Received: 25 July 2025 | Revised: 5 November 2025 | Accepted: 19 December 2025

DOI: 10.33012/navi.751

ORIGINAL ARTICLE

Gyrocompassing: From Gravity Rotation Analysis to Inertial Trajectory Fitting

Pierre Bénet | **Alexis Guinamard**

SBG Systems, Carrières sur Seine, France

Correspondence
Pierre Bénet
SBG Systems, 78420 Carrières sur Seine, France.
Email: pierre.benet@sbg-systems.com

Abstract

Gyrocompassing is the process of using the rotation of the Earth to determine heading. This approach is the state-of-the-art solution when precise and robust heading is needed, such as in ship navigation. Currently, this procedure is performed using a high-grade inertial measurement unit, consisting of an accelerometer and a gyrometer. To circumvent parasitic motions from the ship, the Earth's rotation is not directly measured by the gyrometer. The commonly adopted solution is to measure the rotation of the gravity vector in the inertial frame. A curated, straightforward method for gravity vector fitting based on singular value decomposition was developed here to provide a baseline algorithm. This method was compared with a second algorithm based on inertial trajectory fitting. Our results show how a trajectory-fitting algorithm can improve robustness and accelerate gyrocompassing compared with a traditional gravity analysis algorithm.

Keywords

alignment, gyrocompassing, IMU, initialization, INS, SVD

1 | INTRODUCTION

Inertial navigation systems (INSs) require proper initialization to provide reliable positioning and attitude information. While low-cost systems can tolerate limited accuracy and operational constraints during initialization, some applications, such as marine navigation, require higher availability, robustness, and accuracy. Gyrocompassing INSs represent the de facto solution for such applications. These INSs provide the highest attitude precision and require no additional sensors.

Once initialized, the INS is able to maintain its attitude to within a fraction of a degree or even more precisely, owing to additional sensors or assumptions about the motion of the carrier. This continuous procedure of fine alignment is typically conducted by an extended Kalman filter (EKF). However, the EKF requires a starting point that is sufficiently close to reality, in particular, a heading with a standard deviation (SD) of a few degrees. Otherwise, the EKF may fail to converge properly or may even diverge. Moreover, the success of this filtering is closely related to the estimated confidence in the initialization relative to the true error. Therefore, a high-integrity initialization procedure is needed.

Licensed under CC-BY 4.0 © 2026 Institute of Navigation

In static situations, a gyrometer directly measures the Earth's rotation. The attitude is directly deduced from this rotation and the measured acceleration (Titterton & Weston, 2004). However, in marine situations, the INS is never truly static, and the gyrometer measures the rotation of the ship, which is several times larger than the Earth's rotation.

Thus, directly observing the Earth's rotation has been abandoned in favor of analyzing the apparent gravity in a fixed inertial frame (Gade, 2016). Indeed, as illustrated in Figure 1, in an inertial frame of reference, also called inertial space, the gravity at a specific point on Earth corresponds to a cone whose main axis is the Earth's rotation axis (Gaiffe et al., 2000). The INS should observe the same cone once its accelerometer values are expressed in the inertial frame. This method has been used for decades (Gaiffe et al., 2000) and can be viewed as the basic principle for ship gyrocompassing initialization.

However, the implementation details for solving this problem can vary. In the solution initially referenced by Gaiffe et al. (2000) and later detailed by Lacambre et al. (2020), the acceleration is differentiated in the inertial frame. Gu et al. (2008) averaged the acceleration in the inertial frame during two different time periods to obtain a set of two vectors that are matched with the theoretical vectors, forming a minimal vector alignment problem, also referred to as Wahba's problem (Wahba, 1965). The averaging of accelerometer data has the desired property of naturally filtering high-frequency noise. Sun et al. (2013) and Li et al. (2013) studied filtering strategies to improve these results. Silson (2011) used the same principle with more vectors in Wahba's problem and used the global navigation satellite system (GNSS) velocity to improve gyrocompassing convergence in dynamic scenarios. Che et al. (2015) and Liu et al. (2014) used a horizontal frame to remove sway interferences from the main signal. Xu et al. (2017) and Zhu and Qiao (2019) used adaptive filtering to improve coarse alignment results. Liu et al. (2015) developed a latitude-free alignment using three gravity vector estimates in the inertial frame. Zhao et al. (2016) developed another latitude-free alignment using multiple gravity vectors. Pei et al., (2020) developed an EKF capable of tackling the nonlinearity of initialization using Lie groups and extended this approach to Doppler velocity log (DVL) coupling (Pei et al., 2020).

Lacambre et al. (2020) presented the general necessary conditions for gravity analysis to be valid. The most important factor beyond gyrometer quality is the "true movement hypothesis." That is, how static is the ship? Or how well is the displacement of the boat known? This question also reflects the current uses of such INSs. In particular, the INS is often aided by a GNSS receiver. Additionally, users

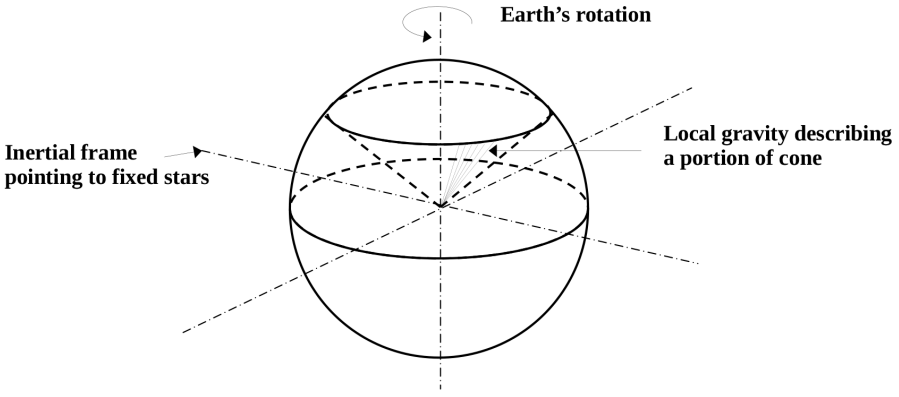

FIGURE 1 Conical behavior of local gravity from an inertial space viewpoint

want to have the best accuracy for a given sensor setup. Hence, Gaiffe et al. (2000) proposed a basic expansion that relates positioning accuracy, averaging time, and gyrocompassing accuracy.

This background directly introduces our patent-pending algorithm based on position observation (Bénet et al., 2024, 2025). The underlying principle of our position-based gyrocompassing is that when performing the mechanization of an INS in a static situation, an incorrect heading would lead to an incorrect correction of the Earth's rotation rate on the gyrometer, which would cause errors in pitch and roll, ultimately resulting in an error in the integration direction of the accelerometer. In the end, the output position from the mechanization would drift proportionally to the cube of time. Conversely, fitting an inertial trajectory with a true trajectory would yield an estimate of the attitude, even if the target trajectory is static. In this case, GNSS information would not be essential, as long as static movement is detected.

In this paper, using an inertial trajectory-fitting algorithm (Bénet et al., 2024, 2025), we observe a performance similar to that of the baseline gyrocompassing algorithm based on gravity rotation analysis. Because the position is obtained from acceleration integration, acceleration data are naturally filtered, with no need for a low-pass filter. This algorithm was initially developed for single-antenna low-dynamic initialization using micro-electromechanical system (MEMS) sensors (Bénet et al., 2024, 2025). Consequently, this algorithm is capable of providing both dynamic-based alignment and gyrocompassing alignment.

However, this initialization comes with an expense, primarily a higher computational burden that arises from the resolution of a nonlinear least-squares problem and the computation of the necessary Jacobian (Bénet et al., 2024, 2025). A computation cost also stems from the estimated variables, which include not only the initial attitude (roll, pitch, yaw) but also a starting velocity, as well as initial gyrometer and accelerometer biases. A dedicated regularization method has been developed to reduce the number of iterations required to no more than 15 iterations. Consequently, the complexity of this initialization procedure remains sufficiently low for real-time initialization in an embedded INS processor.

