors:

- ▶ We can't totally cancel the bias instability error and the velocity random walk using ZUPT.
- ▶ The attitude we resolved has some error, which will lead to the projected (from body frame to navigation frame) acceleration error.

Take Figure 10 for example. The left picture in Figure 10 is the raw data (body frame) from ADIS16470 and the right picture in Figure 10 is the acceleration projected in the navigation frame. It can be seen that the projected acceleration is not zero when it stops moving. As it always changes, we call this baseline shifting.

In order to cancel the baseline shifting, we need to continuously get the shifted bias in real time and subtract it from the projected acceleration. The result is shown in Figure 11.

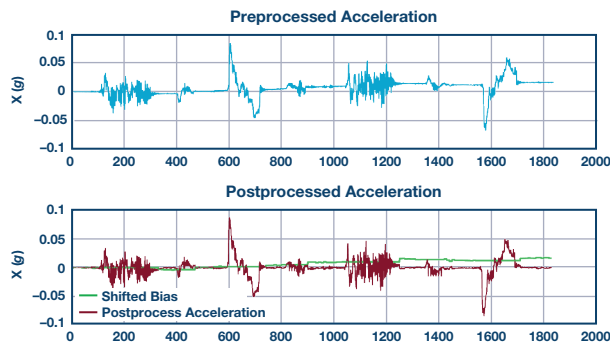


Figure 11. The acceleration before (top) and after (bottom) baseline shift cancellation.

The upper plot is the acceleration before baseline shift cancellation, the green trace in the bottom plot is the baseline shifting we calculated, and the red trace is the acceleration after baseline shift cancellation.

The dead reckoning process can be described briefly using the block diagram in Figure 12. We input the body frame acceleration a_b and the attitude transforming matrix (from AHRS)

$$C_b^n$$

to the DR system. Once this is done, we will get the position in the navigation frame.

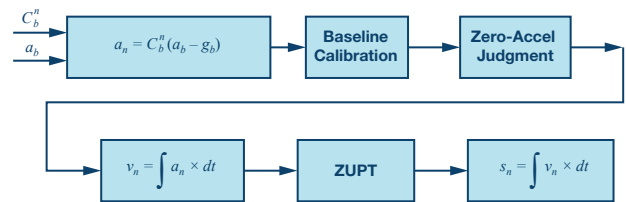


Figure 12. The flow chart for dead reckoning.

Experiment Result and Conclusion

Experiment Result

We build our system, as in Figure 13, by connecting the ADIS16470 evaluation board and the RM3100 compass evaluation board to ADI's ADuCM4050 board using an SPI port. The ADuCM4050 adjusts the data format and does time synchronization (since the data rate of the IMU and the compass are different). Then the captured data is transmitted to the computer using the UART. All the calculations, including calibration, AHRS, and DR, are performed in MATLAB®.

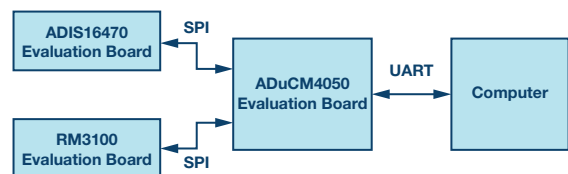


Figure 13. The experiment platform setup.

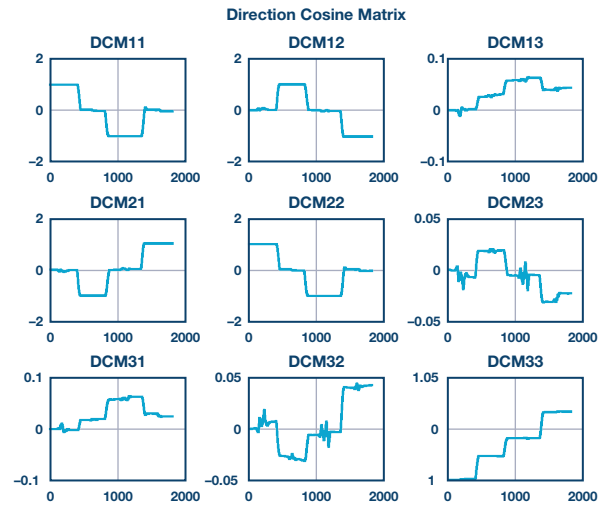
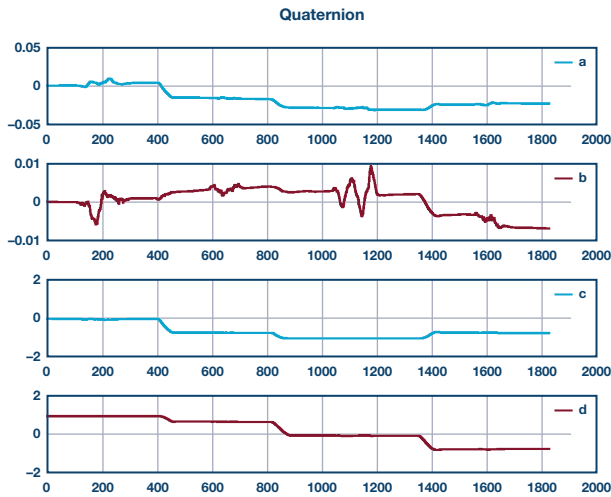


Figure 14. The attitude in quaternion format (left) and DCM format (right).

