


Article

Study of the Navigation Method for a Snake Robot Based on the Kinematics Model with MEMS IMU

Xu Zhao ^{1,2} , Lihua Dou ^{1,*}, Zhong Su ^{1,2} and Ning Liu ²

¹ School of Automation, Beijing Institute of Technology, Beijing 100081, China; zhaoxu00@126.com (X.Z.); sz@bistu.edu.cn (Z.S.)

² Beijing Key Laboratory of High Dynamic Navigation Technology, Beijing Information Science & Technological University, Beijing 100101, China; ning.liu@bistu.edu.cn

* Correspondence: doulihua@bit.edu.cn; Tel./Fax: +86-010-6891-4577

Received: 20 January 2018; Accepted: 9 March 2018; Published: 16 March 2018

Abstract: A snake robot is a type of highly redundant mobile robot that significantly differs from a tracked robot, wheeled robot and legged robot. To address the issue of a snake robot performing self-localization in the application environment without assistant orientation, an autonomous navigation method is proposed based on the snake robot's motion characteristic constraints. The method realized the autonomous navigation of the snake robot with non-nodes and an external assistant using its own Micro-Electromechanical-Systems (MEMS) Inertial-Measurement-Unit (IMU). First, it studies the snake robot's motion characteristics, builds the kinematics model, and then analyses the motion constraint characteristics and motion error propagation properties. Second, it explores the snake robot's navigation layout, proposes a constraint criterion and the fixed relationship, and makes zero-state constraints based on the motion features and control modes of a snake robot. Finally, it realizes autonomous navigation positioning based on the Extended-Kalman-Filter (EKF) position estimation method under the constraints of its motion characteristics. With the self-developed snake robot, the test verifies the proposed method, and the position error is less than 5% of Total-Traveled-Distance (TDD). In a short-distance environment, this method is able to meet the requirements of a snake robot in order to perform autonomous navigation and positioning in traditional applications and can be extended to other familiar multi-link robots.

Keywords: snake robot; MEMS IMU; inertial navigation system mechanization; kinematics model

1. Introduction

The snake robot, which is based on the biological characteristics of snakes, constitutes an important branch of bionic robots. Prof. Hirose, S. developed the first snake robot in 1972 [1]. A snake robot is significantly different from a tracked robot, wheeled robot and legged robot, being a mobile robot with high redundancy. Because of the multi-joint flexible structure design, a snake robot has the advantage of multi-gait motion and the ability to adapt to a complex unknown environment, and can be widely used in disaster rescue, underwater survey, industrial testing and other special environments that traditional robots or humans cannot enter; as a result, increasing attention is being paid to snake robots [2,3].

In 1946, Gray, J., at Cambridge University, divided movement gaits into serpentine movement, rectilinear movement, concertina movement and sidewinding movement in the study of the biological nature of snakes [4]. Shigeo Hirose at Tokyo University established a serpentine gait kinematics model with linkage structures based on the observation of biological snake movement processes and bone anatomy [1]. Liljebäck, P. et al. at the Norwegian University, analyzed the position relationship between a snake robot and obstacles, proposed an obstacle assistant movement gait in planar motion,

and built the kinematics and dynamics model for the snake robot [5,6]. In studying the snake robot's body, the team of Hirose, S. at Tokyo University developed a series of ACM snake robots, in which the ACM-R5 snake robot had amphibious motion ability [7]. A Carnegie Mellon University team studied a unified Snake robot with climbing ability [8]. The Norwegian Technology University developed the snake robot Anna Konda [9] for fire rescue and developed the snake robot Kulko [10] with stress perception.

The Snake robot's autonomous navigation and positioning is the premise of the autonomous movement in complex unknown environments [3]. At present, most research is related to the snake robot's structure, the movement gait, and the gait control method and robot body; however, most of these robots do not have the ability to perform autonomous navigation and positioning, which has hindered their use in complex unknown environments [2]. The only relevant work includes: Tanaka, M. et al., who developed a set of half self-help collision avoidance systems for a snake robot using LiDAR and realized the robot's autonomous motion in a limited space [11]; Tian, Y. et al. who proposed a SLAM algorithm using single LiDAR, with LiDAR fixed on the outside of the robot head joints, and generated a 2D environment map for navigation [12]; and Chavan, P., who realized mapping and navigation for a snake robot using an Ultrasonic-sensor and a PIR-sensor, along with wireless control through ZigBee [13]. Some sensors, such as LiDAR and Ultrasonic, can measure environment depth information, which can be used in a snake robot's navigation and positioning. However, these sensors have too large a volume and require too much energy to be used in a snake robot system. Fu, X. et al. proposed a snake robot control system based on a visual tracking algorithm and realized tracking ground path navigation using a monocular camera [14]. Xiao, X. et al. used two external cameras to detect a snake robot's posture and simplified the attitude estimation by putting fiducials on the snake robot [15]. Visual navigation has high requirements on light, which limits its use of the environment, and often tends to require a high computational effort, increasing the cost of the system. Ponte, H. et al. designed a triangulation measurement sensor that adapts to the size and power of the snake robot using a laser and a black and white camera; the robot can scan an environmental color 3D point cloud when the head is raised, with the robot's posture estimated using the kinematics equation and IMU data [16]. Ohno, K. et al. combined the information from an IMU and a TOF camera, performed estimation of the snake robot's trajectory, and completed 3D SLAM reconstruction [17]. A structured light sensor or a TOF camera combined with IMU information can realize the reconstruction of the environmental depth and robot's navigation and positioning; however, the algorithm used is complex, and the number of calculations is high. Billah, M.M. et al. proposed a navigation system for a snake robot using intelligent inertial sensors to well overcome the limits of Wi-Fi, RFID and other positioning systems; however, the paper did not mention the positioning accuracy [18]. Estébanez, J.G. designed a navigation system for snake robots using integrated GPS and IMU, and developed a kinematic physical model which simulated the robot motion and estimated the robot trajectory when the GPS signal reception was not able to [19]. Yang, W. et al. developed a type of snake robot motion-tracking system with a low-cost MEMS IMU that used three algorithms (low-pass filter, baseline calibration and Kalman filter) to remove noise from the IMU acceleration data and evaluated the tracking algorithm's efficiency using video tracking software; its maximum errors of velocity and position were 10.93% and 12.23% TTD, respectively. However, the system only considered the accelerometer data, which can only track linear motion [20]. In general, LiDAR/Ultrasonic/Visual/Structured light/TOF camera sensors tend to be bulky, highly energy consuming, or require high computational effort. The MEMS inertial sensor not only has a small size and low energy consumption, but is also low in computational load; this paper is based on the work of this sensor.

By analyzing the research work of scholars in the robot autonomous navigation positioning field, it can be seen that there is great difficulty in effectively achieving autonomous positioning, especially when the snake robot implements positioning without an external assistant in complex unknown environments. Inspired by the dead reckoning method for people [21], the main contribution

of the paper is the proposition of an autonomous navigation positioning method based on the constraints of the snake robot's motion characteristics, using the robot's own MEMS IMU to realize the snake robot's autonomous navigation and positioning with non-nodes (used to differentiate from similar Wi-Fi/ZigBee/Beacons and other navigation systems that need nodes and external assistant) and a non-external assistant. The system has small volume, low power consumption, and only requires a simple installation and a small amount of calculation; thus, the system can meet the requirement of the snake rescue robot's autonomous navigation and positioning, and can be extended to other similar robot fields.

2. Analysis of the Snake Robot's Motion Characteristics

A snake robot has multiple movement gaits, and the serpentine movement is the most studied and most efficient of all two-dimensional gaits. It transfers with a lateral wave, and the motion curve is similar to a sine curve. The phase and amplitude fluctuations change over time. The snake robot navigation algorithms in this paper are based on the study of the aforementioned gait. The navigation coordinate system (n coordinate) defines the origin as the navigation system's point P , with axes pointing north, east and the local vertical direction (down); the Carrier coordinate system (b coordinate) is shown in Figure 1; the Carrier coordinate system and the snake robot body are fixed, with axes pointing to the front, right and bottom of the snake robot's movement direction.

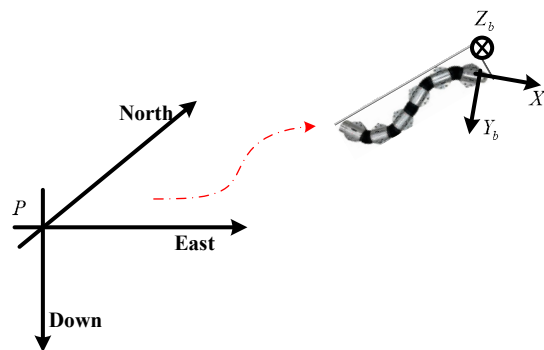


Figure 1. Navigation coordinate system of the snake robot.

