Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 15, No. 3, 2026

Design of a Multi-Sensor Fusion Planar Pose Sensing System Based on STM32

Qian Zhang *, Qiang Jiang, Hongwei Du, Hongru Liu

Sichuan University Jinjiang College, Meishan, Sichuan, 620000, China

*** Corresponding author:** Qian Zhang

Abstract: Addressing issues such as complex system architecture, high hardware redundancy, and poor engineering adaptability in existing mobile robot planar pose sensing solutions, this paper designs and implements a multi-sensor fusion planar pose sensing system based on STM32.Centered around the STM32F103C8T6 microcontroller as the core processing unit, the system employs orthogonally arranged incremental encoders and a high-precision inertial measurement unit (IMU) as its sensing core. It encompasses the full design and implementation process from three-dimensional structure, hardware circuitry, and PCB layout to embedded software and pose estimation algorithms. By combining encoder displacement increments with IMU heading information, the system estimates a mobile platform's position and orientation in real time within a two-dimensional plane. By optimizing hardware resource allocation and designing a lightweight software architecture, it significantly reduces system complexity and development barriers. Featuring simple interfaces, high integration, and strong versatility, this system can widely accommodate the pose-perception needs of small-to-medium-sized planar mobile robots, demonstrating excellent engineering value.

Keywords: STM32; Multi-sensor Fusion; Planar Pose Sensing; Inertial Measurement Unit; Encoder; Pose Estimation.

1. Introduction

With the large-scale application of mobile robotics in warehousing logistics, industrial inspection, and intelligent services, real-time pose perception within a plane has become the core foundation for robots to achieve autonomous navigation and motion control. Among current mainstream planar positioning solutions, technologies such as laser SLAM and visual SLAM offer high positioning accuracy but suffer from complex system architectures, high hardware costs, challenging algorithm development, and strong dependence on environmental features, making them difficult to widely adopt on compact, lightweight robotic platforms[1]. Traditional dead-reckoning approaches that rely solely on wheel encoders for displacement integration suffer from significant heading accumulation errors and rapid degradation in positioning accuracy over long distances. Pure inertial navigation solutions, constrained by the zero-bias drift characteristics of MEMS sensors, cannot sustain stable pose output over extended periods[2].

The pose sensing solution based on encoder-IMU fusion offers advantages such as environmental independence, real-time performance, a simple structure, and low development barriers, making it the mainstream technical approach for pose perception in small-to-medium mobile robots[3]. However, existing solutions often employ high-end microcontrollers and redundant hardware architectures, leading to issues such as wasted hardware resources, high system complexity, difficult secondary development, and poor adaptability across different scenarios, making it challenging to meet lightweight application needs across various environments. To address these challenges, this paper proposes a multi-sensor fusion planar pose-sensing system for planar mobile robots. Based on the STM32F103C8T6 microcontroller, it achieves streamlined hardware design and reconstruction. Paired with a high-precision IMU to ensure heading observation accuracy, while rewriting the embedded software architecture and implementing lightweight pose estimation algorithms. This ultimately results in a full-featured pose-sensing unit that integrates motion data acquisition, real-time pose estimation, and interaction with external data. It significantly reduces system complexity while maintaining performance, enhancing the system's versatility and engineering adaptability.

2. Overall System Architecture Design

The system meets the needs of robots for full-field planar pose measurement. Using a modular, layered design, it has three layers: hardware, software, and application. Each layer has standard interfaces and low coupling, making it easy to expand and adapt. Figure 1 shows the three layers.