In this paper, we present a detailed implementation of the resolution of gyrocompassing using gravity rotation analysis in the inertial frame. As further discussed in Section 2, we formulate this problem as a two-set vector alignment problem, commonly referred to as Wahba's problem (Wahba, 1965). We use singular value decomposition (SVD) to solve this problem, as this approach offers important metrics on the accuracy of the results obtained (Markley, 1988) and robustness (Markley & Mortari, 2000). In Section 3, we summarize the basic principles of our trajectory-fitting algorithm (Bénet et al., 2024, 2025). In Section 4, we compare the two different gyrocompassing algorithms under various conditions in marine and automotive scenarios. In particular, we pay special attention to the method's integrity, comparing the obtained attitude with that of a full EKF real-time kinematic (RTK)-GNSS post-processed solution.

1.1 | Use Cases

Before discussing the algorithms, we introduce the different initialization scenarios covered in this paper:

- **Static gyrocompassing** refers to determining true north when the platform or vehicle is stationary by directly or indirectly measuring the Earth's rotation. In this situation, the platform can still be moderately disturbed. For example,

a moored ship is still disturbed by small waves, but these disturbances should not jeopardize the initialization.

- **Dynamic gyrocompassing** refers to determining true north when the platform or vehicle is moving at cruise speed by directly or indirectly measuring the Earth's rotation. This determination can be achieved either through prolonged data averaging or by using external aiding to compensate for motion (for example, a DVL will provide a relative source of motion).
- **GNSS alignment** can occur when one compensates for movement by using an absolute motion source. If the movement is significant, the GNSS motion primarily constrains the INS orientation rather than the Earth's rotation. In this case, the solution cannot be classified as gyrocompassing.

2 | GRAVITY ROTATION ANALYSIS

2.1 | Gravitational Motion

The two cones that must be aligned include the vector set that describes the local gravity reaction in a known inertial frame $g_i(t)$:

$$g_i(t) = -exp\left(t\left[\Omega_E\right]_\times\right).g_{i0} \tag{1}$$

and the vector set that describes the acceleration felt by the IMU brought back from the body frame to a locally consistent inertial frame, owing to gyrometer integration:

$$a_l(t) = R_b^l(t).a_b(t) \tag{2}$$

with the following terms:

i	A subscript indicating the inertial frame that is equal to the north–east–down (NED) frame at $t = 0$. After $t = 0$, this frame differs from the NED frame, as the NED frame rotates with the Earth whereas the inertial frame does not.
l	A subscript indicating the local inertial frame that is equal to the body frame at $t = 0$. This frame is computed by using gyrometer integration.
g_{io}	$(0, 0, \|\|g\|\|)^T$ is the gravitational acceleration in an inertial frame at $t = 0$.
Ω_E	$(\Omega_{EN}, 0, \Omega_{ED})^T$ is the rotation of the Earth in the NED frame. The latitude is needed for its computation.
$R_b^l(t)$	The rotation from the body frame to the local inertial frame. This term is the integration of the gyrometer data $\dot{R}_b^l(t) = R_b^l(t)[\omega_b(t)]_\times$ starting from the identity $R_b^l(0) = I_{3\times3}$.
$a_b(t)$	The acceleration of the body frame as measured by the accelerometer.
$\omega_b(t)$	The rotation rate of the body frame as measured by the gyrometer.

To align the two cones formed by Equations (1) and (2), the main unknown is introduced: R_l^i, which represents the initial attitude. The acceleration felt by the INS $a_i(t) = R_l^i a_l(t)$ should be largely equal to the theoretical gravity reaction in the inertial frame $g_i(t)$:

$$g_i(t) = R_l^i a_l(t) + \gamma(t) \quad \text{with} \quad \gamma(t) \sim \mathcal{N}(0, \sigma^2) \tag{3}$$

2.2 | Wahba's Problem

Based on the above discussion, we eventually obtain Wahba's problem, which consists of finding the alignment between $a_l(t)$ and $g_i(t)$ by minimizing the following cost function:

$$L(R) = \frac{1}{2}\sum_t w(t)\left\| g_i(t) - Ra_l(t)\right\|^2 \tag{4}$$

Expanding this expression reveals a constant component and a component that is dependent on R. Using the property of trace invariance under cyclic permutation, we place the rotation matrix on the left side of the trace:

$$L(R) = \lambda_0 - \sum_t w(t) g_i(t)^T R a_l(t) = \lambda_0 - tr\left(RB^T\right) \tag{5}$$

where:

$$\lambda_0 = \frac{1}{2}\sum_t w(t)\left\| g_i(t)\right\|^2 + \frac{1}{2}\sum_t w(t)\left\| a_l(t)\right\|^2 \quad \text{and} \quad B = \sum_t w(t) g_i(t) a_l(t)^T \tag{6}$$

Thus, the minimization problem of Equation (4) becomes the problem of maximizing the trace of RB^T.

2.3 | SVD Method

To maximize the trace of RB^T, first, the SVD of B is computed with U, V orthogonal and S being a diagonal matrix, where the positive singular values of B are sorted in descending order:

$$B = USV^T \tag{7}$$

Then, the solution of our problem, R, is computed as the product of U and V^T, where the axis of the minimum eigenvalue is opposed whenever the product of $\det(U)$ and $\det(V)$ is negative:

$$R = U\,\mathrm{diag}\left[1 \;\; 1 \;\; \det(U)\det(V)\right]V^T \tag{8}$$

2.4 | Covariance

To compute and simplify the expression of the covariance of the result, Markley (1988) used the hypothesis that once the solution is computed, the two sets of vectors $g_i(t)$ and $a_i(t)$ are interchangeable. Using one set of vectors or the other would not impact the estimated covariance. The resulting expression treats both terms equally, using the matrix RB^T:

$$P = \left(tr\left(RB^T\right) I_{3\times 3} - RB^T\right)^{-1} \tag{9}$$

Ultimately, we can dynamically compute the mean squared error of our optimization problem to correct our estimated covariance and adapt to noise with unknown intensity. From the two sets of vectors $g_i(t)$ and $a_i(t)$ and the alignment result R, the optimal weights of the observation are given by the inverse of their mean squared error:

$$w^{-1} = \sigma^2 = \frac{1}{3n}\sum_t \left\| g_i(t) - Ra_l(t) \right\|^2 \tag{10}$$

Here, $3n$ is the number of samples in our problem multiplied by 3 to take into account the dimension of each sample. Because the rotation obtained from the SVD is invariant to any scale applied to B, we can compute R via SVD using $\tilde{B} = \Sigma_t g_i(t) a_l(t)^T$, where the weights have been ignored in Equation (6); we can then recover the right covariance by substituting $B = w\tilde{B}$ in Equation (9):

$$P = w^{-1}\left(tr\left(R\tilde{B}^T\right) I_{3\times 3} - R\tilde{B}^T \right)^{-1} \tag{11}$$

2.5 | Filtering

As explained earlier, filtering the input data is important for gravity analysis. The most widely used solution is to average these accelerations over periods of time. For example, Silson (2011) used 1-s averaging periods. We studied the averaging period to determine which value gives the best performance and found a similar averaging period. We also observed that higher robustness and versatility can be obtained by using an averaging period proportional to the window size. Thus, we split the window into $n = 30$ samples, which corresponds to 1-s averaging samples for 30-s windows or 3-s averaging samples for 90-s windows. In this case, Equation (4) has the following form:

$$L(R) = \frac{1}{2}\sum_{k=0}^{n-1} w(k) \left\| \int_{t_k}^{t_{k+1}} g_i(t)dt - R\int_{t_k}^{t_{k+1}} a_l(t)dt \right\|^2 \tag{12}$$

Filtering the signal does not significantly affect the resulting orientation. However, this step greatly impacts the covariance estimation (Equation (11)), which is crucial for distinguishing successful from erroneous initializations. Averaging this measure has two beneficial effects:

- High-frequency sinusoidal noise (e.g., from engines or waves) is eliminated from the mean squared error (Equation (10)), which prevents the covariance from becoming overly pessimistic owing to non-problematic noise.
- Low-frequency noise (e.g., from a moving carrier) becomes more prominent in the mean squared error (Equation (10)). This noise is no longer masked by high-frequency white noise; thus, the covariance increases appropriately for unreliable results.