2.1. Snake Robot Kinematics Model

The serpentine curve was first proposed by Hirose [1]; Ma et al. proved that the curve had the best movement efficiency from continuous muscle contraction [22]. The serpentine movement Serpenoid curve's curvature equation is expressed as [23]:

$$\rho = -\alpha b \sin(bs) \quad (1)$$

where ρ is the curvature of the curve, α is the initial amplitude angle of curve, b is the adjustable constant, and s is the length of curve.

We assume that the robot consists of N joints of length $2l$, and the joints are connected by $N - 1$ motors. Every joint has the same mass m except for the head joint; the mass of each joint is uniformly distributed, and the center of mass is located at the center point (i.e., at a distance l from the two ends of the joint). Neglecting the width of the snake robot, the robot's movement is shown in Figure 2. The snake robot head mass is m_h , IMU is in the head of the centroid position. As shown in Figure 3, x_h, y_h corresponds to the coordinate position for the snake robot head at a certain moment, x_i, y_i to the coordinate position for the joint i . The definition of S facilitates the calculation of the angle between

the relative motion directions of robot joints, whereby joint i corresponds to $S = i \times 2l - 1$. This angle is represented by $\theta(s)$ in Figure 2 and can be expressed as:

$$\theta(s) = \alpha \cos(bs) \tag{2}$$

The angle between the joints shown in Figure 3 can be expressed as:

$$\psi = \theta(s + l) - \theta(s - l) = -2\alpha \sin(bl) \sin(bs) \tag{3}$$

According to the serpentine curve and variable S (the length of the snake robot), we can obtain the input angle formula of each joint (as shown in Figures 2 and 3: θ_h is the angle between the head and the motion direction, and ϕ_i is the angle between joint i and the motion direction):

$$\begin{aligned} \theta_h(t) &= A \sin(\omega t - 0.5\vartheta) \\ \phi_i(t) &= A \sin(\omega t + (i - 1)\vartheta) \end{aligned} \tag{4}$$

where A is the amplitude, $A = -2\alpha \sin(bl)$, $bs = \omega t$, ω is the angular velocity, ϑ is the phase, $i = 1 : N$, and N is the total number of joints. Angular changes between adjacent joints drive the snake robot's serpentine movement. As shown in Figure 3, the direct geometric relationship expression of each joint position is expressed as:

$$\left. \begin{aligned} x_i &= x_h + 2l \cos(\theta_h) + 2l \sum_{k=1}^{i-1} \cos(\theta_h + \sum_{j=1}^k \phi_j) + l \cos(\theta_h + \sum_{k=1}^i \phi_k) \\ y_i &= y_h + 2l \sin(\theta_h) + 2l \sum_{k=1}^{i-1} \sin(\theta_h + \sum_{j=1}^k \phi_j) + l \sin(\theta_h + \sum_{k=1}^i \phi_k) \end{aligned} \right\} \tag{5}$$

The initial geometry center position of the snake robot's head is the origin. We set the global coordinate frame as the position of the snake robot's head and the movement direction as the x -axis, with the y -axis being vertical to the x -axis.

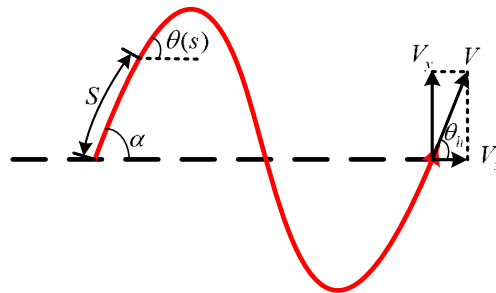


Figure 2. Serpennoid curve.

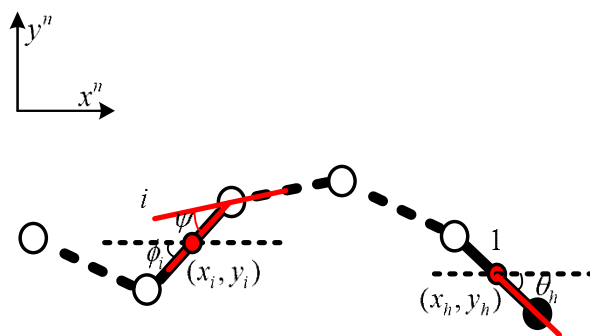


Figure 3. Relative joint angle.

The input joint angle can be deduced from Formula (5) and is used to control the snake robot's movement. The method of changing the angle function to control the snake robot's motion path is called the center adjustment method [24]. The mathematical expression is:

$$\left. \begin{aligned} \theta_h(t) &= A \sin(\omega t - 0.5\theta) + \gamma_h \\ \phi_i(t) &= A \sin(\omega t + (i-1)\theta) + \gamma_i \end{aligned} \right\} \quad (6)$$

If the input joint angle changes γ , the input angle of symmetry axis will deviate from the desired location (zero position) value γ . In this manner, the direction of the snake robot is changed to move toward the navigation points. γ is called the adjustment factor of direction.

2.2. Analysis of the Snake Robot's Motion Restraint Characteristics

Khatib first applied the artificial potential field (APF) method in path planning guidance [25]. The basic principle is that obstacles have repulsion, targets have gravitational pull. An obstacle is in the area of high potential energy, releasing energy outward. A target is in the region of lower potential energy, absorbing the external energy. The potential field force can be represented as:

$$E_T = k \times d^2 \quad (7)$$

where k is a persistent target point, and d is the distance from the robot's head to the target point.

We assume that the robot's target environment is static, and the target coordinates are known. In the environment, the snake robot is affected by the obstacles' repulsive forces and the targets' attractive forces. The paper only considers the attractive forces' influence and assumes that:

$$F(t) = p \times E_T(t) \quad (8)$$

where $F(t)$ is the attractive force, and p is a constant related to the robot.

As shown in Figure 4, when the snake robot is on S' , it is affected by the attractive force from the target and turns an angle γ . The velocity of snake robot's head is v_0 , the angle from the x -axis is θ_h , the attractive force from the target to the robot on S' is F , the angle from the x -axis is α , and β is the angle between the attractive force F and velocity v_0 .

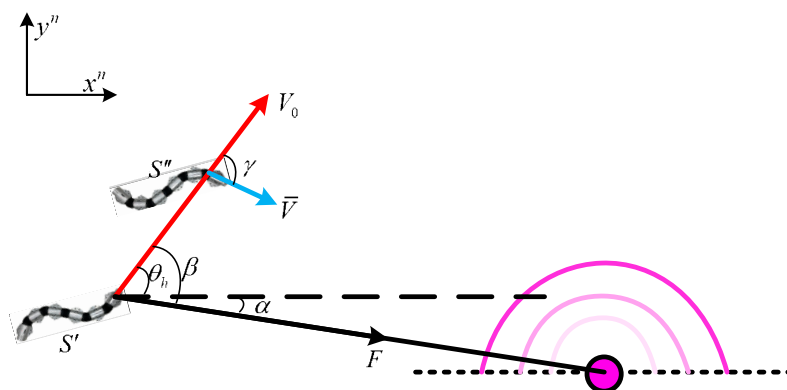


Figure 4. Relationship between the distance and the attractive force.

With no attractive force, the robot's movement velocity in a short period dt is v_0 . Under an attractive force, the velocity will become \bar{v} , and the corresponding velocity v_0 will change angle γ to move toward the target. To calculate the change angle γ , the relationship between force and velocity is shown in Figure 5:

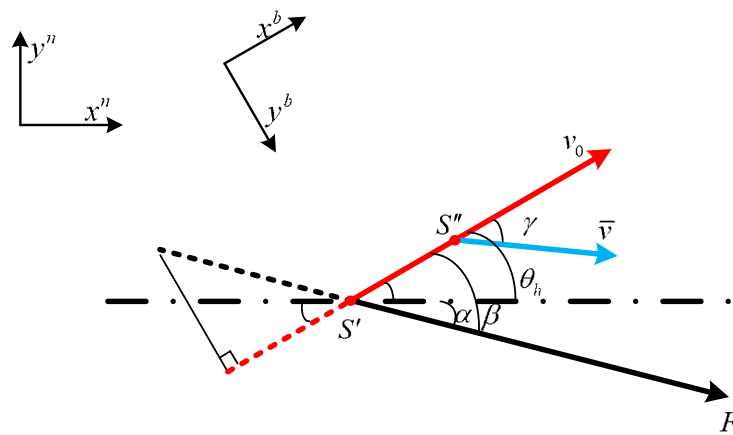


Figure 5. Relationship between the Force and the Velocity.