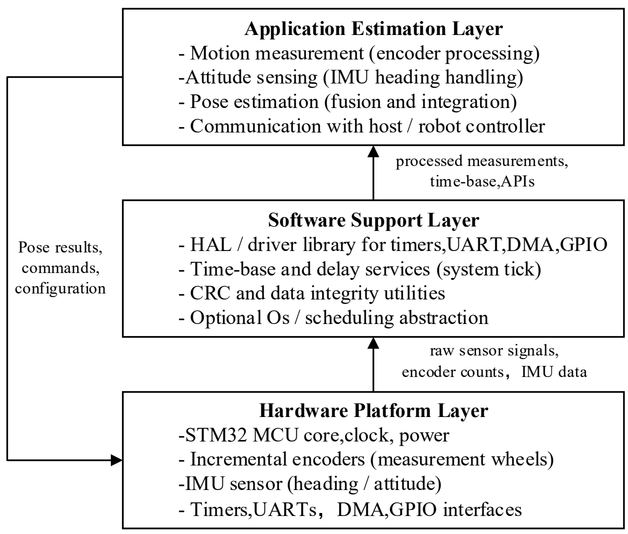

Figure 1. System Architecture Block Diagram

The Hardware Platform Layer centers on the STM32F103C8T6 microcontroller. It interfaces with two quadrature incremental encoders via timers to capture body displacement data and connects to a high-precision IMU via serial port for heading observations. This layer also integrates power management, status indicators, and serial communication interfaces to ensure system power supply, status monitoring, and bidirectional data exchange. The software support layer encapsulates and abstracts low-level hardware drivers, establishes millisecond-level time management to ensure stable sampling cycles, and incorporates CRC verification algorithms to enhance data transmission reliability. The application computation layer follows a standardized workflow for data acquisition, preprocessing, computation, and interaction to process sensor data, perform recursive pose computation, and facilitate external data exchange, serving as the system's core functional component.

The system employs a "fixed-cycle main loop + asynchronous interrupt" scheduling mechanism. The main loop operates at a 1ms sampling cycle, sequentially executing core tasks including sensor data acquisition, attitude and posture calculation, and data communication within each cycle. Concurrently, DMA idle interrupts and serial port receive interrupts asynchronously handle IMU heading data updates and external control command parsing. This approach prevents the main loop from blocking, ensuring system real-time performance and operational stability.

3. Hardware System Design

The hardware is designed for high integration, reliability, and efficient resource use. The design covers circuit planning, PCB layout, and 3D models. The hardware has four core parts: control unit, sensors, power, and communication. Figure 2 shows the full layout and component placement.

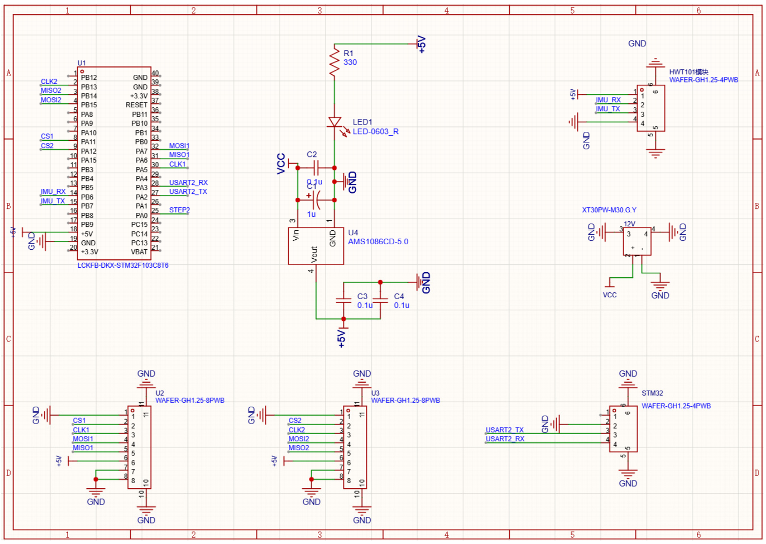

Figure 2. System Hardware Schematic Diagram

(1) Main Control Unit Design

The main control unit employs the STM32F103C8T6 microcontroller. Based on the ARM Cortex-M3 core, this chip operates at a maximum clock speed of 72MHz.with 64KB of Flash and 20KB of SRAM, fully meeting the computational demands of pose estimation algorithms and peripheral control requirements. The chip integrates 3 general-purpose serial ports, 2 SPI interfaces, 4 16-bit timers, and a rich array of GPIO peripherals. This enables seamless adaptation to encoders, IMUs, and other peripheral devices without requiring additional interface expansion chips, significantly simplifying hardware circuit design. This chip offers a robust development ecosystem, a stable supply chain, and an outstanding cost-performance ratio, enabling optimal hardware resource allocation while avoiding the resource wastage and cost overruns associated with high-end controllers[4].

