

Article

Real-Time Data Acquisition System for Array MIMU Based on FPGA+ARM

Xiaoyang Qin ^{1,2}, Huan Wang ^{1,2}, Zhihua Dai ³, Yonghua Wang ⁴, Junqing Zhang ⁵, Tao Guo ^{1,2} and Huiliang Cao ^{6,7,*}

¹ State Key Laboratory of Extreme Environment Optoelectronic Dynamic Measurement Technology and Instrument, North University of China, Taiyuan 030051, China

² School of Instrument and Electronics, North University of China, Taiyuan 030051, China

³ Quanzhou Yunjian Measurement Control and Perception Technology Innovation Research Institute, Quanzhou 362000, China

⁴ School of Civil Engineering, Tangshan University, Tangshan 063000, China

⁵ Tangshan Zhongyu Technology Development Co., Ltd., Tangshan 063000, China

⁶ School of Integrated Circuits and Electronics, Beijing Institute of Technology, Beijing 100081, China

⁷ Bit Chongqing Institute of Microelectronics and Microsystems, Chongqing 400000, China

* Correspondence: caohuiliang@bit.edu.cn

Abstract

To address the issue of low accuracy and stability in the gyroscope components of the micro-inertial-measurement-unit (MIMU) core units, which limits their application in high-precision scenarios, this paper designs a real-time data acquisition system for array MIMU based on FPGA and ARM. This system establishes a complete data chain encompassing raw data acquisition, real-time processing, multi-source information fusion, data storage, and communication with a host computer. It has been successfully applied to a 100-m pipeline position coordinate measurement scenario. The paper begins by discussing the overall system design, including both hardware circuit and software code development. Attitude update algorithms and measurement accuracy evaluation metrics are also introduced. System functionality is validated through static tests and practical pipeline measurements. Experimental results demonstrate that the system improves the accuracy of a single micro-electro-mechanical system (MEMS) gyroscope by a factor of 7.4 to 7.7. It also enables precise calculation of the pipeline position coordinates over the 100 m distance, achieving a horizontal positioning error of less than 0.0774 m and an elevation positioning error of less than 0.0351 m. These results fully confirm the significant effectiveness of the array design in mitigating gyroscope random errors, providing a reliable technical solution for pipeline measurement.

Keywords: MIMU; array-based; real-time; FPGA; ARM; SPI protocol



Academic Editor: Erwin Peiner

Received: 1 December 2025

Revised: 19 December 2025

Accepted: 20 December 2025

Published: 12 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Within inertial navigation systems, the micro-inertial-measurement-unit (MIMU) serves as the core component across numerous navigation systems. It incorporates a gyroscope and accelerometer on each of three orthogonal axis, typically employed to measure angular velocity and acceleration during navigation [1–3]. Compared to conventional mechanical sensors, the MIMU has garnered significant attention in recent years due to its advantages of low cost, compact size, and ease of integration, leading to substantial market demand. Consequently, it has become an essential component in diverse navigation systems, attitude control devices, unmanned aerial vehicles, robotic navigation, and satellite systems [4–6]. Despite their extensive applications, MIMUs suffer from relatively low

precision and stability due to the frequent influence of environmental, electronic, and manufacturing noise on micro-electro-mechanical system (MEMS) gyroscopes. This limitation restricts their use in scenarios demanding high accuracy.

To enhance the accuracy of gyroscope components within MIMUs, optimisation of manufacturing processes and digital signal processing techniques are typically employed. However, improving the precision of micro-mechanical structures through process refinement significantly increases sensor production costs, and substantial accuracy gains via structural modifications prove challenging. Therefore, forming an array circuit by calibrating and compensating MEMS gyroscopes, followed by data fusion to achieve higher measurement accuracy, represents a cost-effective approach. This technique, which constructs high-precision MEMS gyroscopes through array formation using MEMS gyroscopes and data fusion technology while simultaneously reducing drift noise, is termed ‘virtual gyroscope’ technology [7–10].

Gyroscope array technology was first proposed by Bayard et al. at the US Jet Propulsion Laboratory (JPL) and termed the ‘virtual gyroscope’ [11]. In 2003, Bayard et al. pioneered a virtual gyroscope technique that fused data from four identical MEMS gyroscopes. Experimental validation demonstrated the method’s feasibility, achieving approximately double the accuracy. They further concluded that when N identical yet uncorrelated gyroscopes collect data under identical conditions, the drift resulting from fusing their data reduces to $\sqrt{N}$ times the original value. Kamata et al. proposed a small FPGA-based array board comprising 32 MEMS gyroscopes, reducing bias stability from $2.2^\circ/\text{h}$ to $0.5^\circ/\text{h}$ [12]. Wang et al. employed a gyro array of 72 MPU-6000 sensors, achieving approximately 50 reduction in angular error after 20 runs over one hour—effectively halving the fusion error Kamata2020. Dominguez et al. designed and implemented a differential IMU array using 16 ASM330 MEMS sensors, yielding a high-precision virtual gyroscope output [13]. Domestically, Yi et al. designed and implemented a differential IMU array comprising 16 ASM330 MEMS sensors, achieving a high-precision virtual gyroscope output [14]. However, the aforementioned research circuits predominantly employ ARM processors as microcontrollers, sequentially reading sensor data. This sequential approach introduces temporal differences, making it challenging to ensure synchronous data measurement timing. Presently, controllers in data acquisition systems predominantly employ ARM processors, which execute tasks sequentially by processing commands one by one. When utilised as the primary controller for reading gyroscope array data, the inherent programme execution delay makes it challenging to obtain data from the gyroscope array at precisely the same moment.

Therefore, in order to enhance the performance of MEMS gyroscopes and enable simultaneous reading of multiple sensors, this paper designs and implements a real-time signal acquisition system for MIMU arrays based on FPGA+ARM architecture, applied to pipeline measurement scenarios. The system employs FPGA and ARM working in tandem, with the FPGA synchronously acquiring and fusing 60-channel MIMU data via SPI bus. Raw acquisition data is transmitted to the ARM processor through a serial port, where the ARM handles data parsing, multi-source information fusion, data storage, and external communication interactions.

The remaining structure of this paper is organised as follows: Section 2 establishes the overall design scheme for the array MIMU real-time data acquisition system, covering core elements such as hardware circuit design and software code development. Section 3 elaborates on the quaternion-based attitude update method and position update method, introducing relevant evaluation metrics. Section 4 validates and analyses the proposed data acquisition system through static experiments and practical pipeline scenario testing. Section 5 summarises the entire paper and outlines future research directions.

2. Overall Design of the Acquisition System

This chapter shall present the design rationale for the two core components of the array MIMU system, encompassing both hardware circuitry and software code design. Adopting a top-level architectural concept of ‘distributed acquisition, centralised processing, and unified transmission’, the objective is to establish a high-precision hardware measurement system capable of synchronously acquiring multiple MIMU data streams.