To determine γ , the attractive force F is decomposed into two directions: one is parallel to v_0 , and the other is vertical to v_0 . Within a short time dt , the snake robot is under a uniformly accelerated process in both directions. According to Newton's second law, we obtain:

$$\left. \begin{aligned} v_{x^b} &= v_0 + F \sin(\beta) dt / m_h \\ v_{y^b} &= F \cos(\beta) dt / m_h \\ \gamma &= \arctan(v_{x^b} / v_{y^b}) \\ \bar{v} &= \sqrt{v_{x^b}^2 + v_{y^b}^2} \end{aligned} \right\} \quad (9)$$

where m_h is the mass of the robot's head, v_{x^b} is the component along the x^b axis, and v_{y^b} is the component along the y^b axis.

$$\left. \begin{aligned} \gamma &= \arctan((v_0 m_h + F \sin(\beta) dt) / F \cos(\beta) dt) \\ \bar{v} &= \sqrt{m_h^2 v_0^2 + (F dt)^2 + 2 m_h v_0 F \cos(\beta) dt / m_h} \end{aligned} \right\} \quad (10)$$

Substituting Formula (10) into Formula (6), we obtain the relationship between the direction adjustment function of the snake robot and the potential force F .

Formula (10) can be extended to multi-targets:

$$\left. \begin{aligned} \gamma &= \arctan((v_0 m_h + \sum_{j=1}^n F_j \sin(\beta) dt) / \sum_{j=1}^n F_j \cos(\beta) dt) \\ \bar{v} &= \sqrt{m_h^2 v_0^2 + (\sum_{j=1}^n F_j dt)^2 + 2 m_h v_0 \sum_{j=1}^n F_j \cos(\beta) dt / m_h} \end{aligned} \right\} \quad (11)$$

where n is the number of targets.

The influence of the target attractive force field F on the snake robot's motion can be obtained by substituting Formula (10) into Formula (6). By adjusting γ , the robot can be controlled to avoid a collision. The snake robot's head motion adjusting parameters are as shown in (12) [26]:

$$\theta_h(t) = A \sin(\omega t - 0.5\theta) + \arctan((v_0 m_h + F \sin(\beta) dt) / F \cos(\beta) dt) \quad (12)$$

From Formula (12) the potential field attractive force of target F can change the snake robot's head motion direction $\theta_h(t)$ and movement speed. In practice, the snake robot calculates the direction to the target according to its speed, posture and position information. It is controlled to move in the correct direction by adjusting steering force F_k to simulate the attractive force from the potential field.

Although this method has been applied in the field of vehicle and pedestrian navigation for a long time, it has not been applied to snake robots because of the uniqueness of the movement characteristics of snake robots. The novelty of this paper is in applying this technology to a snake robot's navigation by studying the motion model of the snake robot and using the above models, the snake robot motion controller can be designed to realize the robot's multi-gait flexible movements.

2.3. Error Propagation Properties

Formula (12) contains the relationship between the control force F_k and the angle that must be adjusted by the controller to realize the movement of the snake robot. In space, the snake robot's own coordinate system is defined as the carrier coordinate system using superscript b . The navigation coordinate system is defined as the front right down coordinate system along the direction of the robot's head direction with superscript n . These coordinate systems are shown in Figure 1. The snake robot's movement process in the space can be simplified as follows:

$$\begin{cases} \mathbf{v}^n(t) = \int_{t_0}^t \mathbf{a}^n(t) dt + \mathbf{v}^n(t_0) \\ \mathbf{r}^n(t) = \int_{t_0}^t \mathbf{v}^n(t) dt + \mathbf{r}^n(t_0) \end{cases} \quad (13)$$

where $\mathbf{v}(t)$ is the snake robot motion vector speed at moment t , and $\mathbf{r}(t)$ is its motion vector position at moment t . In the process of the snake robot's movement, its error of velocity, attitude and angle can be described as follows:

$$\begin{cases} \delta\boldsymbol{\varphi}_{k+1} = \delta\boldsymbol{\varphi}_k + \mathbf{C}_{b_k|k-1}^n \delta\boldsymbol{\omega}_k^b \Delta t + \mathbf{w}_{\boldsymbol{\varphi}_k} \\ \delta\mathbf{v}_{k+1} = \delta\mathbf{v}_k + \mathbf{C}_{b_k|k-1}^n \delta\mathbf{a}_k^b \Delta t + \mathbf{w}_{\mathbf{v}_k} \\ \delta\mathbf{r}_{k+1} = \delta\mathbf{r}_k + \delta\mathbf{v}_k \Delta t + \mathbf{w}_{\mathbf{r}_k} \end{cases} \quad (14)$$

where $\delta\boldsymbol{\varphi}_k$ is the attitude angle error at the time k and is defined as a column vector, containing the roll angle, pitching angle and yaw angle. $\mathbf{C}_{b_k|k-1}^n$ is the direction cosine matrix at the time k , the details of which will be presented in the next section. $\delta\boldsymbol{\omega}_k^b$ is the bias of the gyro's output. $\delta\mathbf{v}_k$ is the three-axis velocity under the navigation system. $\delta\mathbf{r}_k$ is the three-axis position under the navigation system. $\delta\mathbf{a}_k^b$ is the bias of the accelerometer's output. \mathbf{w}_\bullet is the corresponding system error.

3. Navigation Method under the Constraint of the Snake Robot's Motion Features

3.1. Mechanical Arrangement of the Strapdown Navigation System

The concept of the snake robot using the MEMS inertial measurement unit for navigation and positioning originally comes from the traditional robot's navigation system design. The snake robot's mechanical arrangement of the strapdown navigation system using MEMS-IMU in this paper is shown in Figure 6 [27]. Acceleration and angular velocity values of the carrier are measured separately by an accelerometer and gyroscope attached to a snake robot. The attitude and position of the carrier are calculated by the navigation computer through attitude calculation and force-resolution.

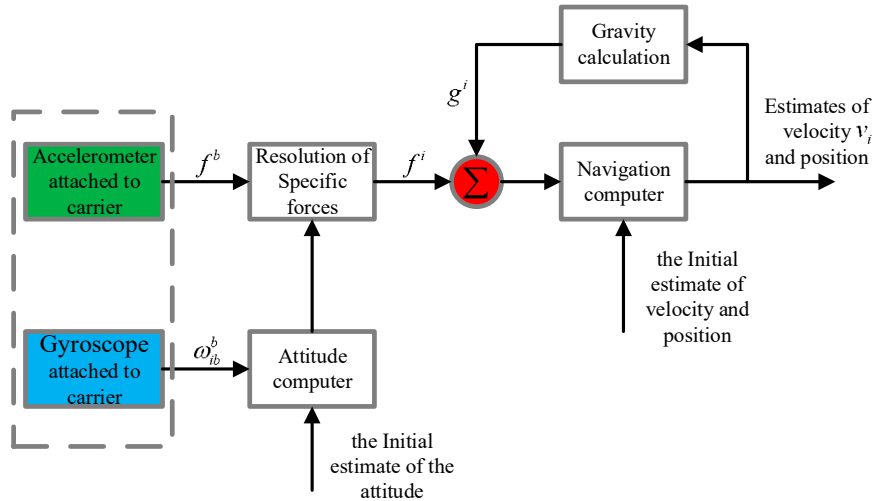


Figure 6. Arrangement of the machinery of the strapdown inertial navigation system.

The snake robot's navigation system can complete the initial alignment in 10 s after power on. According to the angular rate information, it updates the snake robot's posture using the quaternion method. The quaternion update algorithm is as follows, calculating the angular increment first:

$$\Delta = \sqrt{(\omega_x T_m)^2 + (\omega_y T_m)^2 + (\omega_z T_m)^2} \quad (15)$$

where Δ is the angular increment, $\omega_x, \omega_y, \omega_z$ is the three-axis angular velocity scalar value under the carrier system. T_m is the sampling time. Next, the quaternion is updated:

$$\begin{cases} q_{1|k+1} = (1 - \frac{\Delta^2}{8} + \frac{\Delta^4}{384})q_{1|k} - \omega_x T_m (0.5 - \frac{\Delta^2}{48})q_{2|k} - \omega_y T_m (0.5 - \frac{\Delta^2}{48})q_{3|k} - \omega_z T_m (0.5 - \frac{\Delta^2}{48})q_{4|k} \\ q_{2|k+1} = (1 - \frac{\Delta^2}{8} + \frac{\Delta^4}{384})q_{2|k} + \omega_x T_m (0.5 - \frac{\Delta^2}{48})q_{1|k} + \omega_z T_m (0.5 - \frac{\Delta^2}{48})q_{3|k} - \omega_y T_m (0.5 - \frac{\Delta^2}{48})q_{4|k} \\ q_{3|k+1} = (1 - \frac{\Delta^2}{8} + \frac{\Delta^4}{384})q_{3|k} + \omega_y T_m (0.5 - \frac{\Delta^2}{48})q_{1|k} - \omega_z T_m (0.5 - \frac{\Delta^2}{48})q_{2|k} + \omega_x T_m (0.5 - \frac{\Delta^2}{48})q_{4|k} \\ q_{4|k+1} = (1 - \frac{\Delta^2}{8} + \frac{\Delta^4}{384})q_{4|k} - \omega_x T_m (0.5 - \frac{\Delta^2}{48})q_{3|k} + \omega_y T_m (0.5 - \frac{\Delta^2}{48})q_{2|k} - \omega_z T_m (0.5 - \frac{\Delta^2}{48})q_{1|k} \end{cases} \quad (16)$$