The minimal system circuit for the controller includes power filtering, a crystal oscillator, a reset circuit, and a program download circuit. It also reserves an SWD debug interface for convenient online debugging and programming. By allocating GPIO ports functionally, timer channels are assigned for encoder pulse acquisition, while serial port channels are dedicated to IMU data reading and host communication, achieving a rational configuration of peripheral resources.

(2) Sensor Unit Design

The sensor unit serves as the perception core of the system, comprising two components: the orthogonal measuring wheel encoder module and the high-precision IMU module. It provides two types of core measurements—displacement and heading—for pose estimation. The fusion of these measurements forms the core architecture of the system's multi-sensor fusion. Orthogonal Measurement Wheel Encoder Module: This module employs two coaxially mounted incremental encoders, with orthogonally arranged measurement wheels. Each wheel corresponds to the X and Y axes of the platform's intrinsic coordinate system, enabling direct measurement of translational displacement within this coordinate frame. This design eliminates heading error coupling inherent in traditional differential wheel configurations. The encoders connect to the main controller via the SPI interface, supporting a maximum sampling frequency of 1 MHz. This enables precise capture of pulse increments within each sampling cycle. By combining encoder resolution with measurement wheel radius, pulse counts are converted into linear displacement values for the corresponding direction, providing high-precision displacement observations for attitude and position estimation.

The high-precision IMU module employs the HWT101 high-accuracy heading module. This module integrates MEMS inertial sensors with hard- and soft-iron calibration algorithms, achieving heading angle measurement accuracy of ±0.1°. It also exhibits excellent vibration resistance, making it suitable for complex motion scenarios in mobile robots. The IMU connects to the main controller via a serial port and continuously outputs heading angle observation data. This provides precise heading constraints for coordinate transformations, effectively suppressing the cumulative heading error inherent in pure encoder-based positioning and enhancing the long-term stability of pose estimation[5].

(3) Power Management and Communication Interface Design

The power management unit employs the AMS1086CD-5.0 linear regulator chip, which stably converts a wide input voltage range of 6-12V to the system's required 5V operating voltage. Simultaneously, a secondary voltage regulator circuit outputs 3.3V to power the main controller and sensors. The input stage incorporates reverse-polarity protection, while the output stage features multiple filter capacitors to effectively suppress power-supply ripple, ensuring stable system power delivery. Power and operational status indicator lights are also included for convenient field debugging and real-time monitoring[6].

The communication interface unit incorporates two independent asynchronous serial communication ports: one for real-time reading of IMU heading data, and another for bidirectional communication with the host computer or robot controller. This enables periodic reporting of pose data and transmission of reset commands and installation mode configuration instructions. The interface circuitry includes standard connectors, allowing direct compatibility with the robot controller's electrical interface without requiring additional adapter circuits, thereby enhancing system versatility.

(4) PCB and 3D Structural Design

The PCB employs a double-layer design with a compact overall footprint. By optimizing routing rules, digital and analog circuits are zoned separately to minimize high-frequency signal trace lengths, effectively reducing signal interference. Standardized mounting holes and connector pads are reserved, balancing structural installation convenience with hardware expansion flexibility. The complete routing and layout design is shown in Figure 3.

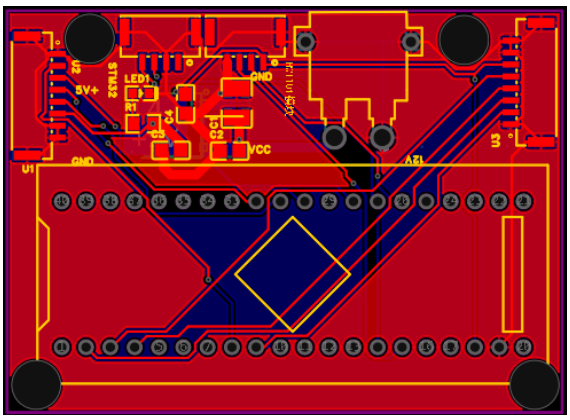

Figure 3. System PCB Layout

The system's 3D structure employs a modular integrated design, consolidating the main control board, measurement wheel, and IMU module within a single structural body. This achieves a compact, integrated system design that can be directly installed as an independent sensing unit on various mobile robot platforms without requiring additional structural adaptations. Simultaneously, the structural design and software parameters support multiple mounting configurations, accommodating diverse mechanical orientations and platform dimensions. Adaptation requires no hardware or core code modifications—only parameter configuration—significantly lowering the system's application threshold. The complete structural design is illustrated in Figure 4.