In our case, averaging the acceleration is equivalent to computing a velocity. No distortion is introduced. Thus, in the literature, this problem is more commonly known as a "velocity loci" alignment. Moreover, formulating the problem as a delta velocity vector alignment enables further extension of this algorithm to use GNSS velocity for in-motion alignment (Silson, 2011). One can be tempted to compute the full integral of accelerations to obtain a velocity signal for alignment or its quadratic integral to obtain a position signal for alignment. Although the mean estimate remains largely the same, the estimated covariance in Equation (11) becomes unreliable for distinguishing successful from erroneous results.

We refer to this method as the baseline method, as it is very similar to the method presented by Silson (2011), primarily differing in the Wahba's resolution method,

which was the method of Davenport (1968). Although that method is known to yield the same result, the SVD method used here is simpler and has a computational speed similar to that of Davenport's method (Liu et al., 2022; Markley & Mortari, 2000). Another significant difference is the adaptive weighting of the covariance of Equation (11), which eliminates the need to know the weight ($w(t)$) associated with each measurement.

3 | INERTIAL TRAJECTORY FITTING

3.1 | Principle

The given solution is to fit a small set of position points (e.g., GNSS trajectory) with a pure inertial trajectory computed using a set of high-frequency inertial sensor data (three-axis accelerometers and gyrometers), without making any prior assumptions about the initial state. This is achieved by iteratively minimizing a function (Bénet et al., 2024, 2025). Figure 2 shows the typical convergence of the algorithm:

- **Iteration 1:** At first, the INS is not initialized: the roll, pitch, and yaw are incorrect, and the initial difference between the inertial and GNSS trajectories is significant. These differences arise because the inertial trajectory is the result of integrating the measured acceleration minus the gravitational acceleration. In practice, the gravitational acceleration is integrated along the wrong axis.
- **Iteration 5:** After a few iterations, the algorithm can achieve convergence for the roll and pitch: the roll and pitch will converge such that the measured acceleration will be in opposition to the actual gravity. However, the yaw can remain incorrect.
- **Iteration 10:** As the yaw converges, we also obtain accurate values for IMU biases and velocity.

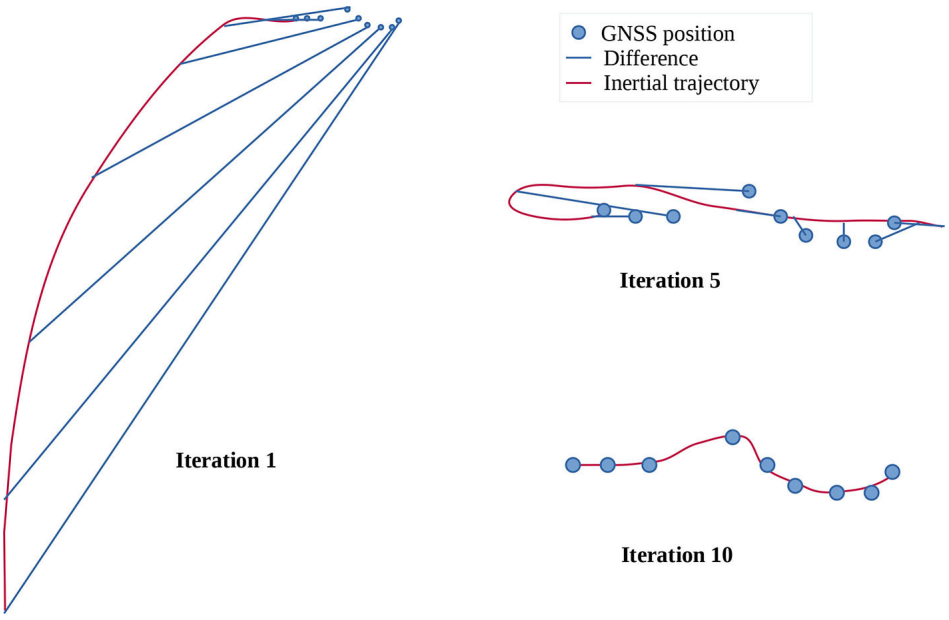

FIGURE 2 Illustration of the process of initializing an INS on a pedestrian trajectory using inertial trajectory fitting with a 1-Hz GNSS receiver over 10 s

3.2 | Gyrocompassing Behavior

This algorithm was initially designed for GNSS-aided dynamic MEMS initialization. However, it can also be used for static gyrocompassing, with or without a GNSS. The only difference is that the trajectory to fit is static. The best fit is achieved when the heading is correct. Otherwise, the Earth's rotation would be integrated along the wrong axis, gradually leading to gravity integration along the wrong axis, resulting in a non-static inertial trajectory.

- To gyrocompass without a GNSS, we use a static position hypothesis (Figure 3). The accuracy of this static hypothesis is given by the residual of the inertial trajectory fitting. The residual can decrease to very low values for short periods of time, yielding a brief gyrocompassing capability for truly static situations. If the true vehicle movement is not sufficiently static, this method will fail.
- When using a GNSS (Figure 4), GNSS noise will prevent very brief gyrocompassing capabilities, but the algorithm will not be affected by vehicle movement. In particular, in dynamic situations, the natural operation of the algorithm will yield shorter initialization times owing to GNSS course information, compared with the time that would be required if observing only the Earth's rotation.

3.3 | Trajectory Difference Function

To fit the inertial trajectory with the GNSS trajectory, a set of parameters is introduced:

$$x = \left(b_a, b_g, v_0, \theta_0\right)^T \tag{13}$$

The accelerometer bias b_a, gyrometer bias b_g, initial velocity v_0, and initial attitude θ_0 are used as input to reconstruct a local inertial trajectory, corresponding to a 12-dimensional vector.

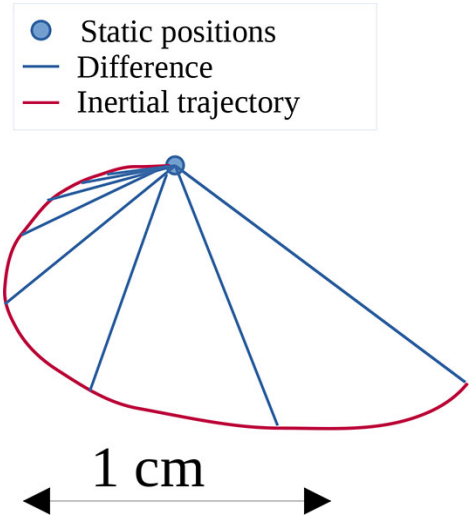

FIGURE 3 Static gyrocompassing using static hypothesis

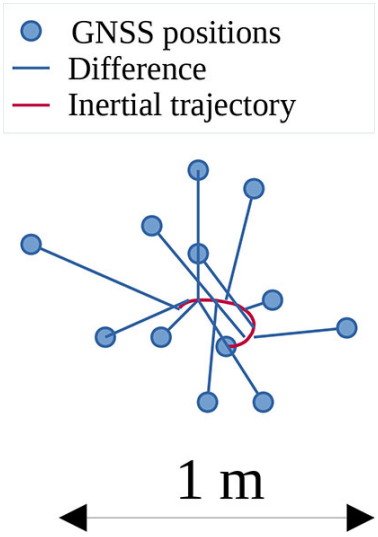

FIGURE 4 Static gyrocompassing using single-point GNSS

The difference function is defined as the stack of the difference of several GNSS points and the associated inertial position:

$$f\begin{pmatrix} b_a \\ b_g \\ v_0 \\ \theta_0 \end{pmatrix} = f(x) = \begin{pmatrix} p_{t_1}(x) - GNSS_1) \\ p_{t_2}(x) - GNSS_2) \\ \cdots \\ p_{t_n}(x) - GNSS_n) \end{pmatrix} \tag{14}$$

The inertial trajectory $p_{t_i}(x)$ is computed via an inertial mechanization that involves integrating a set of differential equations (Titterton & Weston, 2004):

$$\begin{aligned} \dot{R}_t &= R_t\left[\omega_t - b_g - R_t^T\Omega_E\right]_\times \quad \text{with} \quad R_{t_0} = e^{[\theta_0]_\times}.R_0 \\ \dot{v}_t &= R_t.(a_t - b_a) + g - 2\Omega_E \wedge v_t \quad \text{with} \quad v_{t_0} = v_0 \\ \dot{p}_t &= v_t \quad \text{with} \quad p_{t_0} = GNSS_0 \end{aligned} \tag{15}$$

- The initial angle parameter is utilized at the start of the integration using an exponential map $e^{[\theta_0]_\times}$.
- The initial velocity v_0 is also used only at the start of the integration.
- The position starts at the first GNSS point provided by the receiver. Thus, this parameter is not estimated.
- The remainder of the mechanization is accomplished by utilizing raw accelerometer data a_t and gyrometer data ω_t corrected by the accelerometer bias parameter b_a and gyrometer bias parameter b_g. Additionally, the process accounts for gravitational acceleration g and the Earth's rotation rate Ω_E.