2.1. Critical Processor Selection and Justification

Currently, mainstream MIMUs predominantly support the SPI communication protocol, necessitating the selection of a main controller capable of driving multiple SPI interfaces in parallel. Traditional MCUs typically employ analogue switches to toggle chip select signals for time-division multiplexing of SPI interfaces. This approach is prone to data blocking and transmission delays, severely compromising system synchronisation accuracy and real-time performance, and proving inadequate for synchronous reading of multiple MIMUs. Conversely, FPGAs leverage programmable logic resources to instantiate multiple independent SPI controller cores via software programming, enabling simultaneous data acquisition from multiple MIMUs.

2.2. System Hardware Circuit Design

The MIMU array system design comprises multiple modules, with its structural block diagram illustrated in Figure 1. This figure presents an array MIMU data acquisition system based on FPGA+ARM, encompassing the MIMU data acquisition module (based on the SPI bus), magnetometer data acquisition module (based on the I²C bus), STM32F4 and FPGA main control modules, SD NAND Flash storage module, and RS422 communication module. Regarding hardware layout, considering signal integrity and PCB size constraints, the system employs a split-board design: the 60-channel MIMU is integrated onto one PCB. The FPGA is deployed independently on a separate PCB, responsible for synchronous acquisition and real-time averaging processing of the 60-channel MIMU data. While the STM32F4 is configured on another independent PCB, undertaking core functions such as data updating, storage, and transmission.

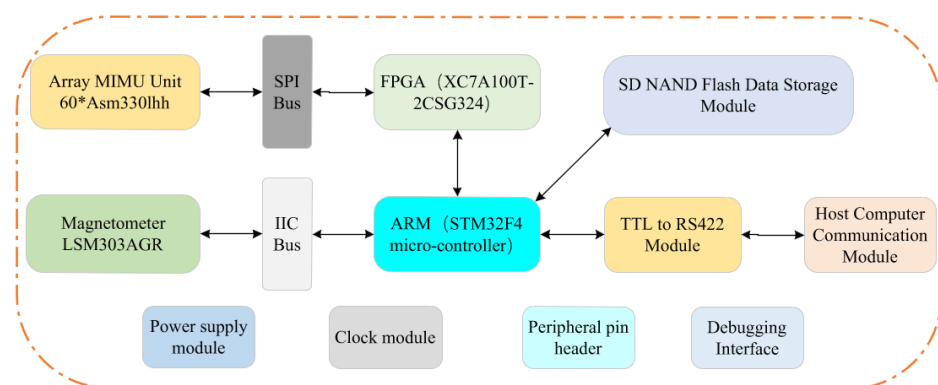


Figure 1. Hardware design of an array MIMU data acquisition system based on FPGA+ARM.

The MIMU array comprises sixty ASM330 sensors, each capable of capturing triaxial angular velocity and triaxial acceleration data. All data is synchronously transmitted back to the FPGA for processing via the SPI communication interface’s master-to-slave input (MOSI) line. Operating within a 3.3V voltage range, the system employs a TLV75533PDBVR voltage regulator to convert 5 V power to 3.3 V. Capacitors are utilised to filter out alternating frequency components within the DC voltage.

For the designed array system, simultaneous reading of 60 sensors requires at least 60 GPIO ports for the MOSI lines. Regarding the master-to-slave output (MISO) line, clock (CLK) line, and chip select (CS) line, since synchronous reading of 60 sensors is required, a single MISO, CLK, and CS line can synchronously control all 60 sensors. Considering that the MISO, CLK, and CS lines must simultaneously drive 60 sensors, the SN74AUP3G34 buffer is employed here to enhance the driving capability of these lines and ensure signal integrity.

The SPI bus topology employed in this design is illustrated in Figure 2a. This configuration connects all MIMU MISO, CLK, and CS lines in parallel to corresponding pins on the FPGA. The FPGA's MOSI line is broadcast-connected to all MIMU MISO lines, enabling unified transmission of command data. Concurrently, each MIMU's MOSI line connects independently to a dedicated FPGA pin, thereby forming 60 distinct parallel data return channels. This topology ingeniously integrates broadcast write and parallel read mechanisms: during the write phase, the FPGA synchronously transmits configuration commands or read instructions to all MIMUs via the shared MOSI line. During the data read phase, the FPGA can simultaneously capture data from all 60 MIMUs, timing diagram as shown in Figure 2b. Within the same clock cycle, it performs data acquisition on multiple sensors concurrently, ensuring the system's real-time capability.

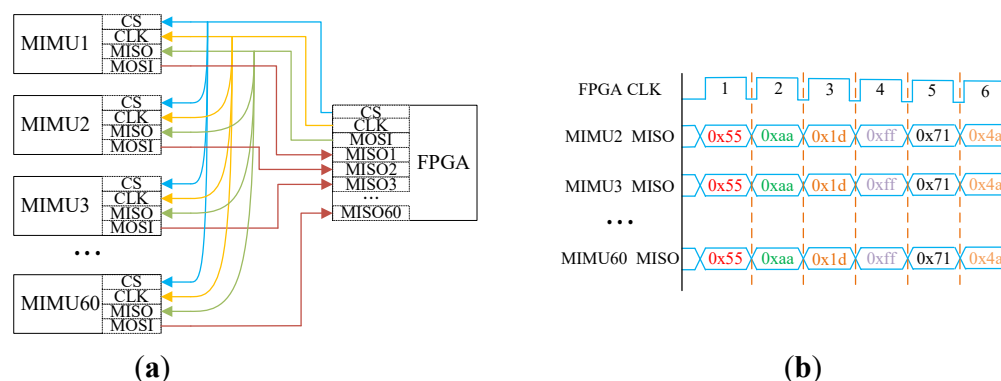


Figure 2. SPI bus topology of the array MIMU and SPI timing diagram for FPGA: (a) SPI bus topology of the array MIMU; (b) SPI timing diagram for FPGA.

Compared to traditional daisy-chain or star topologies, this design significantly optimises the real-time parallel data acquisition capability of multi-sensor systems while fully utilising FPGA pin resources.

To further enhance the performance of the fused virtual MIMU while balancing system routing complexity and manufacturing costs, the MIMU array design integrates 60 sensors in a spatially symmetric configuration, as illustrated in Figure 3. Thirty sensors are arranged on each side of the PCB, with their Z-sensitive axis oriented in mirror symmetry. Within the same board surface, the sensors are further divided into upper and lower groups of 15 each, with their X and Y sensitive axes arranged in vertical symmetry. This configuration not only effectively mitigates the impact of interference such as temperature drift, board vibration, and circuit noise on MIMU measurement results, thereby improving test accuracy, but also provides ample measurement information to enhance system reliability.