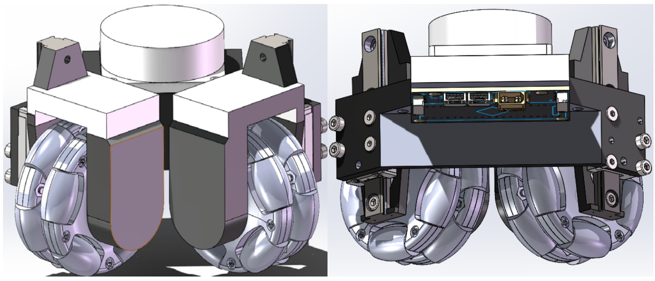

Figure 4. Three-dimensional structural model of the system

4. Software System Design

The software system is developed using the STM32 standard library and adopts a modular, time-sequenced design approach. While ensuring real-time pose estimation, it achieves lightweight code and high maintainability. The overall software workflow comprises three core components: system initialization, main loop task scheduling, and asynchronous interrupt handling. The system's main program flowchart is shown in Figure 5.

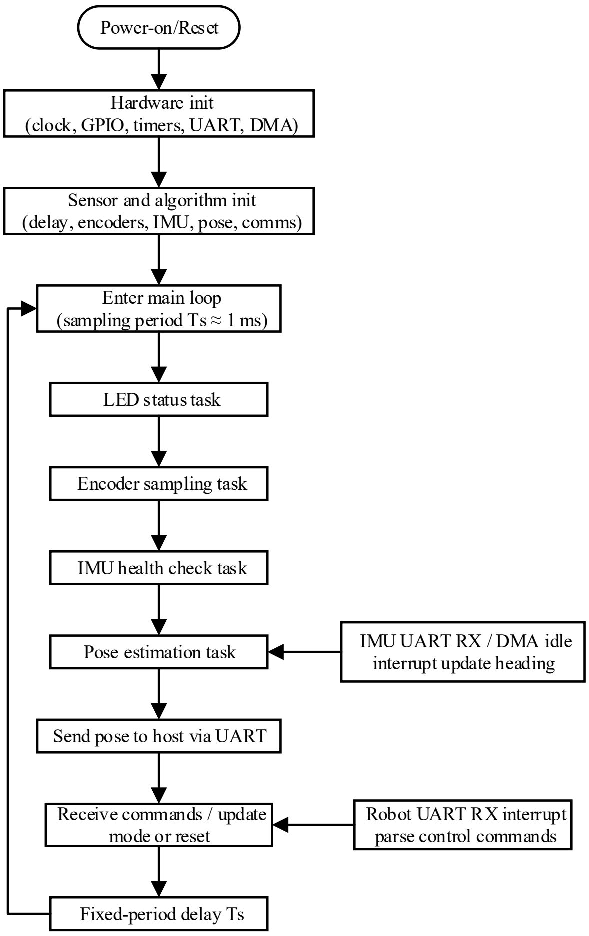

Figure 5. System Main Program Flowchart

(1) System Initialization Process

Upon power-up or reset, the system first performs initialization operations, divided into two core phases:

hardware peripheral initialization and sensor/algorithm initialization. During the hardware peripheral initialization phase, the system clock configuration is completed first, setting the main controller's frequency to 72 MHz. Subsequently, GPIO, timers, serial ports, DMA, and interrupt controllers are initialized in sequence. The encoder sampling timer's counting mode, the serial port's baud rate, and the interrupt triggering method are configured to provide the hardware foundation for subsequent sensor data acquisition and communication.

During the sensor and algorithm initialization phase, the power-on self-test and parameter configuration for the encoder and IMU are completed. After confirming normal sensor communication, the IMU is zero-calibrated, setting the current platform orientation as the heading reference zero position. Simultaneously, the state variables for pose estimation are initialized, with the global position coordinates and heading angle set to zero. Kinematic model parameters are configured, preparing for subsequent pose recursion.