3.4 | Resolution

To make this process possible, not only is the GNSS-inertial fitting error minimized, $||f(x)||^2_{G^{-1}}$, but the sensor bias values are also minimized, $||x - x_0||^2_{P_{in}^{-i}}$, according to the intrinsic specifications of the sensor (Bénet et al., 2024, 2025):

$$\tilde{x} = \arg\min_{x} \left(\left\| f(x) \right\|^2_{G^{-1}} + \left\| x - x_0 \right\|^2_{P_{in}^{-1}} \right) \tag{16}$$

To minimize this value, we apply a Gauss–Newton algorithm (Fletcher, 2000), an iterative algorithm that will converge to the solution. After linearization, we obtain the following Gauss–Newton iteration:

$$\Delta x = -\left(\left(\frac{\partial f(x)}{\partial x} \right)^T G^{-1} \frac{\partial f(x)}{\partial x} + P_{in}^{-1} \right)^{-1} \left(\left(\frac{\partial f(x)}{\partial x} \right)^T G^{-1} f(x) + P_{in}^{-1} (x - x_0) \right) \tag{17}$$

Here, $G = \dfrac{\left\| f(x) \right\|^2}{n}$ is the averaged squared residual, serving as both the covariance of the observation f and its weight. This adaptive weight, which increases during convergence as the solution approaches its optimum, gives the algorithm its excellent convergence properties and robustness. In practice, the algorithm always converges within 15 iterations. The quality of the estimated variables is given by the covariance (Bénet et al., 2024, 2025):

$$P_{out} = \left(\left(\frac{\partial f(x)}{\partial x} \right)^T .G^{-1} . \frac{\partial f(x)}{\partial x} + P_{in}^{-1} \right)^{-1} \tag{18}$$

For the very rare case in which the algorithm does not converge, the high residual of the function f through G^{-1} would be immediately felt on the output covariance, ensuring that integrity is maintained.

As the computation of $f(x)$ involves stacking results of inertial mechanization over different durations, it is difficult to differentiate f with respect to its input $x = (b_a, b_g, v_0, \theta_0)^T$, which consists of the accelerometer bias, gyrometer bias, initial velocity, and initial attitude. For the sake of simplicity, the Jacobian of f is computed numerically using central differences:

$$\frac{\partial f(x)}{\partial x} = \left(\frac{f(x + \varepsilon \cdot e_1) - f(x - \varepsilon \cdot e_1)}{2\varepsilon} \quad \cdots \quad \frac{f(x + \varepsilon \cdot e_{12}) - f(x - \varepsilon \cdot e_{12})}{2\varepsilon} \right) \tag{19}$$

Here, ε is a sufficiently small number and $e_i = (\delta_{1,i}, \delta_{2,i}, \ldots \delta_{12,i})^T$ represents the vectors of the canonical basis, where $\delta_{i,j}$ is the Kronecker delta.

This generic implementation will facilitate subsequent customization of the algorithm. For example, in Section 4.6, $f(x)$ is modified to support body velocity information instead of GNSS information.

4 | RESULTS

4.1 | Test Cases

In our tests, we compared the trajectory-fitting algorithm and gravity analysis. Both algorithms use the same input (IMU data) and can be applied to different windows of data. Both algorithms output the roll, pitch, and yaw, along with the predicted uncertainty. The trajectory-fitting algorithm is also capable of estimating additional data, such as IMU bias and velocity. It can be configured to work with

GNSS position or DVL data, but unless otherwise specified, the trajectory-fitting algorithm operates without such aiding. Table 1 summarizes the different functionalities of both algorithms.

Additionally, Table 1 presents processing time tests for both algorithms developed in C++ using a 60-s data window on a laptop. The reported processing time was averaged over thousands of trials. For this window size, the processing time is dominated by gyrometer integration, which scales proportionally with the window size. In both algorithms, the gyrometer data are first pre-integrated from 200 Hz to 50 Hz. For the gravity rotation analysis, the gyrometer is integrated once over the entire window period. However, for the trajectory-fitting algorithm, the integration is repeated up to 360 times, owing to the inclusion of up to 15 Gauss–Newton iterations, each requiring 24 calls to the mechanization procedure (for central difference calculations of the Jacobian). Indeed, we observed that the trajectory-fitting algorithm requires roughly 300-fold more computation power than the gravity analysis.

As the results are obtained at the start of the integration period for both algorithms, it is best to choose the end of the window as a starting point and to perform the integration in the opposite direction, going backward in time. This corresponds to real-time functioning of the algorithm, allowing for the most up-to-date result for starting the EKF. This approach also prevents the need to reprocess the data over the entire window with the EKF if the other end of the window is chosen for the starting point. Yet, in the case of EKF post-processing, if a backward EKF pass is desired, we perform the opposite, computing the initialization procedures with forward integration so that the result is obtained at the start of the window.

Both automotive (Figure 5) and marine (Figure 6) logs were obtained using the Horizon IMU, which has an initial gyrometer bias of 0.3°/h. This bias results in a gyrocompassing capability of approximately 2° at 50° latitude. Therefore, we expect to achieve this accuracy in our subsequent tests. We consider a solution as valid when the reported uncertainty is less than 5°. As the alignment result is intended to be used to initialize an EKF, this threshold ensures that the filter will operate within its near-linear regime. However, it is strategic to start the EKF as quickly as possible, as the Kalman filter is computationally lightweight and considers the details of all noise models from the sensors. Therefore, we chose the smallest temporal window that primarily yielded an uncertainty between 2° and 5° for the targeted use case.

TABLE 1
Summary of the Two Initialization Algorithms

Property	Trajectory-fitting algorithm	Gravity analysis algorithm
data needed	IMU	IMU
optional data	GNSS, DVL	none
resolution principle	nonlinear least-square	SVD
main parameter	window size (second)	window size (second)
outputs	roll, pitch, yaw, velocity, accelerometer bias, gyrometer bias, full covariance	roll, pitch, yaw, attitude covariance
processing time for a 60-s window*	67 ms	0.24 ms
total latency for a 60-s window*	60.067 s	60.000 s

*on an Intel Core i7-13800 CPU

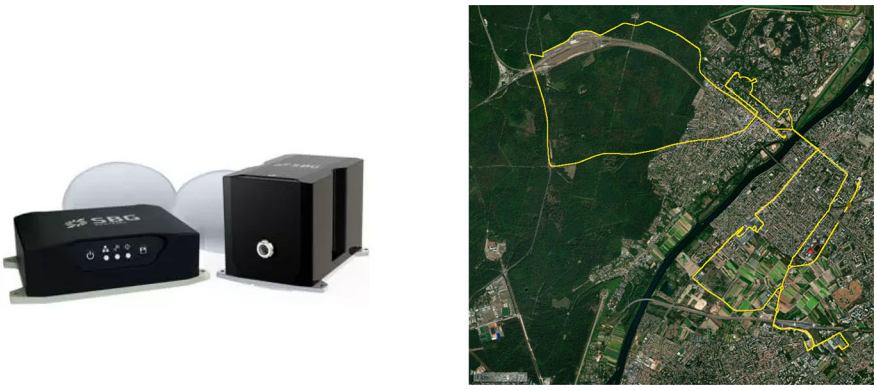

FIGURE 5 Left: Horizon INS (with LCI100 fiber optic gyroscope [FOG] IMU) and GNSS antennas; right: trajectory of a 1-h-long automotive data acquisition performed in Carrieres sur Seine, France, with a Horizon IMU

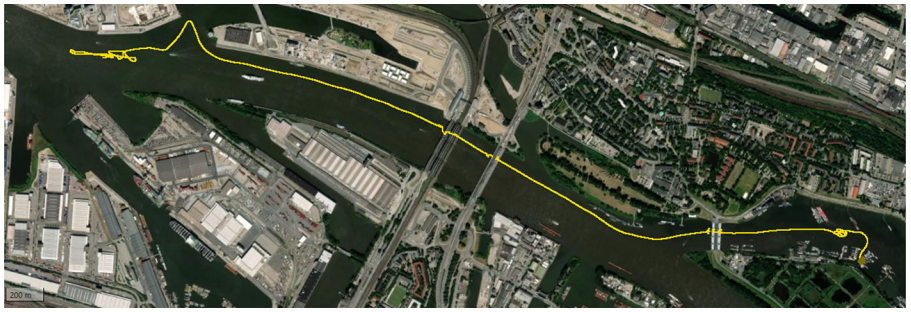

FIGURE 6 Trajectory of a 1-h-long marine data acquisition in Hamburg, Germany, with a Horizon IMU
The first 15 min of the log correspond to a quasi-static situation, with the vessel moored in the harbor.