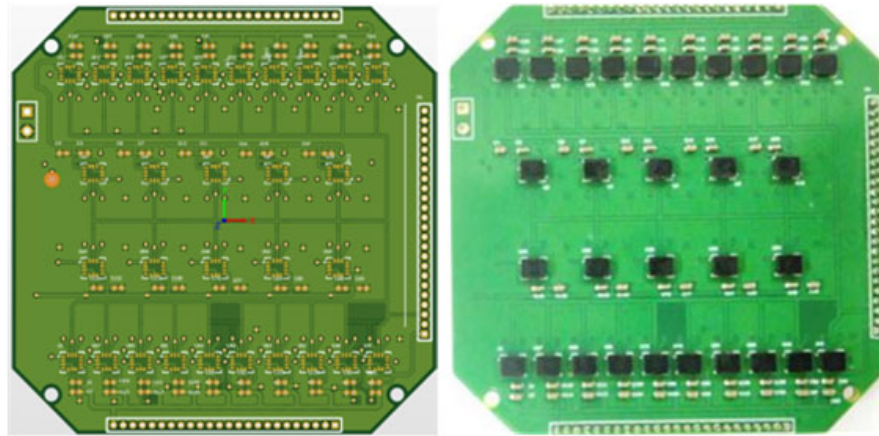


Figure 3. Schematic diagram of front layout and physical image of the array MIMU PCB.

2.3. Software Code Design

The software development for the entire MIMU array system employs a dual-platform architecture based on Vivado and STM32CubeIDE. Specifically, the FPGA logic and control programmes are developed within the Vivado environment to achieve synchronised acquisition and data processing across multiple MIMUs. Meanwhile, the embedded programmes for the ARM processor rely on the STM32CubeIDE environment, responsible for tasks including data reception, attitude angle initialisation, quaternion iteration calculations, updating attitude angles, and data storage. The two components exchange data via the UART serial communication protocol, forming a complete heterogeneous processing system. The software programme execution flow is illustrated in Figure 4.

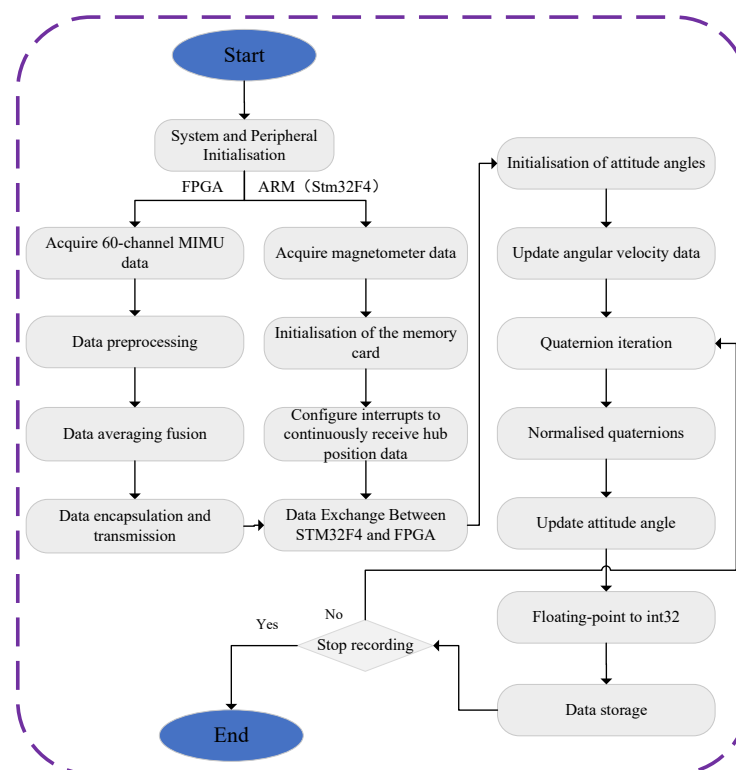


Figure 4. MIMU array system software code flowchart.

The FPGA algorithm centres on hardware parallelism to achieve high-synchronisation data acquisition and primary fusion for 60 MIMUs: upon system power-up, the FPGA first configures the global clock and initialises the custom SPI controller IP core. This IP

core generates 60 independent SPI control signals to drive each MIMU sensor in parallel. Subsequently, this IP core transmits synchronised acquisition commands to all MIMUs in parallel and receives raw data. The hardware-level synchronisation mechanism eliminates the time lag inherent in traditional sequential acquisition via MCU, ensuring consistent data timestamps. The FPGA then validates the raw data and performs real-time averaging fusion of all 60 measurement channels at the hardware level, generating a high signal-to-noise ratio data stream that substantially reduces the computational load on the backend. Finally, the pre-processed fused data is encapsulated into fixed-length frames and transmitted via UART at 921,600 bps to the STM32F4 microcontroller (Manufacturer: STMicroelectronics; Corporate Location: Geneva, Switzerland).

The ARM receives the fused data from the FPGA and, combined with magnetometer information, performs attitude calculation and storage: the UART interrupt service routine parses the data frame, extracts the fused angular velocity and acceleration data, then fuses the triaxial magnetometer data to complete attitude initialisation (calculating the initial attitude quaternion using static accelerometer and magnetometer data). Subsequently, the quaternion is updated in real-time based on the fused angular velocity and undergoes normalisation processing. This converts it into roll, pitch, and yaw angles. The floating-point attitude angles are then converted to int32 format to accommodate the storage format. Finally, the attitude angles, carrier travel distance, and raw inertial data are written to the storage chip, forming a complete closed-loop process of ‘data acquisition-processing-storage’.

3. Theoretical Research

3.1. Quaternion-Based Method for Attitude and Position Update

Defining the carrier coordinate system as b and the navigation coordinate system as n , the coordinate transformation matrix C_b^n from b to n is termed the attitude transformation matrix. As the MIMU is rigidly attached to the carrier, all measured attitude data are based on the carrier coordinate system. Therefore, during attitude estimation, the attitude update matrix C_b^n is required to convert data from the carrier coordinate system to the navigation coordinate system. In fields such as attitude estimation and navigation, quaternions are widely employed due to their advantages over Euler angles, including avoidance of gimbal lock issues, higher computational efficiency, and reduced storage requirements [15–17]. The following outlines the derivation process for solving quaternions used in gyroscope-based attitude updates. First, the mathematical definition of quaternions is as follows:

$$Q = q_0 + \mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} = \cos \frac{\theta}{2} + \mathbf{u}^n \sin \frac{\theta}{2}. \quad (1)$$

In the equation, q_0, q_1, q_2, q_3 denote real numbers, while $\mathbf{i}, \mathbf{j}, \mathbf{k}$ represent the unit vector $\mathbf{u}^n$ indicating the rotation axis and direction, with θ denoting the rotation angle.