where $q_{1|k+1}$ is the first value of quaternion at time $k + 1$ and so on; $q_{1|k}$ is the first value of quaternion at time k . Next, normalize its quaternion:

$$A = \sqrt{q_{1|k+1}^2 + q_{2|k+1}^2 + q_{3|k+1}^2 + q_{4|k+1}^2} \quad (17)$$

$$\begin{cases} q'_{1|k+1} = \frac{q_{1|k+1}}{A} \\ q'_{2|k+1} = \frac{q_{2|k+1}}{A} \\ q'_{3|k+1} = \frac{q_{3|k+1}}{A} \\ q'_{4|k+1} = \frac{q_{4|k+1}}{A} \end{cases} \quad (18)$$

where A is the square sum of the quaternion at time $k + 1$. $q'_{1|k+1}$ is the first normalized value of the quaternion at time $k + 1$, and so on. Next, the direction cosine matrix is obtained as follows:

$$C_b^n = \begin{bmatrix} (q'_{1|k+1})^2 + (q'_{2|k+1})^2 - (q'_{3|k+1})^2 - (q'_{4|k+1})^2 & 2(q'_{2|k+1}q'_{3|k+1} - q'_{1|k+1}q'_{4|k+1}) & 2(q'_{2|k+1}q'_{4|k+1} + q'_{1|k+1}q'_{3|k+1}) \\ 2(q'_{2|k+1}q'_{3|k+1} + q'_{1|k+1}q'_{4|k+1}) & (q'_{1|k+1})^2 - (q'_{2|k+1})^2 + (q'_{3|k+1})^2 - (q'_{4|k+1})^2 & 2(q'_{3|k+1}q'_{4|k+1} - q'_{1|k+1}q'_{2|k+1}) \\ 2(q'_{2|k+1}q'_{4|k+1} - q'_{1|k+1}q'_{3|k+1}) & 2(q'_{3|k+1}q'_{4|k+1} + q'_{1|k+1}q'_{2|k+1}) & (q'_{1|k+1})^2 - (q'_{2|k+1})^2 - (q'_{3|k+1})^2 + (q'_{4|k+1})^2 \end{bmatrix} \quad (19)$$

Next, we can obtain the corresponding attitude information:

$$\begin{cases} \psi = \arctan\left(\frac{c_{32}}{c_{33}}\right) \\ \theta = \arcsin(-c_{31}) \\ \gamma = \arctan\left(\frac{c_{21}}{c_{11}}\right) \end{cases} \quad (20)$$

Compensating the gravity according to the above matrix information and specific information, we can obtain the acceleration of the navigation frame and compute the velocity and position information at the same time.

$$\widetilde{a}_k = \mathbf{C}_{b|k-1}^n \cdot a_k^b - [0 \ 0 \ g]^T \quad (21)$$

$$\mathbf{v}_{k|k-1} = \mathbf{v}_{k-1|k-1} + \widetilde{a}_k \cdot \Delta t \quad (22)$$

$$\mathbf{r}_{k|k-1} = \mathbf{r}_{k-1|k-1} + \mathbf{v}_{k|k-1} \cdot \Delta t \quad (23)$$

where $a_k^b = [f_x, f_y, f_z]^T$ is the accelerometer's output value after a compensation filter.

3.2. Position Estimation Filter Design

At the end of each movement cycle, the snake robot uses the controller to perform zero-velocity control, directing the robot to be static in milliseconds. It corrects the navigation error in the static moment. While the snake robot is static for a short time, its velocity and angular velocity are zero. According to the state characteristics, the mechanical arrangement design algorithm is shown in Figure 7. Using the accelerometer and gyroscope to acquire the acceleration and angular velocity information of the snake robot, the momentary stationary state of the snake robot can be obtained by analyzing the periodic data. The stationary signal is used to drive the Kalman filter to integrate its own strapdown solution. To achieve the final accurate speed, position and attitude information output. In the arrangement, the most important aspect is its core filter design [28].

The MEMS inertial measurement unit is known to spread gradually over time, especially for a snake robot that is rapidly changing in a small cycle using the MEMS strapdown method to measure the speed, posture and position information. However, if, in a short period of time, the integral error is eliminated in a regular time, then the operation information of the snake robot can be effectively measured within a certain scope; the constraint process schematic is shown in Figure 8 [29]. If only the traditional inertial navigation solution is used, the position error will increase with the accumulation of time. Because the snake robot has the characteristics of periodic motion, we zero the speed and angular velocity to carry on the error restraint, which greatly reduces the position error.

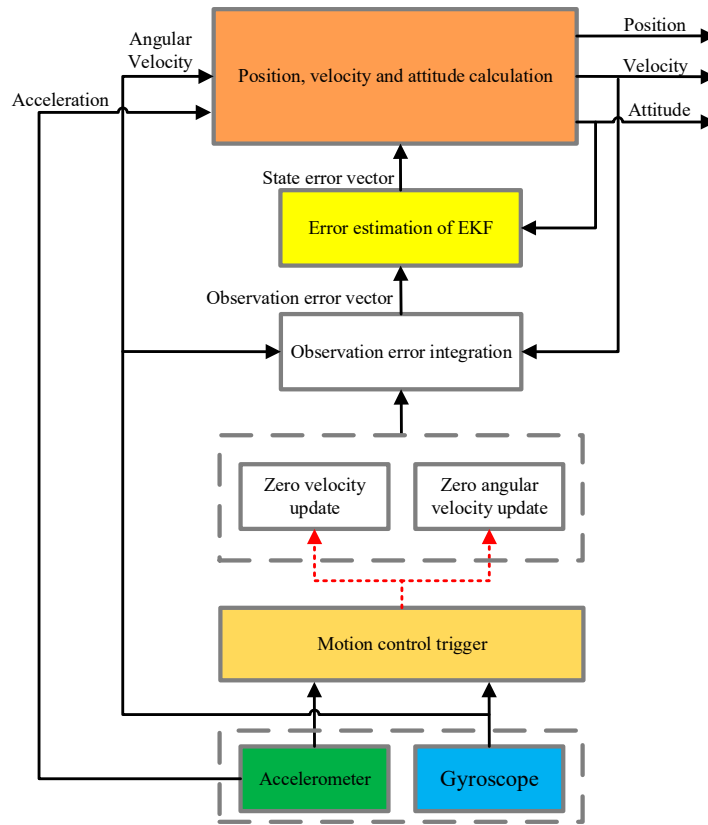


Figure 7. Flow diagram of the inertial navigation.

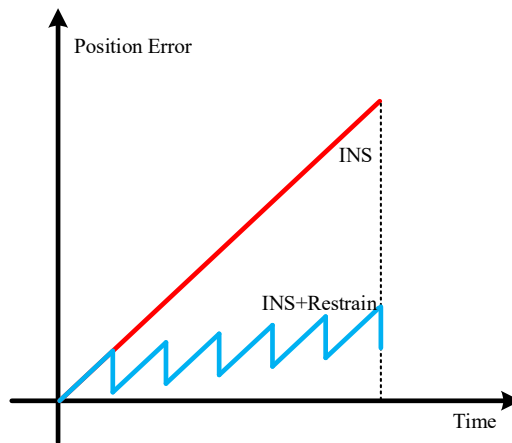


Figure 8. Diagram of the restraining process.