(2) Main Loop Task Scheduling

After initialization, the system enters a main loop with a fixed sampling period set to 1ms. Core tasks are executed in priority order within each cycle to ensure orderly system functionality. The LED status indication task controls the illumination and flashing frequency of the indicator light based on the system's operational status, visually indicating conditions such as normal operation, sensor faults, or communication anomalies to facilitate field debugging and troubleshooting. The encoder sampling task reads the pulse increments from both encoders during the current sampling cycle. Combining parameters such as encoder resolution and wheel radius, it converts them into linear displacements in two orthogonal directions within the body coordinate system, constructing the body displacement increment vector.

The IMU health monitoring task tracks the IMU's communication status and data validity. If communication interruptions or data anomalies occur, the algorithm immediately triggers fault alarms and activates a heading angle retention mechanism to prevent attitude and position calculation errors caused by invalid data. The Pose Estimation Task executes pose estimation algorithms based on preprocessed encoder displacement increments and IMU heading angle data. It performs coordinate transformations and recursive updates of the global pose to obtain the platform's current global position coordinates and heading angle.

The pose data reporting task packages the latest computed pose data into structured data frames, appends header and CRC-check fields, and periodically transmits them via the serial port to the host computer or robot controller, ensuring real-time pose data output. The Control Command Parsing Task reads external control commands from the serial port receive buffer, parses and validates them, and executes operations such as soft reset of pose, installation mode configuration, and sensor parameter adjustment based on command content. This enables online configuration of the system's operational state.

Upon completing all tasks, the system employs precise timer delays to compensate for the remaining time in the current sampling cycle. This ensures strict stability of the main loop's sampling period, providing a constant discrete time step for pose estimation algorithms. This approach prevents estimation errors caused by periodic fluctuations.

(3) Asynchronous Interrupt Handling Mechanism

To prevent the main loop from blocking and enhance system real-time performance, a two-level asynchronous interrupt handling mechanism is designed to ensure real-time response for core data. The IMU serial port DMA idle interrupt is triggered after the IMU serial port completes receiving a full frame of heading data. Within the interrupt service routine, the IMU data frame is immediately parsed and the heading angle extracted. The system's heading angle state variable is updated to ensure real-time heading data, preventing angular errors caused by sampling delays in the main loop.

The master serial port receive interrupt is triggered when control commands are received from the host computer. Within the interrupt service routine, the command is received and cached, and the command reception flag is set. This flag is then used by the main loop for subsequent parsing and execution, ensuring rapid response to control commands.

5. Design of Pose Estimation Algorithm

The pose estimation algorithm constitutes the system's core. By fusing encoder displacement increments with heading angle observations from the IMU, it performs real-time recursive calculation of the platform's global pose. The algorithm employs a lightweight design for stable operation on low-computational-power controllers while maintaining estimation accuracy and real-time performance. The complete algorithmic principles and execution flow are illustrated in the Pose Estimation Principle Block Diagram (Figures 6 and 7).

(1) Definition of Coordinate Systems and State Variables

To achieve a standardized description of planar motion, two core coordinate systems are first defined: the global coordinate system and the body coordinate system. The global coordinate system is a fixed coordinate system referenced to the worksite or map. Its origin and axes can be flexibly configured based on the application scenario. The body coordinate system is a moving coordinate system rigidly attached to the platform body. Its origin is set at the system installation reference point, and its two axes align with the measurement direction of the orthogonal measurement wheel, moving synchronously with the platform rigid body.

The platform's pose state in the global coordinate system can be expressed as:

$$P_k = \begin{bmatrix} X_k \\ Y_k \\ \psi_k \end{bmatrix}$$

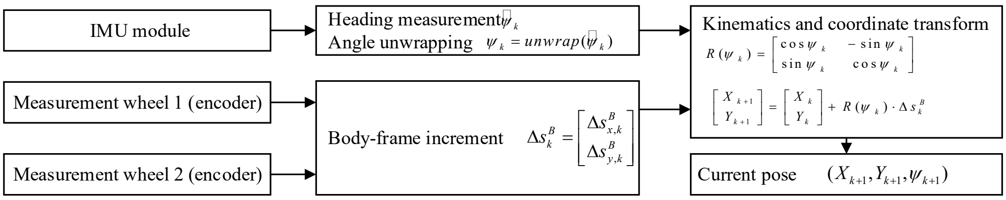

Figure 6. Principle Diagram of Pose Estimation

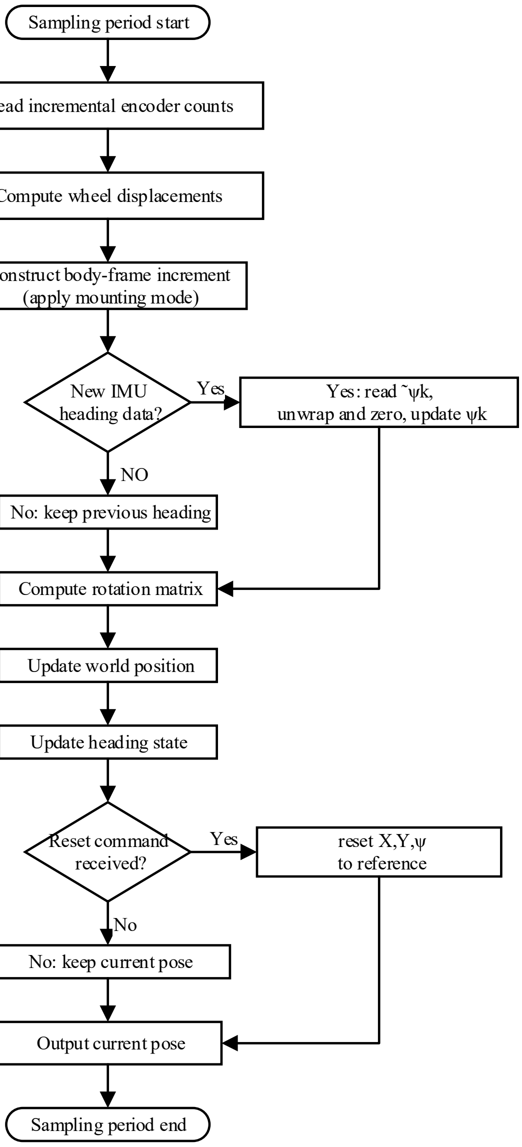

Figure 7. Algorithm Execution Flow Diagram

Where X_k and Y_k denote the platform's two-dimensional position coordinates in the global coordinate system, ψ_k represents the platform's heading angle relative to the global coordinate system, and k denotes the discrete sampling time.

Within a single discrete sampling period, the displacement increment of the platform in its inertial coordinate system can be expressed as:

$$\Delta s_k^B = \begin{bmatrix} \Delta s_{x,k}^B \\ \Delta s_{y,k}^B \end{bmatrix}$$

Where $\Delta s_{x,k}^B$ and $\Delta s_{y,k}^B$ represent the displacement increments along the X and Y axes of the platform's inertial coordinate system, respectively.

(2) Measurement Preprocessing

During the k sampling cycle, the pulse increments from both encoders are first read. Combining the encoder line count and the measuring wheel radius, the pulse count is converted into the corresponding linear displacement of the measuring wheel using the conversion formula:

$$\Delta l = \frac{2\pi r \cdot \Delta n}{N}$$

Where Δn represents the pulse increment of the encoder within a single cycle.

After completing the linear displacement conversion, based on the installation orientation and geometric arrangement of the measuring wheels, the linear displacements from both wheels are converted into two-dimensional displacement increments within the body coordinate system. The software incorporates configurable installation mode parameters. By adjusting the sign of displacement components and their axial mapping relationships, it adapts to various mechanical configurations, such as rotational installation or left-right swapping of measuring wheels. This eliminates the need for hardware modifications, enhancing the system's adaptability.

Raw heading angle observations from IMUs are typically constrained within the range $[-\pi, \pi]$. When a platform rotates continuously beyond this range, angular jumps occur at π or $-\pi$, rendering the data unusable for coordinate transformation. To address this, the system performs angle disentanglement on raw heading angle observations. By comparing heading angle observations from adjacent time points, when the absolute value of the observation difference exceeds , the current observation is automatically adjusted to eliminate angle jumps. This yields a physically continuous cumulative heading angle. The system also supports power-on or command-triggered zero-point calibration, setting the current orientation as the reference zero position to ensure the physical meaning of the heading angle remains unambiguous[7].