The angular velocity information output by the gyroscope is expressed as in Equation (2):

$$\boldsymbol{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T. \quad (2)$$

The relationship between quaternions and angular velocity may be expressed by the following equation:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}. \quad (3)$$

Solving Equation (3) yields the gyroscope-updated quaternion $q_\omega = [q_0, q_1, q_2, q_3]^T$. Substituting this into Equation (4) produces the attitude update matrix C_b^n :

$$C_b^n = \begin{bmatrix} 1 - 2(q_2^2 + q_3^2) & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & 1 - 2(q_1^2 + q_3^2) & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & 1 - 2(q_1^2 + q_2^2) \end{bmatrix}. \quad (4)$$

Equation (4) is denoted as Equation (5):

$$C_b^n = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix}. \quad (5)$$

Based on the conversion relationship between quaternions and euler angles, the attitude angles can be determined as shown in Equation (6):

$$\begin{cases} \phi = \arctan\left(\frac{T_{12}}{T_{22}}\right) \\ \theta = \arcsin(T_{32}) \\ \gamma = \arctan\left(-\frac{T_{31}}{T_{33}}\right) \end{cases}. \quad (6)$$

In the equation, ϕ , θ , and γ denote the heading angle, pitch angle, and roll angle respectively.

By mounting the measurement system described herein onto a moving carrier, with the positive direction of its X-axis accelerometer defined as the carrier's forward direction, the displacement of the carrier along the X, Y, and Z axis directions can be determined by combining the carrier's forward distance with its attitude angle information.

$$\begin{cases} \Delta X = \cos(\Delta\theta) \cdot \cos(\Delta\phi) \cdot \Delta d \\ \Delta Y = \cos(\Delta\theta) \cdot \sin(\Delta\phi) \cdot \Delta d \\ \Delta Z = \sin(\Delta\theta) \cdot \Delta d \end{cases} \quad (7)$$

where Δd denotes the incremental forward distance, representing the difference between the current actual forward distance and the last stored actual forward distance; $\Delta\theta$ denotes the incremental pitch angle, representing the difference between the current actual pitch angle and the last stored actual pitch angle; $\Delta\phi$ denotes the incremental heading angle, representing the difference between the current actual heading angle and the last stored actual heading angle; ΔX , ΔY , and ΔZ denote coordinate increments, representing the displacement changes in the three axes within the global coordinate system.

3.2. Zero-Bias Stability

Bias stability refers to the degree of random fluctuation in the output angular velocity or acceleration of an inertial sensor when stationary and free from interference. It constitutes a crucial performance metric for inertial sensors, reflecting their precision and reliability. Bias stability is typically expressed as the standard deviation of the output angular velocity over a specified time period. The output data of the inertial sensor is averaged using a smoothing time of 10 s. The specific calculation formula is as follows:

$$B_s = \frac{1}{K} \sqrt{\frac{1}{N-1} \sum_{i=1}^N (\omega_i - \omega)^2}, \quad (8)$$

$$\omega = \frac{1}{N} \sum_{i=1}^N \omega_i. \quad (9)$$

Herein, B_s represents the zero bias stability of the inertial sensor; K denotes the calibration factor of the inertial sensor; N signifies the number of data points smoothed over the specified time period; ω_i denotes the i -th data output from the inertial sensor following smoothing over the specified time period; ω and denotes the average value of the angular velocity or acceleration output by the inertial sensor.

3.3. Allan Variance

The Allan variance double-logarithmic curve plots the square root of Allan variance against the ratio of sampling time block length on a double-logarithmic coordinate system, revealing noise characteristics across different timescales. The specific calculation steps are as follows:

Step 1: Divide the entire data set into N blocks based on window length τ , with each block containing n data points and no overlap between blocks;

Step 2: Calculate the mean for the n data points within each block:

$$\bar{y}_k = \frac{1}{n} \sum_{i=k}^{k+n-1} y_i. \quad (10)$$

Step 3: Calculate the difference between adjacent blocks and compute the RMS value:

$$\sigma^2(\tau) = \frac{1}{2(N_c - 1)} \sum_{k=1}^{N_c-1} (\bar{y}_{k+1} - \bar{y}_k)^2. \quad (11)$$

Step 4: Alter the value of n , repeat Steps 1–3, and plot a double-logarithmic curve showing the Allan standard deviation versus block length, which constitutes the Allan variance curve.

Based on the slope of the Allan variance curve, five typical error noise components can be identified as shown in Table 1: Quantisation Noise (QN), Angular Random Walk (ARW), Bias Instability (BI), Rate Random Walk (RRW), and Rate Ramp (RR). QN originates from errors during analogue-to-digital conversion, manifesting as subtle signal distortion. ARW primarily describes gyroscope angular drift caused by random noise over extended integration periods, constituting one of the principal error sources limiting sensor accuracy. BI typically presents as low-frequency noise. RRW denotes the random drift in the gyroscope's angular velocity output caused by white noise; RR constitutes a deterministic error, exhibiting a time-dependent phenomenon primarily induced by the drift of physical parameters within the sensor's internal structural components [18,19].

Table 1. Five typical types of random error in Allan variance.

Noise Type	Symbol	Unit	Allan Standard Deviation
QN	Q	urad	$\sigma_{QN}(\tau) = \sqrt{3}Q/\tau$
ARW	N	$^\circ/\sqrt{h}$	$\sigma_{ARW}(\tau) = N/\sqrt{\tau}$
BI	B	$^\circ/h$	$\sigma_B(\tau) = \sqrt{2 \ln 2 / \pi} B$
RRW	K	$^\circ/h\sqrt{h}$	$\sigma_{RRW}(\tau) = \sqrt{\tau/3} K$
RR	R	$^\circ/h^2$	$\sigma_R(\tau) = \tau R / \sqrt{2}$

4. Experimental Validation and Analysis

This experiment employs the SPI interface of an FPGA to communicate with the MIMU, transmitting data to an STM32F4 microcontroller. The STM32F4 parses, processes, and stores this data for subsequent retrieval. Upon completion of the test, commands are sent from the host computer to the microcontroller, which then transmits the stored data

back via the RS422 bus. Data analysis is subsequently performed on the PC, forming a complete experimental system. The board-level circuit assembly platform for this experimental system is illustrated in Figure 5. Testing is conducted on a high-precision turntable, initially under static conditions followed by dynamic pipeline testing in realistic environments to validate the effectiveness of the measurement system. Throughout this process, the system operated at a supply voltage of 5 V, a current of 0.12 A, and a power consumption of 0.6 W.