4.2 | Methodology

To compare the two methods, we launched thousands of trials along the trajectories with 1-s intervals directed forward and backward. This approach artificially doubles the number of our samples compared with a simple forward analysis. We compared the initialization results with a reference solution generated by a tightly coupled inertial GNSS-RTK forward/backward/merge EKF. The EKF heading of the ground truth was initialized using dual GNSS antennas, preventing us from using one of the two proposed methods to initialize the EKF and obtaining a ground truth independent of the presented algorithms.

In addition to comparing the accuracy of the two methods, we sought to highlight the integrity indicators obtained for both methods and their availability. In particular, we found that the proposed SD predictions for both algorithms overbound the Gaussian cumulative distribution function (CDF) up to 3σ (Yan et al., 2025). As our initialization solution could be applied in the future for initializing integrity-assured positioning solutions, we want the user to be able to easily derive the protection level (PL) for their required integrity risk. The Gaussian overbound is a convenient tool for deriving the PL. The principle is that the probability of a result exceeding x times its confidence level must be less than the probability of this event occurring on a Gaussian normal distribution (Yan et al., 2025). Formally, the CDF of the absolute error over its reported SD must overbound the Gaussian normal CDF, as shown in Figure 7. In this context, the complementary

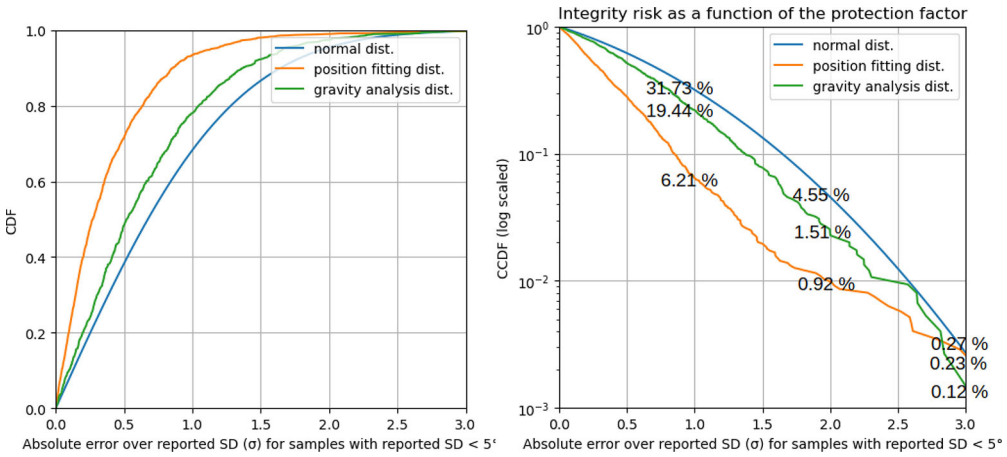

FIGURE 7 CDF and CCDF of the absolute error over the reported SD for *available* samples (reported SD of less than 5°)
These results were derived from the static marine gyrocompassing test (Figures 8 and 9). The CCDF values (integrity risk) at 1σ, 2σ, and 3σ are provided as a percentage and are reported for the different tests in this paper.

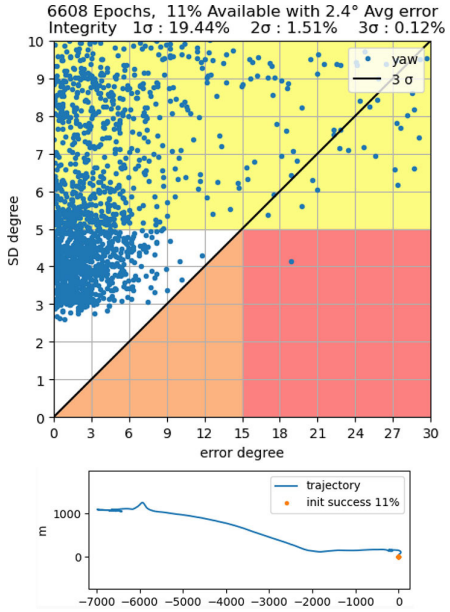

FIGURE 8 Gravity analysis trial for a marine scenario using 60-s data windows

CDF (CCDF) gives the integrity risk as a function of the protection factor, which may be desirable for computing PLs (Figure 7).

The remainder of the analysis was performed by using Stanford diagrams (Tossaint et al., 2006), comparing the actual error with the reported SD. Several values are provided at the top of each graph:

- The availability is computed as the proportion of samples with a predicted heading SD of less than 5°.
- The average error is computed for *available* samples.
- The 1σ, 2σ, and 3σ integrity risks are computed as the error rates greater than 1, 2, and 3 times the predicted yaw SD for *available* samples.

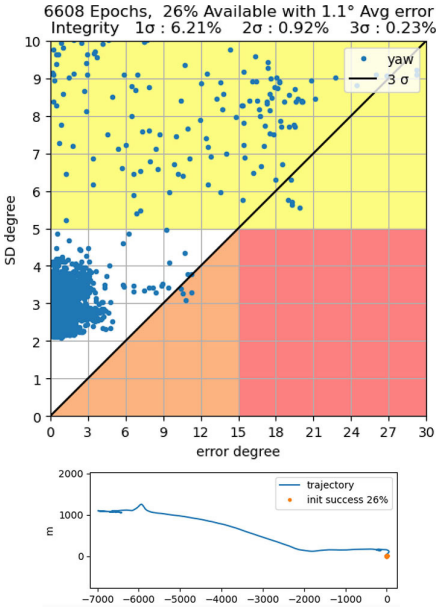

FIGURE 9 Trajectory-fitting trial for a marine scenario using 60-s data windows

In our development, we took care that the rate of large outliers does not exceed the theoretical rate of a normal distribution. The 3σ integrity risk should not exceed 0.3%. For this reason, the 3σ value is used as the diagonal; thus, the scale is adapted accordingly on the x and y axes. The color-coded regions indicate major outliers in the orange and pink regions, whereas unavailable samples are shown in the yellow region and above. In some cases, different clusters detach on the graph, showing different working situations (engine on, engine off, moving, resting, etc.). We can also look for points on the graph that are close to the diagonal of the graph, as these are solutions with high errors that can threaten the integrity of the algorithm.

4.3 | Static Marine Gyrocompassing

The goal of this trial was to test initialization for a moored ship. This scenario takes place in Hamburg on a survey ship. The log begins with 15 min of rest, with the ship moored, followed by 45 min of travel at an average speed of 4 m/s. When using trajectory fitting (Figure 9), we observe a cluster of successful initialization trials around an SD of 3°, with 26% availability, which corresponds to the full time spent at the harbor. When gravity analysis is used (Figure 8), the cluster has a higher reporter SD and error. Hence, only 11% of the solution can be safely used, which corresponds to less than half of the time spent at the harbor. The gravity analysis method struggles to yield the required accuracy for the moored ship within 60 s.