(3) Kinematic Modeling and Pose Prediction

Under the assumption of planar rigid-body motion, the platform's movement over short durations can be approximated as planar small-displacement motion. Using a two-dimensional rotation matrix, displacement increments in the local coordinate system can be mapped to the global coordinate system, enabling recursive pose updates.

The two-dimensional rotation matrix from the local coordinate system to the global coordinate system is:

$$R(\psi_k) = \begin{bmatrix} \cos\psi_k & -\sin\psi_k \\ \sin\psi_k & \cos\psi_k \end{bmatrix}$$

Based on this rotation matrix, the recursive formula for global position coordinates is:

$$\begin{bmatrix} X_{k+1} \\ Y_{k+1} \end{bmatrix} = \begin{bmatrix} X_k \\ Y_k \end{bmatrix} + R(\psi_k) \cdot \Delta s_k^B$$

The recursive formula for heading angle is:

$$\psi_{k+1} = \psi_k + \Delta\psi_k$$

where $\Delta\psi_k$ represents the heading angle increment within the current sampling period, derived from the differential of the disentangled IMU heading observation.

(4) Engineering Implementation Flow of the Algorithm

The pose estimation algorithm executes in a fixed sequence within each sampling cycle, fully synchronized with the software main loop to ensure stable operation. Upon entering the sampling cycle, the timestamp updates from k to $k+1$. The incremental pulse counts from both encoders are first read to compute the linear displacement of the corresponding measurement wheel, constructing the displacement increment in the body coordinate system. Then, it checks whether new IMU heading data has been received. If new data is available, it reads the raw heading angle, performs angle disentanglement and zero calibration, and updates the current heading angle. If no new data is available, it uses the heading angle from the previous time step.

After completing the heading angle update, compute the two-dimensional rotation matrix based on the current heading angle. Update the position coordinates in the global coordinate system using the pose recursion formula, then update the heading angle state. Simultaneously, check whether a pose reset command has been received. If a command is received, reset the global position and heading angle to reference values. If no command is received, maintain the currently computed pose state. Finally, the platform's current pose is output for subsequent data reporting and control purposes.

6. System Design Optimization and Engineering Adaptability Analysis

The multi-sensor fusion planar pose sensing system designed in this paper achieves comprehensive optimization across hardware, software, and structural dimensions based on existing technical solutions. It effectively addresses the issues of high system complexity, excessive redundancy, and poor adaptability in current approaches while ensuring core performance and functional integrity.

(1) Hardware System Lightweighting and Streamlined Optimization

Existing comparable solutions predominantly employ high-end main control chips with multi-chip expansion architectures, resulting in wasted hardware resources, complex circuit design, and high failure rates. The hardware system described in this paper achieves optimal resource allocation using the STM32F103C8T6 main controller. By leveraging the chip's built-in peripheral resources, it enables direct integration of encoders, IMU, and communication interfaces without requiring additional expansion chips. This reduces the number of core circuit components by over 40%, significantly lowering hardware design complexity and failure rates. Simultaneously, selecting a high-precision IMU ensures heading observation accuracy without increasing system complexity, balancing system performance and hardware design.

(2) Modularization and Portability Optimization of Software Architecture

This paper reconfigures the system's embedded software architecture using a layered, modular design approach. Functions such as hardware drivers, system support, algorithmic computation, and communication interaction are decoupled and encapsulated. Standardized interfaces between modules facilitate future functional expansion and platform portability. Concurrently, the pose estimation algorithm underwent lightweight optimization. While maintaining computational accuracy, this simplifies the processing flow and reduces computational demands, enabling stable operation on a low-frequency Cortex-M3 core. Within a 1ms sampling cycle, the algorithm's computational time does not exceed 100μs, demonstrating excellent real-time performance. The scheduling mechanism, combining a fixed-cycle main loop with asynchronous interrupts, ensures both the timing stability of core tasks and real-time response for critical data, thereby enhancing system operational stability.