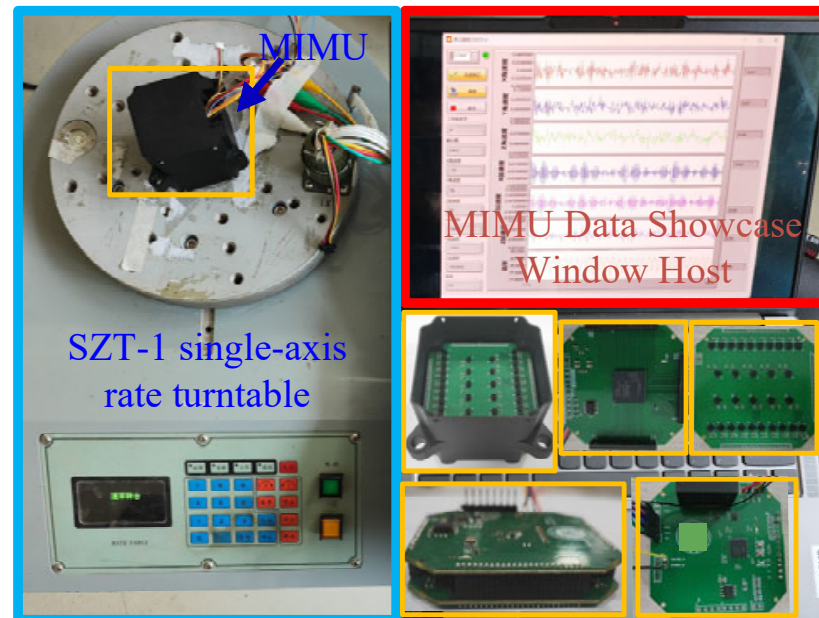


Figure 5. Test diagram of turntable for array MIMU acquisition system.

4.1. Static Test Results

To evaluate the performance of the virtual gyroscope realised by the designed circuit, the MIMU data acquired from static experiments are first analysed. The experimental setup employed a sampling rate of 100 Hz, recording angular velocity data from both individual gyroscopes and the fusion of 60 gyroscopes. Angular velocity data spanning two hours of testing are extracted for processing.

A visual comparison of the X, Y, and Z-axis angular velocity data from both individual gyroscopes and the fusion of 60 gyroscopes is shown in Figure 6. The calculated zero-bias stability for the X, Y, and Z axis of the single gyroscope are $6.795^\circ/\text{h}$, $6.826^\circ/\text{h}$, and $6.418^\circ/\text{h}$ respectively. For the 60-gyroscope fusion system, the corresponding values are $0.891^\circ/\text{h}$, $0.893^\circ/\text{h}$, and $0.867^\circ/\text{h}$. The improvement factors for the X, Y, and Z axis array gyroscopes relative to a single gyroscope are 7.635, 7.644, and 7.403 respectively.

By plotting the angular velocity double-logarithmic curve using Allan variance, as shown in Figure 7, quantitative analysis yielded specific values for ARW, BI, RRW, and others, as presented in Table 2. The comparison of typical random errors from Allan variance in Table 2 demonstrates that, compared to a single gyroscope, an array comprising 60 gyroscopes exhibits a significant advantage in suppressing random errors. The X-axis array gyroscope achieved a 7.67 times improvement over a single gyroscope in angular random walk; the Y-axis array gyroscope achieved a 7.587 times improvement; and the Z-axis array gyroscope achieved a 7.609 times improvement. Regarding zero-bias instability, the X, Y, and Z-axis array gyroscopes achieved improvement factors of 7.50, 7.439, and 7.583, respectively, compared to a single gyroscope. For rate random walk, the X, Y, and Z-axis array gyroscopes achieved improvement factors of 7.579, 7.572, and 7.390 respectively compared to a single gyroscope. Overall, the 60-gyroscope array demonstrated

approximately 7.4–7.7 times improvements over individual gyroscopes for all three typical random error categories (ARW, BI, RRW) across the X, Y, and Z axis. This performance approaches the theoretical value of 7.74, conclusively validating the effectiveness of array design in mitigating gyroscope random errors.

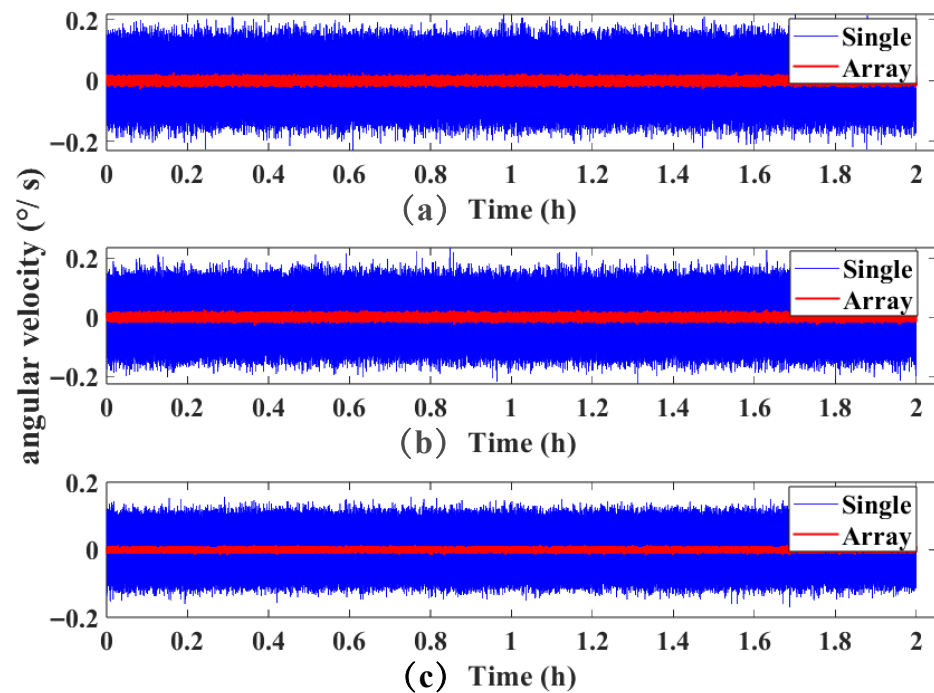


Figure 6. Comparison of angular velocity data for the X, Y, and Z axis: (a) angular velocity data for X-axis; (b) angular velocity data for Y-axis; (c) angular velocity data for Z-axis.