3.2.1. Establish the State Equation

According to the error propagation characteristics shown in Equation (14), build its state equation. First, choose its state variables as follows:

$$\mathbf{X}_k = \left[\delta\boldsymbol{\varphi}_k \quad \delta\boldsymbol{\omega}_k^b \quad \delta\mathbf{r}_k \quad \delta\mathbf{v}_k \quad \delta\mathbf{a}_k^b \right]^T \tag{24}$$

to obtain:

$$\begin{cases} \mathbf{X}_{k+1} = f(\mathbf{X}_k) + \mathbf{G}_k \mathbf{W}_k \\ \mathbf{Z}_{k+1} = \mathbf{H}_k \mathbf{X}_k + \mathbf{V}_k \end{cases} \tag{25}$$

where \mathbf{W}_k is the system process noise matrix given by $\mathbf{W}_k = [-\mathbf{C}_b^n \boldsymbol{\omega}^b \quad -\mathbf{C}_b^n \mathbf{a}^b]$. \mathbf{G}_k is the corresponding coefficient of the noise matrix. \mathbf{Z}_k is the observed quantity, \mathbf{H}_k is the observation matrix, and \mathbf{V}_k is the observation noise matrix. The $f(\mathbf{X}_k)$'s specific definition is as follows:

$$f(\mathbf{X}_k) = \begin{Bmatrix} \delta \boldsymbol{\varphi}_k + \mathbf{C}_b^n \cdot \delta \boldsymbol{\omega}_k^b \cdot \Delta t \\ \delta \boldsymbol{\omega}_k^b \\ \delta \mathbf{r}_k + \delta \mathbf{v}_k \cdot \Delta t \\ \delta \mathbf{v}_k + \mathbf{C}_b^n \cdot \delta \mathbf{a}_k^b \cdot \Delta t \\ \delta \mathbf{a}_k^b \end{Bmatrix} \quad (26)$$

3.2.2. Extended Kalman Filter

EKF is representative of the traditional nonlinear estimation. The basic approach is to perform partial linearization for the nonlinear state function and measurement function according to the first-order Taylor polynomial expansion and then apply the linear system Kalman filtering equation. According to this approach, we obtain:

$$\Phi_k = \frac{\partial f(\mathbf{X}_k)}{\partial \mathbf{X}_k} = \begin{bmatrix} 3 \times 3 & \Delta t \cdot \mathbf{C}_{b|k-1}^n & 0 & 0 & 0 \\ 0 & 3 \times 3 & 0 & 0 & 0 \\ 0 & 0 & 3 \times 3 & \Delta t \cdot 3 \times 3 & 0 \\ -\Delta t \cdot \mathbf{S}(\mathbf{a}'_k^n) & 0 & 0 & 3 \times 3 & \Delta t \cdot \mathbf{C}_{b|k-1}^n \\ 0 & 0 & 0 & 0 & 3 \times 3 \end{bmatrix} \quad (27)$$

According to state error vector matrix, the linearization state transfer model is given as below:

$$\delta \mathbf{x}_{k|k-1} = \Phi_k \delta \mathbf{x}_{k-1|k-1} + \mathbf{w}_{k-1} \quad (28)$$

where $\delta \mathbf{x}_{k|k-1}$ is the predicted state error; $\delta \mathbf{x}_{k-1|k-1}$ is the state error after filtering at time $k-1$; \mathbf{w}_{k-1} is process noise and can be expressed as a collaborator's variance matrix:

$$\mathbf{Q}_K = E(\mathbf{w}_k \mathbf{w}_k^T) \quad (29)$$

where $\mathbf{C}_{b|k-1}^n$ is the transfer matrix; Δt is the sampling time; $\mathbf{S}(\mathbf{a}'_k^n)$ is a skew-symmetric matrix of the acceleration, which is used to estimate the pitch angle and roll angle of the transducer and is given by:

$$\mathbf{S}(\mathbf{a}'_k^n) = \begin{bmatrix} 0 & -a_{zk} & a_{yk} \\ a_{zk} & 0 & -a_{xk} \\ -a_{yk} & a_{xk} & 0 \end{bmatrix} \quad (30)$$

where \mathbf{a}'_k^n is the acceleration value under the navigation system. The specific expression is:

$$\mathbf{a}'_k^n = \mathbf{C}_{b|k-1}^n \cdot \mathbf{a}'_k^b = (a_{xk}, a_{yk}, a_{zk}) \quad (31)$$

The measurement model is as follows:

$$\mathbf{z}_k = \mathbf{H} \delta \mathbf{x}_{k|k} + \mathbf{v}_k \quad (32)$$

where \mathbf{z}_k is the observation matrix; \mathbf{H} is the measurement matrix; \mathbf{v}_k is the measurement noise, and its covariance matrix is expressed as:

$$\mathbf{R}_k = E(\mathbf{v}_k \mathbf{v}_k^T) \quad (33)$$

The state update equation is:

$$\delta \mathbf{x}_{k|k} = \delta \mathbf{x}_{k|k-1} + \mathbf{K}_k \cdot \left[\mathbf{m}_k - \mathbf{H} \delta \mathbf{x}_{k|k-1} \right] \quad (34)$$

where \mathbf{K}_k is the Kalman gain, \mathbf{m}_k is a 9D measurement matrix at zero velocity, including three errors at zero velocity, three errors at zero angular velocity and three errors on the attitude angle. \mathbf{K}_k is expressed as:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}^T \left(\mathbf{H} \mathbf{P}_{k|k-1} \mathbf{H}^T + \mathbf{R}_k \right)^{-1} \quad (35)$$

where $\mathbf{P}_{k|k-1}$ is the estimate of the covariance matrix, and its expression is calculated according to the information of time $k - 1$ and is given as follows:

$$\mathbf{P}_{k|k-1} = \Phi_{k-1} \mathbf{P}_{k-1|k-1} \Phi_{k-1}^T + \mathbf{Q}_{k-1} \quad (36)$$

Last, update the covariance matrix:

$$\mathbf{P}_{k|k} = (\mathbf{I}_{15 \times 15} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_{k|k-1} (\mathbf{I}_{15 \times 15} - \mathbf{K}_k \mathbf{H})^T + \mathbf{R}_k \quad (37)$$

Repeating the above steps, the conventional EKF function of the fusion the snake robot's motion characteristics can be realized.

3.3. Analysis of the Trigger Correction Mechanism under the Restriction of Snake Robot Behavior

3.3.1. Velocity-Assisted Correction of the Snake Robot's Movement

Because of the sensor's measurement error, noise and algorithm error, the calculated speed value is not zero at speed zero. Speed-assisted correction in the design just uses this principle, which is referred to as zero-speed correction, or Zero-Velocity-Update (ZUPT). When the static snake robot is detected, obtain the error of the calculated velocity value by performing coordinate transformation and integral operation on the accelerometer's output. The error is the observation value of the filter. During the static moment, the velocity output error is as follows:

$$\Delta \mathbf{v}_k = \mathbf{v}_k - \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \quad (38)$$

ZUPT's corresponding observed value and the observation matrix is as follows:

$$\begin{aligned} \mathbf{Z}_k &= [\Delta \mathbf{v}_k]^T = [\mathbf{v}_k]^T - \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\ \mathbf{H}_k &= \begin{bmatrix} \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{O}_{3 \times 3} \end{bmatrix} \end{aligned} \quad (39)$$

3.3.2. Velocity-Assisted Correction and Fusion Angular Velocity-Assisted Correction

Similar to the velocity-assisted correction principle, when the snake robot is detected at the static moment in the serpentine cycle, the output angular velocity is zero, in theory. As a result of the sensor's measurement error, noise and algorithm error, the calculated angular velocity value is not zero. Zero-presents-rate-update (ZARU) refers to the Zero angular velocity correction, i.e., the error of the gyroscope's angular velocity output is used as the observed quantity for the filter when the snake robot motion model detects the IMU to be static.

ZARU's corresponding observed value and the observation matrix is as follows:

$$\begin{aligned} \mathbf{Z}_k &= [\Delta \mathbf{w}_k]^T = [\mathbf{w}_k]^T - \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \\ \mathbf{H}_k &= \begin{bmatrix} \mathbf{O}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} & \mathbf{O}_{3 \times 3} \end{bmatrix} \end{aligned} \quad (40)$$

To achieve a higher precision, using ZUPT and ZARU at the same time, the corresponding observed value and the observation matrix is as follows:

$$Z_k = \begin{bmatrix} \Delta v_k & \Delta w_k \end{bmatrix}^T = \begin{bmatrix} v_k & w_k \end{bmatrix}^T$$

$$H_k = \begin{bmatrix} O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 3} & I_{3 \times 3} & O_{3 \times 3} \\ O_{3 \times 3} & I_{3 \times 3} & O_{3 \times 3} & O_{3 \times 3} & O_{3 \times 3} \end{bmatrix} \tag{41}$$

The process of the method is shown in Figure 9: This figure details the navigation solution process for zero speed and zero angular velocity correction. Each periodic static motion of the snake robot is triggered by a motion controller, and then the Kalman filter observations are corrected at this moment to correct the attitude, velocity and position information of the navigation solution and improve the navigation accuracy.

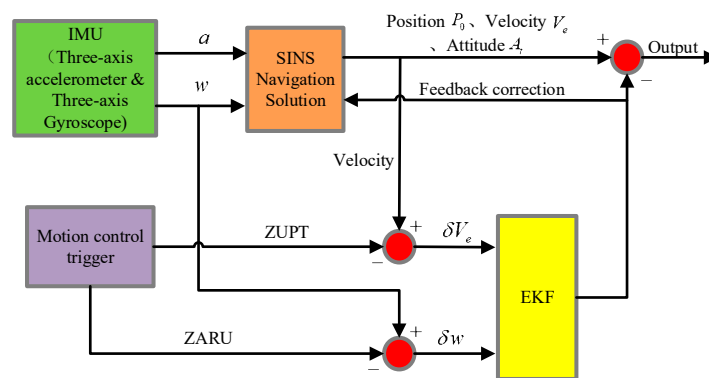


Figure 9. Flow diagram of the correction solution.