4.4 | Static Automotive Gyrocompassing

In the automotive log, it is crucial that the window period be shorter than the typical time spent at a red traffic light. Otherwise, the system would not be able to find a single sufficiently long static situation within the 1-h log. Consequently, in Figures 10 and 11, only a few percent of points are visible, which correspond to the periods when the car was static for more than 20 s. The remainder of the yaw

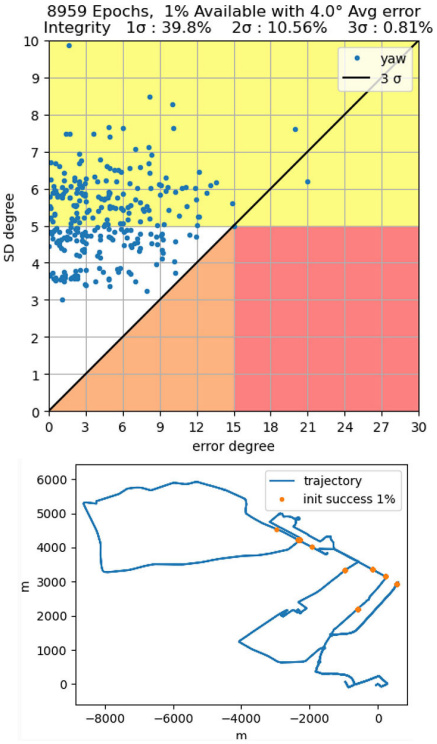

FIGURE 10 Gravity analysis trial for an automotive scenario using 20-s windows

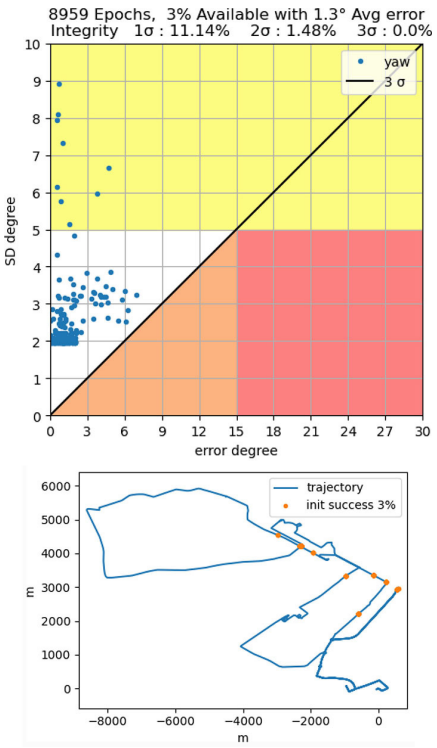

FIGURE 11 Trajectory-fitting trial for an automotive scenario using 20-s windows

samples come from dynamic situations and have too much uncertainty to fit in the Stanford diagrams.

We observe 3% of availability for the trajectory-fitting algorithm, with a cluster at an SD of 2°. This result indicates that the algorithm achieved an accuracy corresponding to the theoretical performance of the gyrometer capability.

In contrast, despite having a carefully chosen data filter, the gravity analysis algorithm only achieves initialization with the required uncertainty in 1% of cases. In fact, gravity analysis shows a degradation of performance in the presence of engine vibrations, whereas the trajectory-fitting method performs similarly regardless of whether the engine is on or off. As shown by the map of initialization success, the gravity-fitting algorithm provided at least one accepted solution for every location at which the trajectory fitting yielded a successful initialization.

4.5 | Dynamic Marine Gyrocompassing

Figure 12 shows the capability of gravity analysis for dynamic ship gyrocompassing. Because gravity analysis is computationally lightweight, it is possible to use a large window of data (10 min in this work). In this situation, the system is capable of initializing most of the time for a boat traveling at approximately 4 m/s. The ability to achieve dynamic gyrocompassing is important for boats, as it may be necessary to initialize the INS without mooring in open sea under poor weather conditions. However, based on the integrity values, the algorithm requires additional protection logic. Indeed, the gravity analysis here is not limited only by noise but also by the initial gyrometer bias uncertainty. Moreover, the covariance estimation in the gravity analysis algorithm does not account for this uncertainty, in contrast to the trajectory-fitting algorithm. As explained in Section 4.1, the accuracy should not exceed 2° because of the unknown initial gyrometer bias uncertainty. Hence, clamping the reported SD to a minimum value of 2° yields more admissible integrity data (1σ: 22.8%, 2σ: 6.45%, 3σ: 0.31%) while keeping the same availability and average error.

Another means for alignment in dynamic situations is to use a GNSS. As shown in Figure 13, the trajectory-fitting algorithm automatically finds the best solution between single-antenna alignment and gyrocompassing alignment. From an operational perspective, if the user requires a heading alignment before the start of a

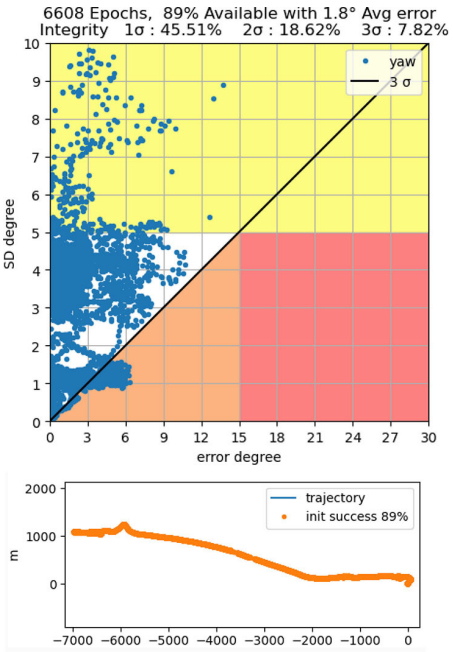

FIGURE 12 Gravity analysis trial for a marine scenario using 10-min data windows

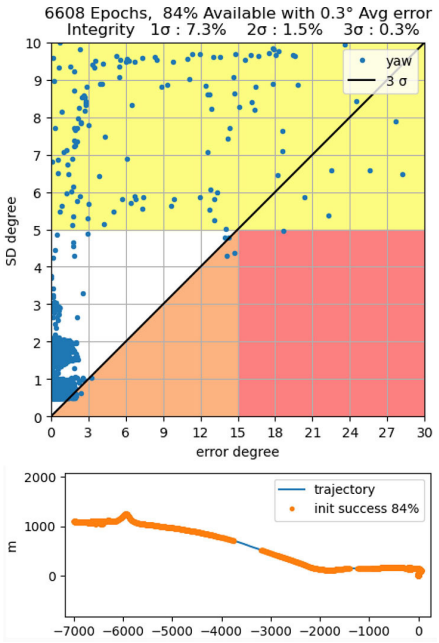

FIGURE 13 Trajectory-fitting trial for a marine scenario using 1-min data windows and GNSS data

mission, they can wait for 1 min for the trajectory-fitting algorithm to gyrocompass; however, if they start the ship immediately, the trajectory-fitting algorithm will find the heading as a result of GNSS displacement, typically within 20 s (Bénet et al., 2024, 2025). Using such joint initialization, we achieve a low error of 0.3° and a broad availability of 84%, which is only reduced by GNSS gaps in the middle of the log caused by the two bridges shown in Figure 6. Here, the heading uncertainty drops well below 2° with high robustness because the heading uncertainty is no longer driven by gyrocompassing but by the GNSS course.

In contrast, dynamic initialization using gravity analysis for automotive scenarios yielded no good solution with a 10-min averaging time, as the dynamics were too harsh. Fortunately, for a ground vehicle, a brief rest of 20 s is not a difficult constraint for initialization. Using the trajectory-fitting algorithm, joint GNSS gyrocompassing initialization remains fully available with a 60-s data window for all situations in which the metric GNSS position is available.

4.6 | MEMS Gyrocompassing Using a DVL

In the latter section, we presented concrete examples in which the trajectory-fitting algorithm showed better performance, on average. Here, we aim to extend the method further to demonstrate gyrocompassing using a non-gyrocompassing IMU, i.e., using an IMU with an initial bias that is too high for static gyrocompassing but with a sufficiently low bias stability to enable its use in dynamic situations. In this case, we used the Apogee MEMS INS (Figure 14), which has an initial bias of approximately 100°/h and a bias instability of 0.05°/h.