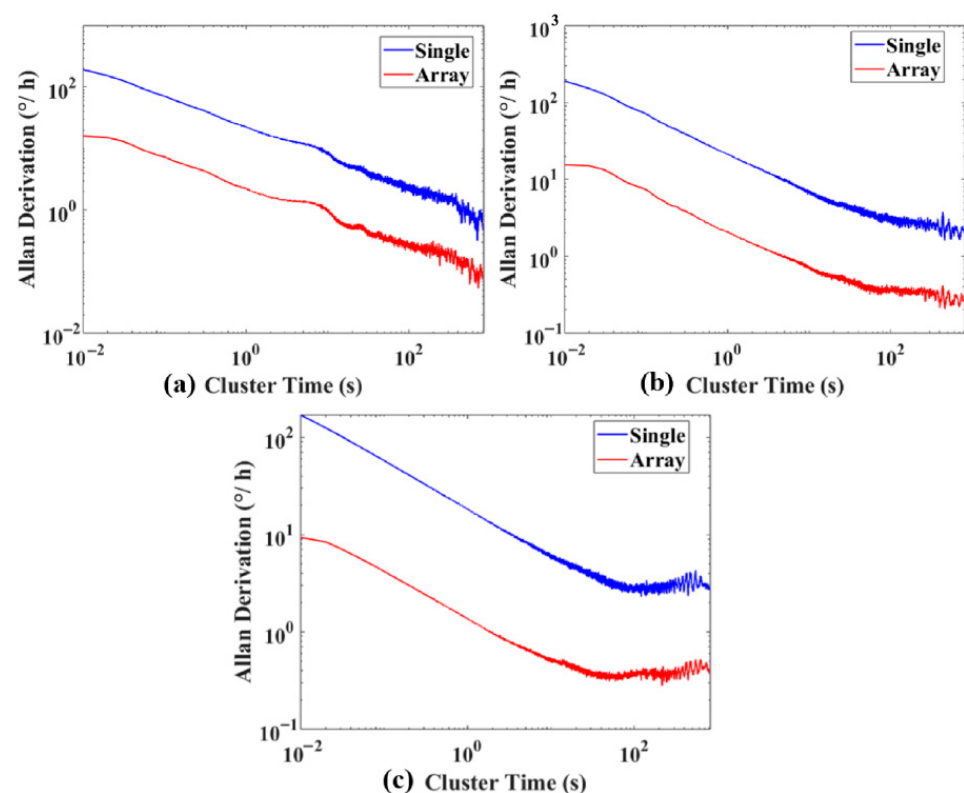


Figure 7. Allan variance plot comparisons: (a) X-axis; (b) Y-axis; (c) Z-axis.

Table 2. Allan variance comparisons of typical random errors.

Axis	Noise Type	Value	Improvement
X-axis	ARW ($^{\circ}/\sqrt{h}$)	0.345 (Single), 0.045 (Array)	7.67 times
	BI ($^{\circ}/h$)	2.214 (Single), 0.295 (Array)	7.50 times
	RRW ($^{\circ}/h/\sqrt{h}$)	44.128 (Single), 5.822 (Array)	7.579 times
Y-axis	ARW ($^{\circ}/\sqrt{h}$)	0.349 (Single), 0.046 (Array)	7.587 times
	BI ($^{\circ}/h$)	2.321 (Single), 0.312 (Array)	7.439 times
	RRW ($^{\circ}/h/\sqrt{h}$)	46.122 (Single), 6.091 (Array)	7.572 times
Z-axis	ARW ($^{\circ}/\sqrt{h}$)	0.312 (Single), 0.041 (Array)	7.609 times
	BI ($^{\circ}/h$)	1.873 (Single), 0.247 (Array)	7.583 times
	RRW ($^{\circ}/h/\sqrt{h}$)	35.642 (Single), 4.823 (Array)	7.390 times

4.2. Test Results for 100 m Pipeline

To further demonstrate the practical performance of the array MIMU, measurements are conducted on a standard pipe, as shown in Figure 8. Due to the frequent presence of magnetic interference within pipelines, which inevitably affects magnetometers, selecting high-precision magnetometers and calibrating them through multi-position ellipsoidal fitting significantly reduces this impact. Additionally, during pipeline testing, we solely rely on magnetometers to determine the carrier's initial heading angle. To further ensure data accuracy, multiple magnetic readings are collected during the magnetic data acquisition phase. The program automatically filters out outliers and averages the remaining magnetic data to guarantee the accuracy of the initial heading angle.

**Figure 8.** Standard pipeline actual test scenario.

Mount the array MIMU on a servo-hub trolley to conduct bidirectional measurements along the pipeline, defining its X-axis as the trolley's forward direction. Initially, determine the trolley's initial attitude angles using the accelerometer and magnetometer. Subsequently, record angular velocity data and wheel travel distance every 10 cm of forward movement. Upon experiment completion, stored data is retrieved via the host computer to obtain quaternion and positional information, as illustrated in Figures 9 and 10. Subsequent attitude calculation of the quaternion and positional data using MATLAB R2020a yields the pipeline's actual measurement trajectory. However, this trajectory represents relative position coordinates; subsequent integration with the pipeline's starting point coordinates is required to ultimately derive the pipeline's absolute position coordinates.

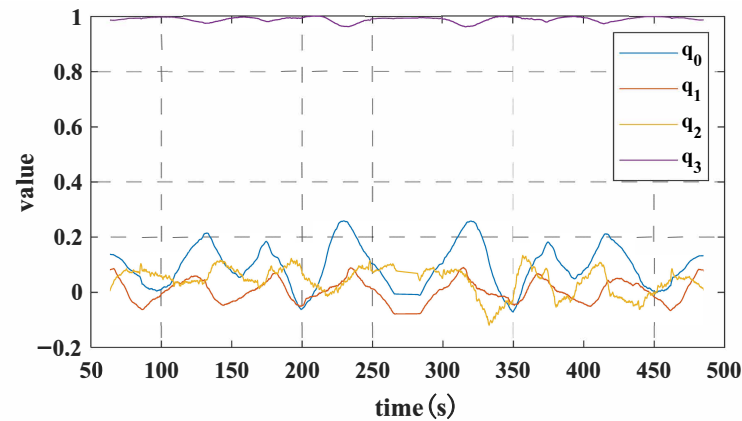


Figure 9. Round-trip measurement of quaternion data.

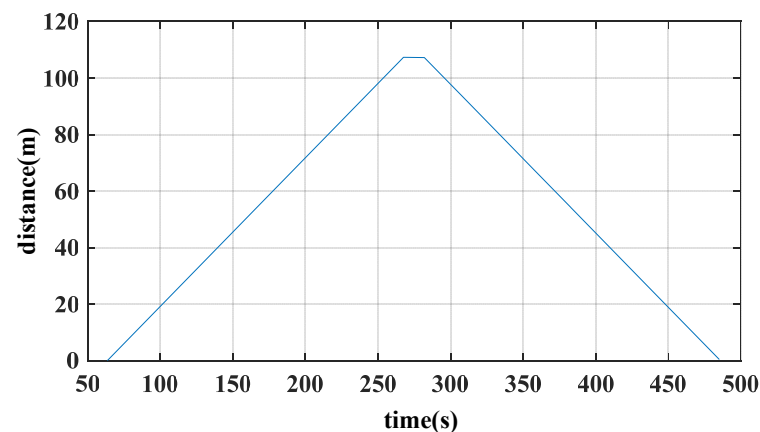


Figure 10. Round-trip measurement of positional data.