4. Experiment

4.1. Prototype Development

The snake robot consists of a total of 6 joints, and 5 sets of servos. In each joint, there is a Ni-MH battery pack and a slave controller. At each link, two Futaba S9157 servos are mounted orthogonally, each servo with a speed of 0.14 s/60° @ 6.0 V and a torque of 30.6 kg·cm @ 6.0 V. The snake robot’s head structure mainly includes an IMU system (Figure 10) and master controller circuits. Slave controller and the IMU system communicate with the master controller through the RS-422 bus. The sensor head also integrates a high-resolution camera that can provide a video image for remote operation personnel, LED lighting for the camera system, IR sensors for obstacle avoidance and a Ni-MH battery pack that is used as a power supply for the entire system.

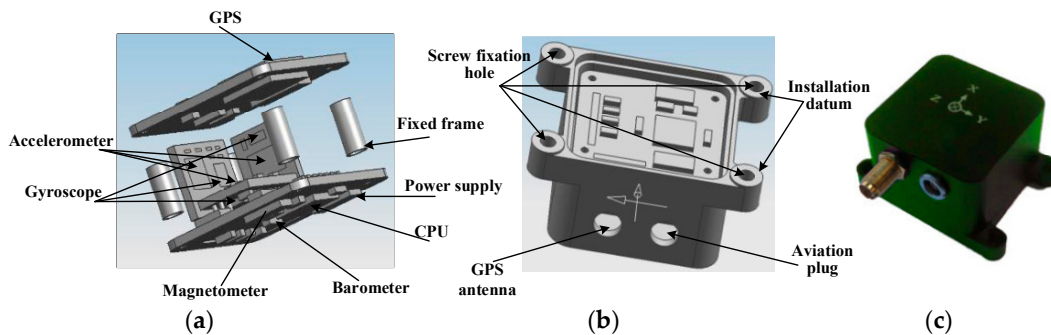


Figure 10. IMU and internal structure. (a) Exploded views of the IMU; (b) Assembly views of the IMU; (c) Overall views of the IMU.

4.1.1. Design Process

3D model printing technology is used in the iterative design of the snake robot. It can reduce the design cycle and cost. Ultimately, the snake robot's shell and body use aluminum alloy to maintain the structural strength and decrease the overall weight of the snake robot to improve its movement flexibility. The robot parts all use socket head cap screw fasteners, which is convenient for disassembly. The internal circuit and battery of the joint are fixed on the Teflon elastic double O-type frame to ensure its shake-resistance and stability. In each of the joints of the external structure, an O-type rubber ring is used to prevent water and other debris from entering the robot. In the joints of the movement mechanism, a steel wire-reinforced canvas bellows material is used to ensure movement flexibility, as well as waterproofing and wear-resistance capability. The details of the connections between the structures are shown in Figure 11. The snake robot's head is a specially designed scratch-resistant glass. It improves the robot's environmental compatibility, providing features such as anti-scratch and wear-resistance abilities, which do not decrease the camera transmission efficiency and maintains acceptable distortion. The final installed and integrated snake robot is shown in Figure 12.

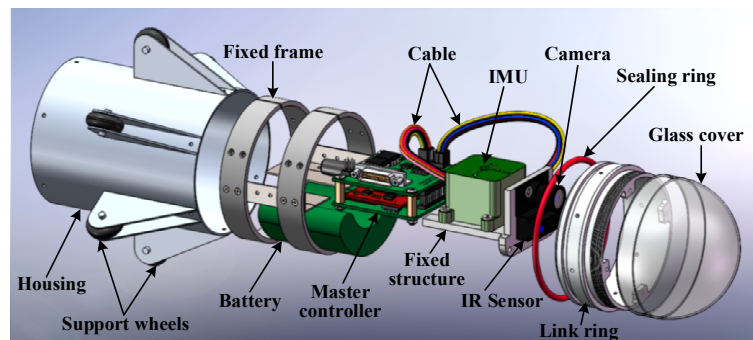


Figure 11. Exploded views of the sensor head.



Figure 12. Snake robot body.

4.1.2. IMU System Sensor

The IMU system mainly comprises three high-precision single-axis accelerometers (Model: ADXL103, Analog Devices, Inc., Norwood, MA, USA) and gyroscopes (Model: ADXRS642, Analog Devices, Inc., Norwood, MA, USA), GPS, a barometer, a three-axis magnetometer, a 32-bit ARM kernel main controller chip, a 16-bit AD data-acquisition circuit, and a secondary regulated power supply. The main technical indicators are as shown in Table 1. The main controller can measure the real-time sensor data, position, velocity and attitude information of the output carrier through EKF (note: in this paper, the GPS, barometer and magnetometer are turned off to study the inertial navigation precision). The 16-bit AD data-acquisition circuit can ensure the accuracy of the acquisition

data from the sensor. The secondary regulated power supply can ensure a stable power supply for the system. The IMU system fixes the snake robot's head structure through a specially designed frame to ensure its rigid connection and installation accuracy. The IMU measurement data can also be sent to the PC through the main controller at the same time via the ZigBee sensor. We also provide a variety of control mode designs for the system in order to adapt to different test scenarios, including automatic control, infrared, Bluetooth, PC serial port control and so on.

Table 1. Main technical indicators of the MEMS IMU.

	Specifications	Index Value
Accelerometer	Range	± 1.7 g
	Bias Instability	25 mg
	Non-linearity	<0.2%
	Velocity random walk	<0.75 m/s/h ^{1/2}
	Bandwidth	2500 Hz
Gyroscope	Range	$\pm 300^\circ$ /s
	Bias Instability	20° /h
	Non-linearity	<0.1%
	Angle random walk	$<0.02^\circ$ /s ^{1/2}
	Bandwidth	2000 Hz
Systems	Update Time	5 ms
	Power	0.15 A @ 5 Vdc

Finally, the host controller PCB includes all communication and power supply circuits and integrates the temperature and humidity sensors and current monitoring module to monitor the environment information and running status of the power supply system.

4.2. Experimental Verification

To verify the feasibility and precision of the above algorithm, considering the snake robot's running state in the actual use, we designed a linear motion, turn motion, turn-back motion and circular motion. The test scenario is on a rough cement road, as shown in Figure 13. The actual trajectory of the robot was obtained by tagging the snake robot's motion track points during the process.

Over the entire test process, the movement information from IMU was sent to the host computer for real-time monitoring and was collected via ZigBee.



Figure 13. Outdoor experiment.

4.2.1. Linear Motion Test

Before the linear motion, the snake robot's master controller set a target point, which was 15 m away from the origin on the host computer, and then set the snake robot to static for 200 ms at the end

of each movement cycle. The host computer collects real-time sensor data and monitors the running status of the snake robot using the zero speed and zero angular velocity constraints, respectively, to verify the algorithm's feasibility and accuracy.

Figure 14a (top) shows the original three-axis acceleration values collected by the sensor during the linear motion of the snake robot. Figure 14a (bottom) shows the original values of the three-axis angular velocity. The abscissa of the two graphs is the time-axis, the unit is second. Figure 14a shows that, in the process of snake robot motion, its acceleration and angular velocity are cyclically changed. The accelerometer x -axis and y -axis represent the snake robot's motion speed change state. The data noise is higher because of the rough uneven ground. The z -axis is mainly for Earth's gravity. The gyroscope's x -axis and y -axis represent the snake robot's pitch and roll motion. Because there is no pitch and roll motion in serpentine movement, the x -axis and y -axis angular rate value is small. The z -axis is a sensitive element of the snake robot in serpentine angle motion, as it is sufficiently sensitive to the serpentine movement periodicity of the robot.

Figure 14b shows the constraint points for each cycle in the snake robot's linear motion. The enlarged image is a magnified view of two cycles inside the red circle. Where the red curve is the x -axis acceleration value, the blue curve is the z -axis angular velocity value (shown as a gray curve in Figure 14a (bottom)), and the purple curve is the robot velocity value. Black dotted lines indicate the constraints (represented by 0 and 1), and the positions indicated by the arrows are the constraint points for each cycle. As can be seen from the enlarged image in Figure 14b, the snake robot is static for 200 ms in each cycle. During this period, zero speed and zero angular velocity correction constraints are triggered. As shown by the black dotted lines in Figure 14b, the algorithm can accurately detect every zero-velocity moment and correct in each cycle.