The gyrometer bias must be estimated to measure the Earth's rotation. In this context, the first gyrocompassing algorithm of Section 2 fails. To estimate this initial gyrometer bias, movement is required, and this movement must be known. To this end, an additional DVL was employed (Figure 14). This phased-array DVL provides 10-Hz velocity measurements with a mean error of 2 cm/s. The DVL was

FIGURE 14 Top left: Apogee IMU subsea; bottom left: Teledyne Pathfinder DVL; right: 6-h-long acquisition in southern France with an Apogee IMU and a Teledyne DVL

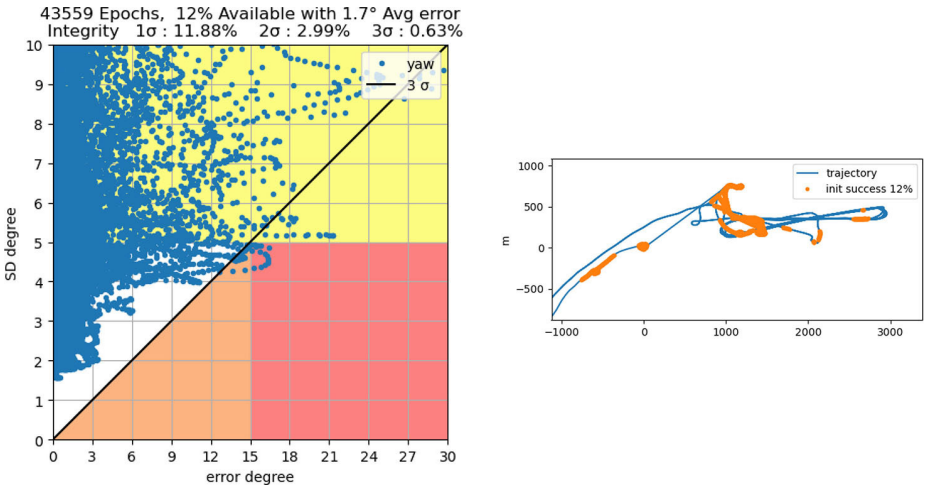

FIGURE 15 MEMS gyrocompassing: Velocity-fitting trial for a marine scenario using 180-s windows of initialization data, with a DVL

mounted facing downward with an azimuth angle of 45°, as recommended by the manufacturer. Because the Apogee INS supports this DVL as an aiding sensor, the INS was used to synchronize and log the DVL outputs. Prior to testing, fine misalignment of the DVL with respect to the INS was estimated using a forward/backward/merge EKF.

In our prior work (Bénet et al., 2024, 2025), we found that by using trajectory fitting in GNSS-aided initialization, the horizontal gyrometer bias can be estimated with an uncertainty of 10°/h in 20 s. Here, using a 180-s window and DVL, the horizontal gyrometer bias can be reduced to 0.3 °/h. This improvement explains why the system is capable of gyrocompassing in this scenario.

Using a general optimization framework to solve the trajectory-fitting problem for initialization enabled us to quickly derive a velocity-fitting algorithm. Because velocities are already computed as sub-steps of computing positions, they can be used instead of positions directly in $f(x)$. Hence, Equations (16), (17), and (19) remain valid in this case.

Figure 15 shows results for a 6-h-long marine log obtained using an SBG Apogee and a Teledyne Pathfinder DVL. The velocity fitting successfully initialized during dynamic movement. When the ship performed turns, the algorithm succeeded

in providing a heading. From an operational perspective, a 3-min-long turning maneuver is required for attitude initialization when using high-grade MEMS sensors if the GNSS is denied.

5 | CONCLUSION

In this paper, we compared a classical gravity analysis algorithm with a trajectory-fitting algorithm for coarse gyrocompassing. To do so, we developed a curated resolution method for acceleration-based gyrocompassing using vector observations. We hope that such an implementation will serve as a baseline for further research in the field. Subsequently, we demonstrated how a trajectory-fitting algorithm, initially developed for GNSS single-antenna alignment, can be used, as is, for gyrocompassing without a GNSS.

Each algorithm presented here provides a robust and tight estimation of the respective uncertainty in all situations. Hence, even if an unaided initialization procedure is processed on a window that is too small for the dynamic encountered, the output uncertainty will increase and prevent the system from starting with a poor initialization point.

Table 2 summarizes the different tests carried out in this study. The car test is very close to the initialization procedure of an airplane on a tarmac, where the airplane remains still for a short period, allowing the INS to initialize using a measurement of Earth's rotation (Titterton & Weston, 2004). In this context, external perturbations such as wind and engine noise should be considered to avoid erroneous alignment (Titterton & Weston, 2004). For this test, the trajectory-fitting alignment provided three-fold greater availability than the gravity analysis, as the latter did not perform well in the presence of engine noise. In this scenario, using a larger window of 30 s did not improve the results, because there were far fewer periods when the car stayed still for 30 s than for 20 s.

Although gravity analysis arises from early work in the field, it remains highly efficient and computationally lightweight for marine scenarios (the use cases for which it was designed). Using a larger window of 90 s for marine scenarios allowed the gravity analysis to successfully perform gyrocompassing throughout the entire period in the harbor (with an availability of 24%). Moreover, as the gravity analysis has a lower computational cost than the trajectory-fitting algorithm, it can be used for very long averaging periods of, for example, 10 min. Thus, the gravity analysis algorithm is the only algorithm presented here capable of unaided initialization for a cruising ship.

TABLE 2
Summary of Gyrocompassing Tests Performed in This Work
RMSE: root mean squared error

Use case	Time	Gravity analysis algorithm		Trajectory-fitting algorithm	
		Availability	RMSE	Availability	RMSE
automotive	20 s	1%	4.0°	3%	1.3°
automotive	30 s	1%	2.2°	1%	0.7°
marine	60 s	11%	2.4°	26%	1.1°
marine	90 s	24%	1.6°	27%	0.7°
marine	600 s	89%	1.8°	-	-
marine + GNSS	60 s	-	-	84%	0.3°
MEMS + DVL	180 s	-	-	12%	1.7°

Both algorithms performed similarly for the typical use case of quasi-static gyrocompassing. However, the trajectory-fitting algorithm consistently outperformed gravity analysis in terms of accuracy and availability when the time period was constrained. The trajectory-fitting algorithm is also more versatile, as it can perform course alignment when available and automatically fall back to gyrocompassing when no dynamics are encountered. This algorithm can also work with low-cost MEMS to high-grade FOG sensors, providing the most competitive results in all cases.

Of course, this effectiveness comes with a higher computational burden. In real-time processing, for the trajectory-fitting algorithm, an additional mechanization procedure can be used to bring the result from 67 ms in the past (Table 1) to the current time or the output SD can be increased to make the result valid in a real-time context. This concern becomes more important if this delay is greater because of the use of a lightweight embedded processor or larger sliding windows.

REFERENCES

Bénet, P., Saidani, M., & Guinamard, A. (2024). Robust single antenna INS initialization for low dynamic applications. *Proc. of the 2024 International Technical Meeting of the Institute of Navigation*, Long Beach, CA, 1021–1032. https://doi.org/10.33012/2024.19530

Bénet, P., Saidani, M., & Guinamard, A. (2025, April). *Method for aligning an inertial navigation system* [US Patent App. 18/924,69] U.S. Patent and Trademark Office. https://patents.google.com/patent/US20250130072A1

Che, Y., Wang, Q., Gao, W., & Yu, F. (2015). An improved inertial frame alignment algorithm based on horizontal alignment information for marine SINS. *Sensors*, *15*(10), 25520–25545. https://doi.org/10.3390/s151025520

Davenport, P. B. (1968). *A vector approach to the algebra of rotations with applications* (tech. rep. No. D-4696). National Aeronautics and Space Administration. https://ntrs.nasa.gov/citations/19680021122

Fletcher, R. (2000). *Practical methods of optimization*. John Wiley & Sons. https://doi.org/10.1002/9781118723203

Gade, K. (2016). The seven ways to find heading. *The Journal of Navigation*, *69*(5), 955–970. https://doi.org/10.1017/S0373463316000096

Gaiffe, T., Cottreau, Y., Faussot, N., Hardy, G., Simonpietri, P., & Arditty, H. (2000). Highly compact fiber optic gyrocompass for applications at depths up to 3000 meters. *Proc. of the 2000 International Symposium on Underwater Technology* (*Cat. No. 00EX418*), 155–160. https://doi.org/10.1109/UT.2000.852533

Gu, D., El-Sheimy, N., Hassan, T., & Syed, Z. (2008). Coarse alignment for marine SINS using gravity in the inertial frame as a reference. *Proc. of the 2008 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, Monterey, CA, 961–965. https://doi.org/10.1109/PLANS.2008.4570038