Repeating the experiment three times, the comparison between the measured absolute position coordinates and the true position coordinates is shown in Figures 11–13. The red curve represents the pipeline's true position coordinates, which were measured by the collaborating unit using millimeter-grade total stations and electronic levels in accordance with the Hebei Provincial Local Metrology Technical Specifications, while the other three coloured curves indicate the absolute position coordinates derived from processing the test data. Compared with the standard pipeline, the measured absolute position coordinates largely coincide with the true position coordinate curve. Calculations indicate a horizontal positioning error of less than 0.0774 m and an elevation positioning error of less than 0.0351 m.

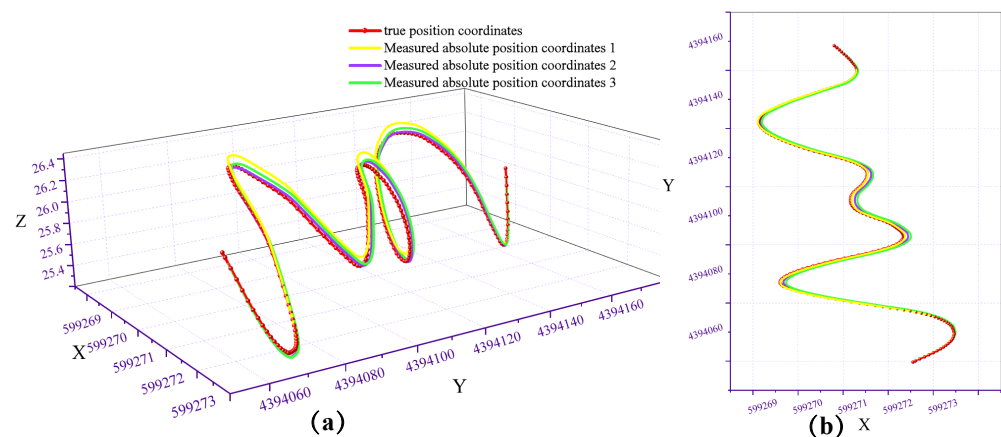


Figure 11. Comparison of the measured trajectory and the true trajectory: (a) 3D view; (b) top view.

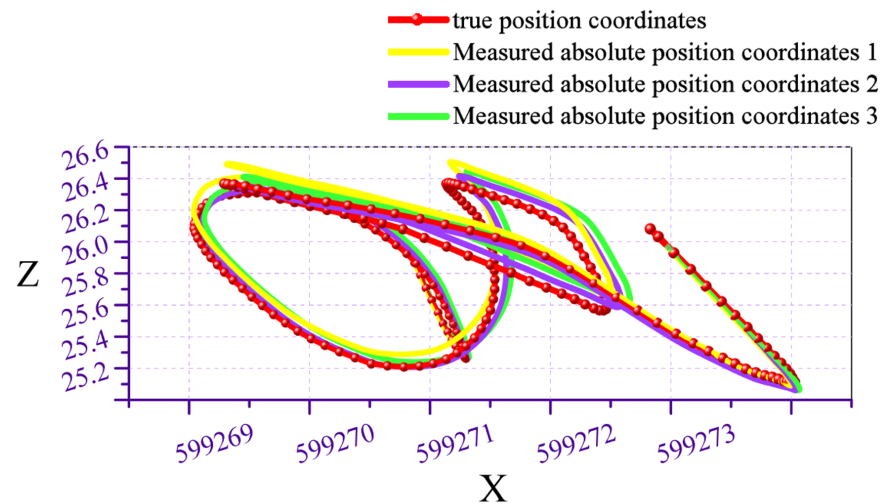


Figure 12. Comparison of measured trajectory with actual trajectory in front view.

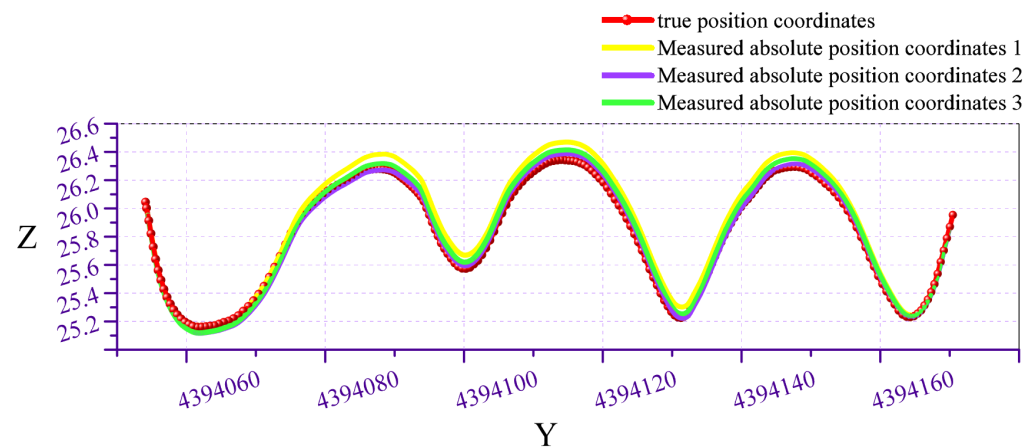


Figure 13. Side-view comparison of measured trajectory versus actual trajectory.

5. Conclusions

This paper presents a real-time data acquisition system for array MIMU based on FPGA+ARM. It enables MIMU data reading, processing, multi-source information fusion, data storage, and communication with host computers, establishing a complete data link encompassing acquisition, transmission, processing, and storage. The paper details the overall system design (hardware circuit design and software code development), the construction of attitude update algorithms, and measurement accuracy evaluation metrics. Static experiments validated the system's performance, demonstrating a 7.4 to 7.7 times improvement over a single gyroscope. In practical pipeline measurement applications, the system accurately calculated the position coordinates of a 100-m pipeline, achieving horizontal positioning errors below 0.0774 m and vertical positioning errors below 0.0351 m. This fully demonstrates the array design's significant effectiveness in mitigating gyroscope random errors, providing a reliable technical solution for pipeline surveying and enabling its application in real-time scenarios demanding high precision. Subsequent work will prioritize the integration of real-time noise reduction algorithms based on Kalman filtering while strictly maintaining the system's real-time performance. Specifically, this involves constructing a two-stage noise reduction architecture combining "array fusion + Kalman filtering"—using the hardware-level mean fusion from 60 MIMU channels as the observation input for Kalman filtering, and employing the filtering algorithm to dynamically estimate gyro random errors for real-time compensation. Next, optimize the adaptive filtering parameter adjustment strategy. Finally, embed this integrated algorithm into an FPGA to

achieve a secondary enhancement in gyroscope measurement accuracy, further suppressing inherent gyroscope random errors. Simultaneously, expand its application scenarios to high-precision domains such as drones, robotics, and underground space exploration.