The images of Figure 14c (top, middle and bottom) show the covariance of position, velocity and attitude, respectively. Covariance is an important index for verifying whether the Kalman filter is normal. It can be seen from the figure that the covariances of position and attitude angle gradually become larger, and the covariance of velocity basically changes periodically. This is where the position and attitude of the robot keep accumulating changes, and the velocity error will be corrected every time the robot's zero velocity point is detected. The rapid convergence of covariance per cycle indicates that the Kalman filter is working properly and can effectively estimate the systematic error.

The images of Figure 14d (top and bottom) show the zero errors of the accelerometer and the gyroscope observed by the Kalman filter, respectively. It can be seen from the figure that the zero error can be estimated and converged at each cycle.

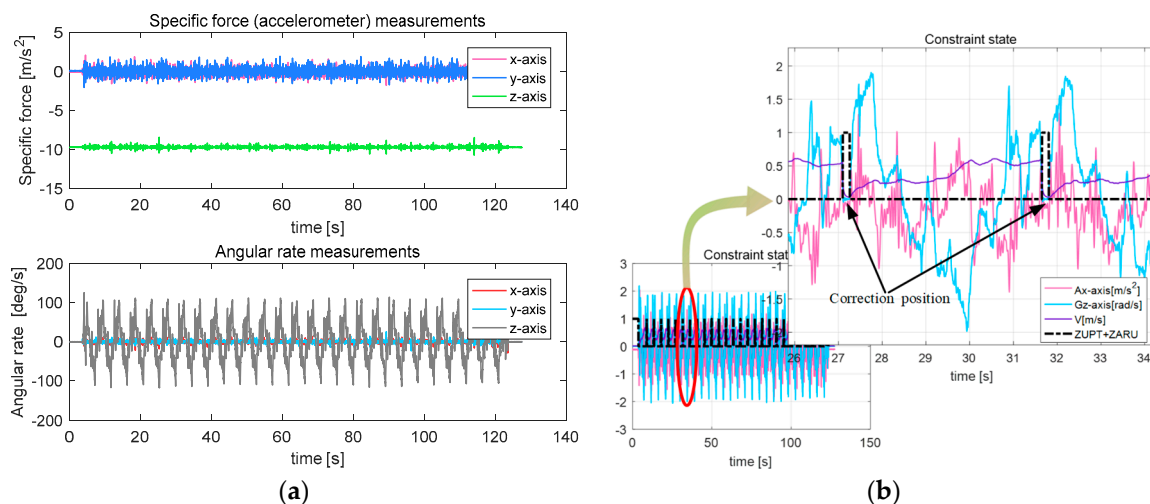


Figure 14. Cont.

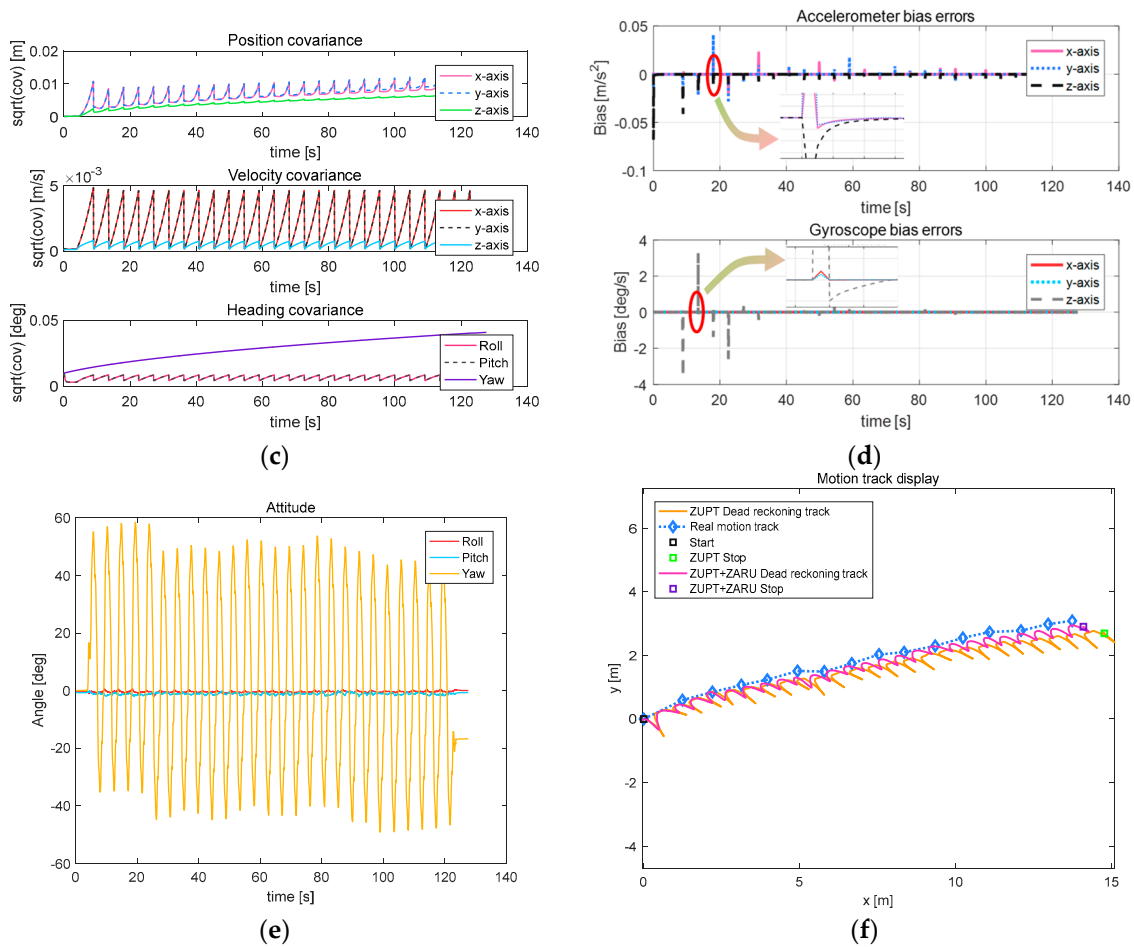


Figure 14. Parameters related to linear motion under velocity and angular velocity constraints. (a) Output result of the MEMS IMU; (b) Constraint state; (c) Result of the covariance; (d) Bias errors; (e) Attitude of the snake robot; (f) Moving path of the snake robot.

The pitch and roll axis attitude angle is found to exhibit basic fluctuations near zero, with the heading axis attitude angle periodically changing based on the robot's serpentine movement, as shown in Figure 14e. The resulting trajectory curve is shown in Figure 14f; the orange curve is the motion curve under the velocity restriction, with a horizontal error of 0.85 m. The red curve is the motion curve under velocity and angular velocity restriction, with a horizontal error of 0.41 m. The horizontal error is 6.07% and 2.92% of TTD, respectively (the results of the other three tests are: 6.83% and 3.65%, 5.94% and 2.95%, and 6.36% and 3.27%). From the above data, the horizontal error precision under velocity and angular velocity restriction is increased by 3% compared to that under velocity restriction only. Because of the length restrictions of this paper, we only consider the situation under velocity and angular velocity restrictions in the next tests.

4.2.2. Turn Motion Test

In the actual use of a snake robot, the turn motion is the most common movement scenario. After the passing the linear motion test, the next test is to set two target points that are on the vertical direction relative to the start point by the host computer to examine the feasibility of the algorithm in larger heading deviation.

In Figure 15a–d, the snake robot is performing the serpentine movement in a fixed period. Even for the large turning radius, the accelerometer data changes little. The cyclicity of constraints, covariance

of position, velocity and attitude, bias error convergence of accelerometer and gyroscope are consistent with those of linear motion.

Figure 15e shows that the robot’s heading attitude angle has changed 90 degrees, and the other two axes are still approximately zero. From the resulting trajectory in Figure 15f, the actual walking distance is 6.5 m, the horizontal error is 0.28 m, 4.31% of TTD (the results of the other three tests are: 4.25%, 4.97%, 3.84%).