(3) Universal Optimization of Integrated Structure

This paper completes the integrated 3D structural design of the system, integrating the main control unit, sensor unit, and measuring wheel mechanism within a single structural body. This achieves a compact, integrated system design that can be directly installed as an independent sensing unit on various mobile robot platforms without requiring additional structural adaptations. Simultaneously, the structural design and software parameters support multiple mounting configurations, accommodating diverse mechanical orientations and platform dimensions. Adaptation is achieved solely through parameter configuration without hardware or core code modifications, significantly lowering the system's application threshold.

(4) Engineering Application Value Analysis

The multi-sensor fusion planar pose sensing system designed in this paper achieves a full-process autonomous design for mobile robot planar pose perception. It offers core advantages, including a simple interface, high integration, strong versatility, and a low development threshold. For upper-layer application systems, this system functions as a black-boxed pose sensing unit. Host computers need not concern themselves with low-level details such as encoder counting, IMU protocol parsing, or pose estimation algorithms. Real-time platform pose data can be obtained via standard serial interfaces, while system configuration and reset can be accomplished through standardized commands. This enables seamless integration with various mobile robot control systems and motion control algorithms.

This system can be widely applied to meet planar pose-sensing requirements across scenarios such as small-to-medium mobile robots, warehouse AGVs, industrial inspection equipment, and educational robots. Compared to existing solutions, it significantly reduces system complexity and application barriers, demonstrating strong engineering application value and promising prospects for widespread adoption.

7. Conclusion

To address the issues of high system complexity, excessive redundancy, and poor engineering adaptability in existing planar pose-sensing solutions, this paper designs and implements a multi-sensor fusion planar pose-sensing system based on the STM32. The entire process—from hardware circuitry and PCB layout to 3D structure—was independently designed. Leveraging the STM32F103C8T6 microcontroller, the hardware system achieved streamlined optimization and optimal resource allocation. The embedded software architecture was restructured using a fixed-cycle main loop combined with asynchronous interrupt scheduling to ensure real-time performance and stability. A lightweight design and engineering implementation of the pose estimation algorithm were completed. Through multi-sensor fusion of encoders and IMUs, real-time estimation of the platform's pose within a two-dimensional plane was achieved.

The system designed herein significantly reduces complexity and development barriers while preserving core functionality and performance. It offers excellent versatility and engineering adaptability, providing stable, reliable pose perception for diverse mobile robot platforms with strong practical application value. Future work may further optimize the pose estimation algorithm and incorporate filtering techniques to suppress sensor noise, thereby enhancing the system's pose estimation accuracy and long-term stability.

Acknowledgments

The authors would like to gratefully acknowledge the financial support provided by Sichuan University Jinjiang College for the University-level Youth Scientific Research Fund Project (Grant No. QNJJ-2025-B12).

References

[1] Saad Mokssit, Daniel Bonilla Licea, Bassma Guermah, et al. Deep Learning Techniques for Visual SLAM: A Survey[J]. IEEE Access, 2023, 11: 20026-20050.

[2] R. Cechowicz, Marcin Bogucki, et al. Indoor vehicle tracking with a smart MEMS sensor[J]. MATEC Web of Conferences, 2019, 252: 02004.

[3] Amjady N. Short-term hourly load forecasting using time series modeling with peak load estimation capability [J]. IEEE Transactions on Power Systems, 2001, 16(4): 798-805.

[4] Yilan Dong, et al. Design and Implementation of Embedded Web Server Based on ARM[J]. IOP Conference Series Materials Science and Engineering, 2019, 612(5): 052067.

[5] Li Wei, Jinling Wang, et al. Effective Adaptive Kalman Filter for MEMS-IMU/Magnetometers Integrated Attitude and Heading Reference Systems[J]. Journal of Navigation, 2012, 66(1): 99-113.

[6] Chunlei Shi, Benjamin Walker, E. Zeisel, et al. A Highly Integrated Power Management IC for Advanced Mobile Applications[J]. IEEE Journal of Solid-State Circuits, 2007, 42(8): 1723-1731.

[7] Dongsheng Wang, Yongjie Lu, Lei Zhang, et al. Intelligent Positioning for a Commercial Mobile Platform in Seamless Indoor/Outdoor Scenes based on Multi-sensor Fusion[J]. Sensors, 2019, 19(7): 1696.