Author Contributions: Conceptualization, X.Q., H.W. and J.Z.; methodology, Z.D.; data curation, X.Q., Y.W. and J.Z.; writing—original draft preparation, X.Q., H.W. and Z.D.; writing—review and editing, H.C.; supervision, H.C.; project administration, T.G. and H.C.; funding acquisition, Y.W., J.Z. and H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Science and Technology Innovation Key R&D Program of Chongqing under Grant No. CSTB2025TIAD-STX0016, the Higher Education Institution Science and Technology Research Project of Hebei Province (No. CXY2024048), the National Key Research and Development Program of China under Grant (No. 2022YFB3205000), and the National Natural Science Foundation of China under Grant (No. 52475586 and U2230206).

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: Author Junqing Zhang was employed by the company Tangshan Zhongyu Technology Development Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Wei, W.; Tian, H.; Cai, Q.; Cui, R.; Li, H.; Cao, H. Design and fabrication of a quad-mass high-frequency triaxial MEMS gyroscope. *Opt. Precis. Eng.* **2025**, *33*, 1916–1928. [\[CrossRef\]](#)
2. Liu, L.; Li, Z.; Lu, R.; Zhou, Z.; Chen, H.; Jiang, W. An Enhanced Smartphone GNSS/MEMS-IMU Integration Seamless Positioning Method in Urban Environments. *IEEE Sens. J.* **2024**, *24*, 41251–41263. [\[CrossRef\]](#)
3. Liu, R.; Lyu, P.; Lai, J.; Lu, J.; Hu, H. Mechanism and implementation of error suppression in MEMS inertial navigation system via rotational modulation and array. *Meas. Sci. Technol.* **2025**, *36*, 095113. [\[CrossRef\]](#)
4. Patrizi, G.; Carratù, M.; Ciani, L.; Sommella, P.; Catelani, M.; Pietrosanto, A. Temperature Sensitivity Analysis of Inertial Measurement Unit Under Dynamic Conditions. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 9510209. [\[CrossRef\]](#)
5. Jiang, B.; Jiao, J.; Li, Y.; Wang, Z.; Su, Y. An eight-mass MEMS gyro with area-variable comb capacitance. In Proceedings of the 2024 IEEE 19th International Conference on Nano/Micro Engineered and Molecular Systems (NEMS), Kyoto, Japan, 2–5 May 2024; pp. 1–4. [\[CrossRef\]](#)
6. Yang, Y.; Song, Y.; Xin, Y.; Wang, G.; Yi, H.; Zhang, Z.; Rao, J. Low stress packaging of the small three-axis gyro. *J. Phys. Conf. Ser.* **2024**, *2740*, 012020. [\[CrossRef\]](#)
7. Giner, J.; Maeda, D.; Ono, K.; Shkel, A.M.; Sekiguchi, T. MEMS Gyroscope With Concentrated Springs Suspensions Demonstrating Single Digit Frequency Split and Temperature Robustness. *J. Microelectromech. Syst.* **2019**, *28*, 25–35. [\[CrossRef\]](#)
8. Tsukamoto, T.; Tanaka, S. Rate Integrating Gyroscope Using Independently Controlled CW and CCW Modes on Single Resonator. *J. Microelectromech. Syst.* **2021**, *30*, 15–23. [\[CrossRef\]](#)
9. Lan, J.; Wang, K.; Song, S.; Li, K.; Liu, C.; He, X.; Hou, Y.; Tang, S. Method for measuring non-stationary motion attitude based on MEMS-IMU array data fusion and adaptive filtering. *Meas. Sci. Technol.* **2024**, *35*, 086304. [\[CrossRef\]](#)
10. Xue, L.; Wang, X.; Yang, B.; Yuan, W.; Yuan, G. Analysis of Correlation in MEMS Gyroscope Array and its Influence on Accuracy Improvement for the Combined Angular Rate Signal. *Micromachines* **2018**, *9*, 22. [\[CrossRef\]](#) [\[PubMed\]](#)
11. Bayard, D.S.; Ploen, S.R. High accuracy inertial sensors from inexpensive components. US patent 0187623, 2 October 2003
12. Kamata, H.; Kimishima, M.; Sawada, T.; Suga, Y.; Takeda, H.; Yamashita, K.; Mitani, S. MEMS Gyro Array Employing Array Signal Processing for Interference and Outlier Suppression. In Proceedings of the 2020 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL), Hiroshima, Japan, 23–26 March 2020; [\[CrossRef\]](#)
13. Wang, J.; Olson, E. High-performance inertial measurements using a redundant array of inexpensive gyroscopes (RAIG). In Proceedings of the 2015 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), San Diego, CA, USA, 14–16 September 2015; pp. 71–76. [\[CrossRef\]](#)
14. Yi, H.; Chen, D.; Jiang, H.; Bai, H.; Li, G.; Wen, G.; Huang, Y. Research and implementation of virtual high precision gyroscope based on differential array. *Navig. Position. Timing* **2024**, *11*, 138–150. [\[CrossRef\]](#)
15. Dai, Q.; Xiao, G.R.; Zhou, G.H.; Ye, Q.Q.; Han, S.Y. A novel Gaussian sum quaternion constrained cubature Kalman filter algorithm for GNSS/SINS integrated attitude determination and positioning system. *Front. Neurobot.* **2024**, *18*, 1374531. [\[CrossRef\]](#)

16. Wu, C.G.; Jing, Z.L.; Hao, C.J.; Yan, F.Z. Gyro Denoising and Fusion Quaternion Attitude Estimation Based on Improved Least Squares. In Proceedings of the 2019 Chinese Automation Congress (CAC), Hangzhou, China, 22–24 November 2019; pp. 1496–1500. [[CrossRef](#)]
17. Du, Q.; Ding, B.; Wei, Y. Experimental analysis of quaternion-based attitude estimation algorithms. In Proceedings of the 2024 36th Chinese Control and Decision Conference (CCDC), Xi'an, China, 25–27 May 2024; pp. 4095–4099. [[CrossRef](#)]
18. Rudyk, A.V.; Semenov, A.O.; Kryvinska, N.; Semenova, O.O.; Kvasnikov, V.P.; Safonyk, A.P. Strapdown Inertial Navigation Systems for Positioning Mobile Robots—MEMS Gyroscopes Random Errors Analysis Using Allan Variance Method. *Sensors* **2020**, *20*, 4841. [[CrossRef](#)] [[PubMed](#)]
19. Li, K.; Wang, K.; Song, S.; Liu, X.; He, X.; Hou, Y.; Tang, S. Improved strong tracking Sage–Husa adaptive algorithm for multi-MEMS IMU data fusion. *Rev. Sci. Instrum.* **2025**, *96*, 055002. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.