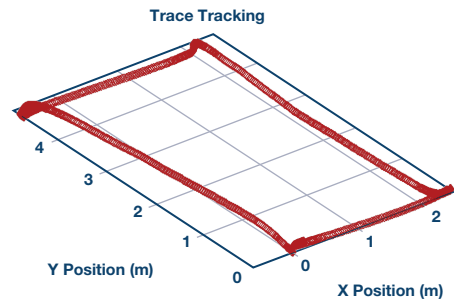
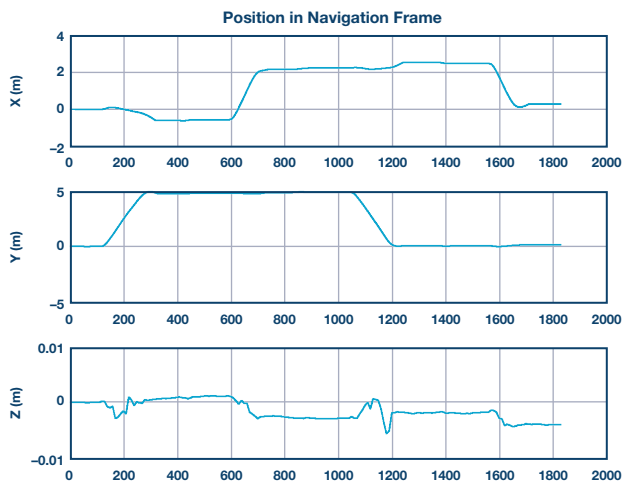


Figure 15. The position calculation result.

The experiment was implemented by placing the boards and computer on a swivel chair and pushing the swivel chair around a circle in our laboratory.

- ▶ AHRS output: The attitude is shown in quaternion format and DCM format as shown in Figure 14.
- ▶ DR output: The dead reckoning result with X-Y-Z position and 3D plot are shown in Figure 15.

Conclusion

This article shows a basic process used to build a navigation system using ADI's IMU ADIS16470 and geomagnetic sensor RM3100 by introducing the calibration, AHRS, and DR methods we used. With limited conditions, such as the platform and experiment environment, it is difficult for us to further test the platform and algorithm.

There are a lot of methods that can be used to improve the result, for example:

- ▶ Using an odometer or UWB distance measurement to fuse the accelerometer with the IMU to get better distance in DR.

- ▶ Using a more complicated kinematics model that involves more characteristics at the sensor level and system level in AHRS and DR, such as the vibration, acceleration, and deceleration models of the system, the ground surface flatness, etc. This means giving more boundary conditions to our calculation to achieve more accurate results.
- ▶ Using more accurate numerical calculation methods, such as using Simpson's rule or cubic splines interpolation to do integration in DR, or by using the Newton method instead of the Gauss-Newton method to solve nonlinear MSE equations, etc.

The last, but most important, point we find in experiments is that INS is extremely closely related to the application, or the kinematics pattern. For example, we performed the experiment in two places: a laboratory without carpeting on the floor and an office with carpeting. The DR result shows a huge difference if we use the same set of parameters. Accordingly, no matter which application, such as patient tracking, AGV navigation, or parking localization, or for different conditions in the same application, we all need to comprehensively understand its kinematics model.

References

1. Long Li and Zhang He. “Automatic and Adaptive Calibration Method of Tri-axial Magnetometer.” Chinese Journal of Scientific Instrument, 2013.
 2. Timothy Sauer. *Numerical Analysis* (2nd Edition). Pearson, 2011.
 3. David H. Titterton. *Strapdown Inertial Navigation Technology* (2nd Edition). The Institution of Electrical Engineers, 2004.
- Gongmin, Yan. “Research on Vehicle Autonomous Positioning and Orientation System.” Ph.D. thesis. Northwestern Polytechnical University, China, 2006.
- Marins, João Luís. “An Extended Kalman Filter for Quaternion-Based Orientation Estimation Using MARG Sensors.” IEEE, 2001.
- Prikhodko, Igor P. and Brock Bearss. “Towards Self-Navigating Cars Using MEMS IMU: Challenges and Opportunities.” IEEE, 2018.
- RM3100. PNI Sensor Corporation, 2018.
- Woodman, Oliver J. “An Introduction to Inertial Navigation.” University of Cambridge, August 2007.

Joel Li [Zhao Li] [joel.li@analog.com] joined ADI in 2017 as an applications engineer and currently works on some applications of IMU and lidar in China. Joel received his master's degree in photoelectric detection from the Chinese Academy of Sciences in 2013.



Joel Li

Van Yang [van.yang@analog.com] is a field applications engineer at Analog Devices in Shanghai, China. He joined ADI in 2015 to support regional medical and industrial customers in China. Prior to joining ADI, Van worked at Texas Instruments as an FAE for four years. Van earned his master's degree in communication and information systems from Huazhong University of Science and Technology in Wuhan, in 2011. In his spare time, he is a super fan of basketball and enjoys hiking.



Van Yang