Lacambre, J.-B., Le Breton, L., & d'Harcourt, P. (2020). Why inertial systems need to be aligned and how to do this effectively. *Saint-Germain-en-Laye, France: iXblue SAS*. https://www.exail.com/media-file/10675/exail-white-paper-why-inertial-systems-need-to-be-aligned-and-how-to-do-this-effectively.pdf

Li, Q., Ben, Y., & Sun, F. (2013). A novel algorithm for marine strapdown gyrocompass based on digital filter. *Measurement*, *46*(1), 563–571. https://doi.org/10.1016/J.MEASUREMENT.2012.08.015

Liu, X., Xu, X., Zhao, Y., Wang, L., & Liu, Y. (2014). An initial alignment method for strapdown gyrocompass based on gravitational apparent motion in inertial frame. *Measurement*, *55*, 593–604. https://doi.org/10.1016/j.measurement.2014.06.004

Liu, Y., Xu, X., Liu, X., Yao, Y., Wu, L., & Sun, J. (2015). A self-alignment algorithm for SINS based on gravitational apparent motion and sensor data denoising. *Sensors*, *15*(5), 9827–9853. https://doi.org/10.3390/s150509827

Liu, Y., Yu, R., Xiong, Z., & Guo, Y. (2022). Research on algorithms for multi-vector attitude determination. *Mathematical Problems in Engineering*, *2022*(1), 6137308. https://doi.org/10.1155/2022/6137308

Markley, F. L. (1988). Attitude determination using vector observations and the singular value decomposition. *Journal of the Astronautical Sciences*, *36*(3), 245–258. https://www.researchgate.net/publication/243753921_Attitude_Determination_Using_Vector_Observations_and_the_Singular_Value_Decomposition

Markley, F. L., & Mortari, D. (2000). Quaternion attitude estimation using vector observations. *The Journal of the Astronautical Sciences*, *48*, 359–380. https://doi.org/10.1007/BF03546284

Pei, F., Xu, H., Jiang, N., & Zhu, D. (2020). A novel in-motion alignment method for underwater SINS using a state-dependent Lie group filter. *NAVIGATION*, *67*(3), 451–470. https://doi.org/10.1002/navi.387

Pei, F., Yang, S., Zhu, D., & Yin, S. (2020). A novel coarse alignment method for SINS using special orthogonal group optimal estimation. *Sensors*, *20*(20), 5740. https://doi.org/10.3390/s20205740

Silson, P. M. (2011). Coarse alignment of a ship's strapdown inertial attitude reference system using velocity loci. *IEEE Transactions on Instrumentation and Measurement*, *60*(6), 1930–1941. https://doi.org/10.1109/TIM.2011.2113131

Sun, F., Lan, H., Yu, C., El-Sheimy, N., Zhou, G., Cao, T., & Liu, H. (2013). A robust self-alignment method for ship's strapdown INS under mooring conditions. *Sensors*, *13*(7), 8103–8139. https://doi.org/10.3390/s130708103

Titterton, D., & Weston, J. L. (2004). *Strapdown inertial navigation technology* (Vol. 17). IET. https://doi.org/10.1049/PBRA017E

Tossaint, M., Samson, J., Toran, F., Ventura-Traveset, J., Sanz, J., Hernandez-Pajares, M., & Juan, J. (2006). The Stanford-ESA integrity diagram: Focusing on SBAS integrity. *Proc. of the 19th International Technical Meeting of the Satellite Division of the Institute of Navigation* (*ION GNSS 2006*), Fort Worth, TX, 894–905. https://www.ion.org/publications/abstract.cfm?articleID=6963

Wahba, G. (1965). A least squares estimate of satellite attitude. *SIAM review*, *7*(3), 409–409. https://doi.org/10.1137/1008080

Xu, X., Xu, X., Zhang, T., Li, Y., & Wang, Z. (2017). A coarse alignment method based on digital filters and reconstructed observation vectors. *Sensors*, *17*(4), 709. https://doi.org/10.3390/s17040709

Yan, P., Zhong, Y., & Hsu, L.-T. (2025). Principal Gaussian overbound for heavy-tailed error bounding. *IEEE Transactions on Aerospace and Electronic Systems*, *61*(1), 829–852. https://doi.org/10.1109/TAES.2024.3448405

Zhao, L., Guan, D., Cheng, J., Xu, X., & Fei, Z. (2016). Coarse alignment of marine strapdown INS based on the trajectory fitting of gravity movement in the inertial space. *Sensors*, *16*(10), 1714. https://doi.org/10.3390/s16101714

Zhu, Y., & Qiao, P. (2019). Coarse alignment of strapdown inertial navigation system based on LMS adaptive filtering. In *2019 IEEE 4th International Conference on Image, Vision and Computing* (*ICIVC*), 701–705. https://doi.org/10.1109/ICIVC47709.2019.8980991

How to cite this article: Bénet, P. & Guinamard, A. (2026). Gyrocompassing: From gravity rotation analysis to inertial trajectory fitting. *NAVIGATION, 73*. https://doi.org/10.33012/navi.751

APPENDIX

A | RESULTS FOR A LARGER WINDOW SIZE

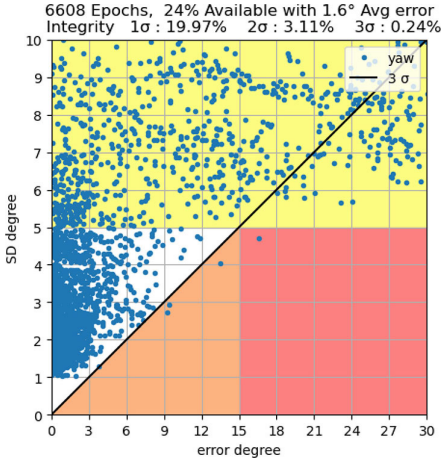

FIGURE 16 Gravity analysis trials for a marine scenario with larger data windows (90 s)

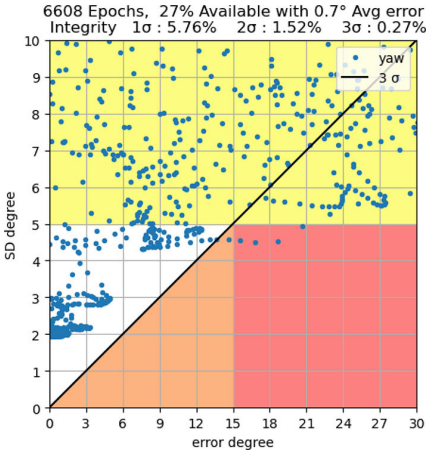

FIGURE 17 Trajectory-fitting trials for a marine scenario with larger data windows (90 s)

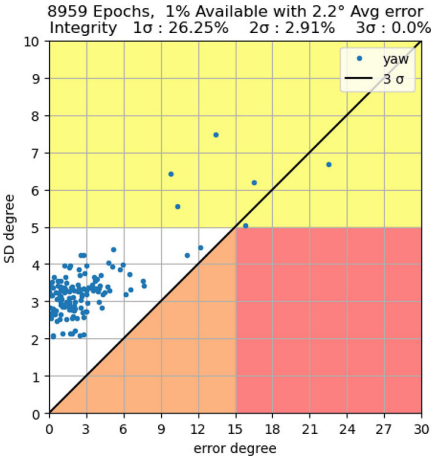

FIGURE 18 Gravity analysis trials for an automotive scenario with larger data windows (30 s)

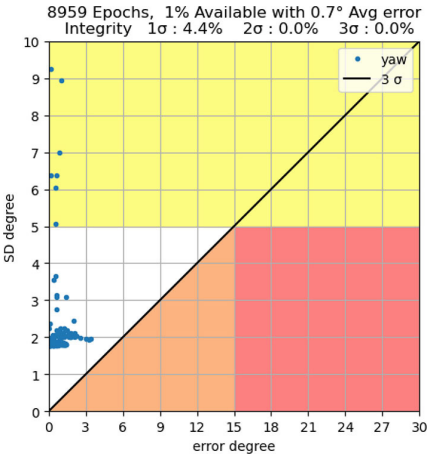

FIGURE 19 Trajectory-fitting trials for an automotive scenario with larger data windows (30 s)