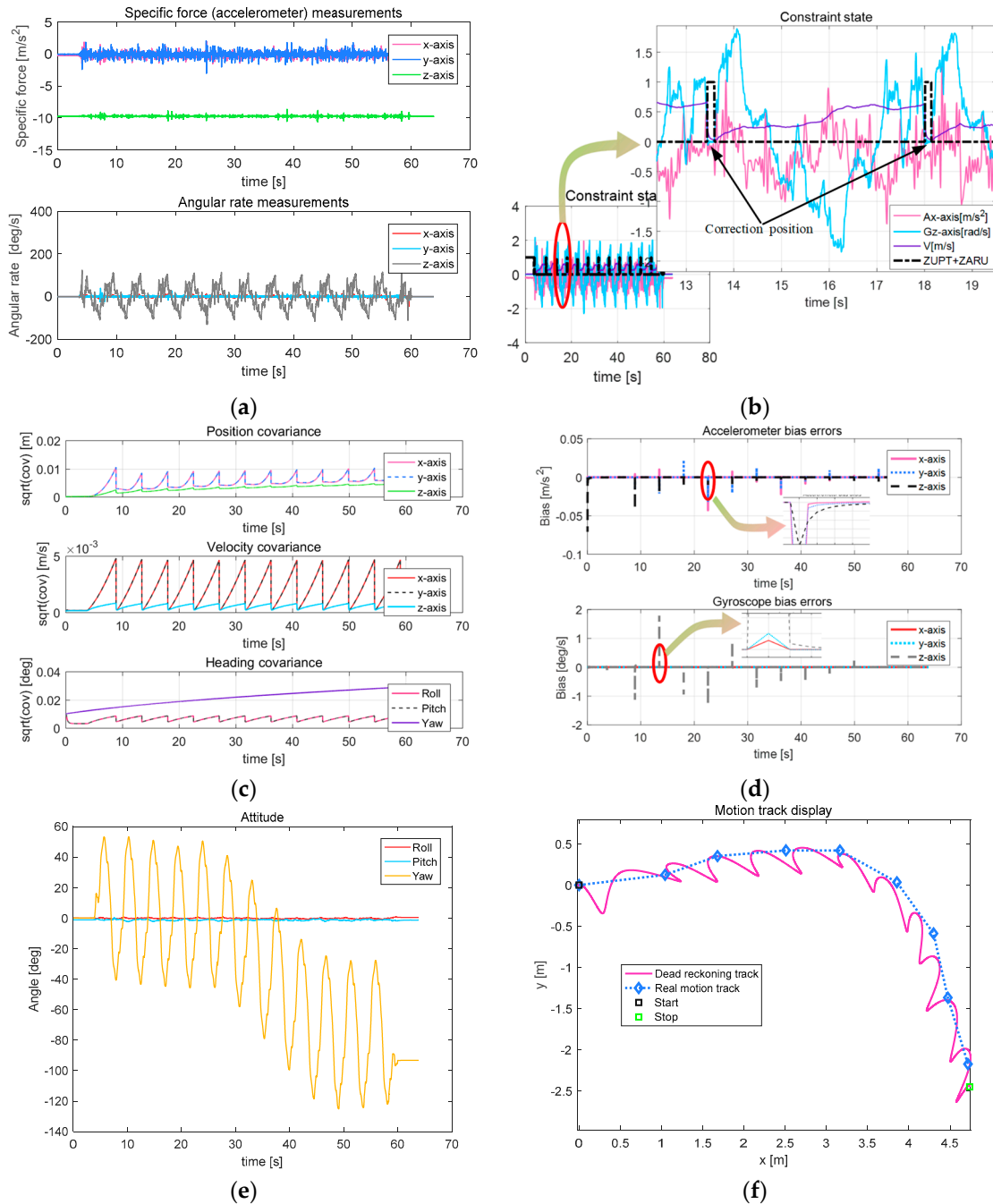


Figure 15. Parameters related to turn motion. (a) Output result of the MEMS IMU; (b) Constraint state; (c) Result of the covariance; (d) Bias errors; (e) Attitude of the snake robot; (f) Moving path of the snake robot.

4.2.3. Turn-Back Motion Test

In addition to the straight line and turn movement, a typical motion of a robot is to return to the origin in the mission process. After passing the tests of the algorithm in a straight line and turning movements, the next test is to reset the two target points by the host computer: the endpoint to go to (15 m away from the start point) and the endpoint to return to. Further investigation on the algorithm’s feasibility for larger heading turn states and for the return path of the snake robot is shown below.

Figure 16a–d shows that, because the turn-back movement obviously has a long range and more serpentine cycles, the convergence of data, including accelerometer and gyroscope data, constraint cyclicity, the covariance of position, velocity and attitude, and bias errors of the accelerometer and gyroscope, are consistent with those of both linear and right-angle turn movements.

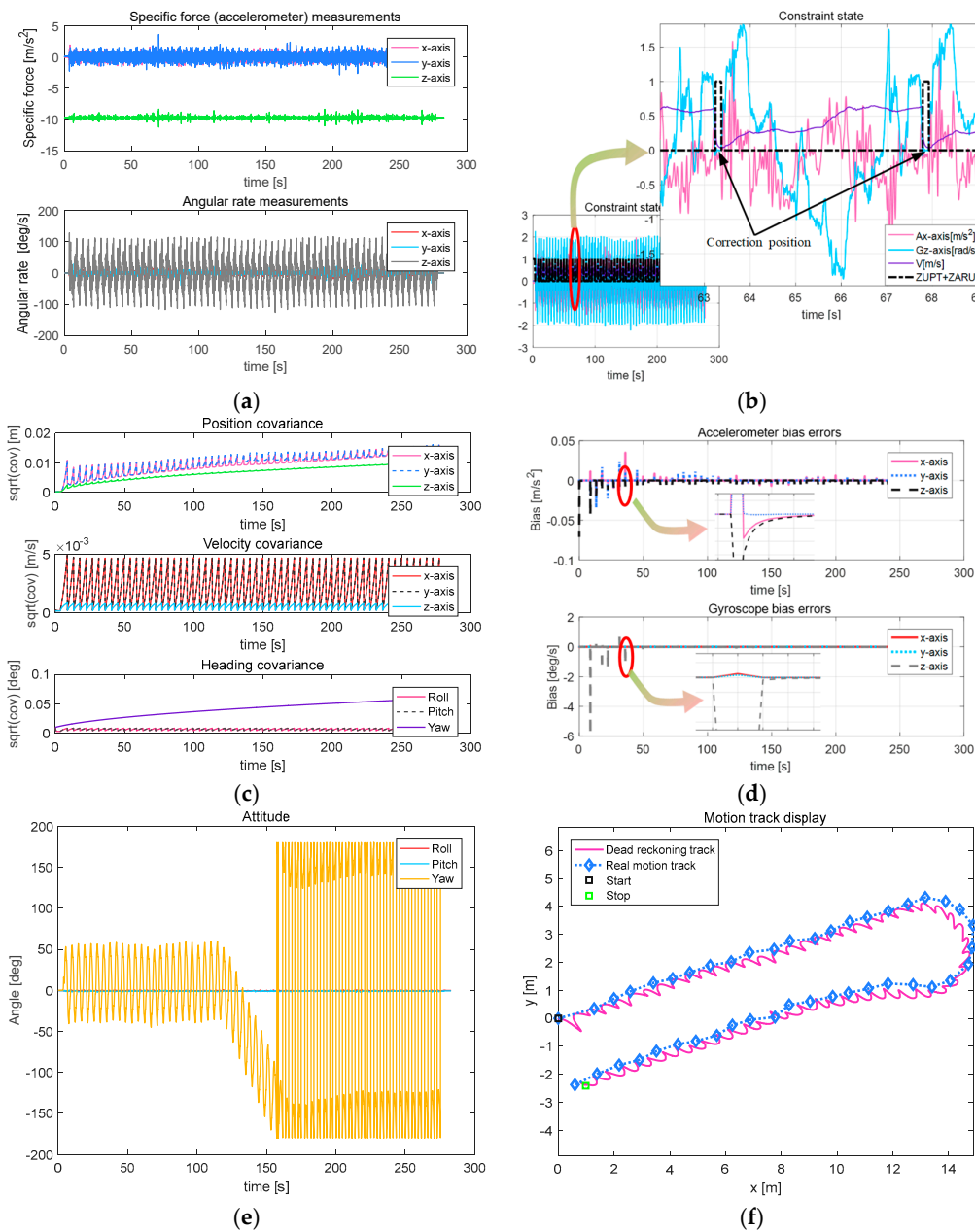


Figure 16. Parameters related to turn-back motion. (a) Output result of the MEMS IMU; (b) Constraint state; (c) Result of the covariance; (d) Bias errors; (e) Attitude of the snake robot; (f) Moving path of the snake robot.

Figure 16e shows that the robot’s heading attitude angle has changed 180 degrees because of the turn-back movement, with the other two axes remaining approximately zero. From the resulting trajectory in Figure 16f, the actual walking distance is 31.6 m, the horizontal error is 0.58 m, 1.84% of TTD (The results of the other three tests are: 2.37%, 3.01%, 2.18%).

4.2.4. Circular Motion Test

After passing the tests of the algorithm in straight line, turn and turn-back movements, the last test is to design a type of circular motion to evaluate the algorithm’s accuracy in returning to origin after many turns and linear motions. Because of the limitations of space, four target points (5 m × 5 m rectangular) are reset by the host computer; the motion curve is shown in Figure 17f.

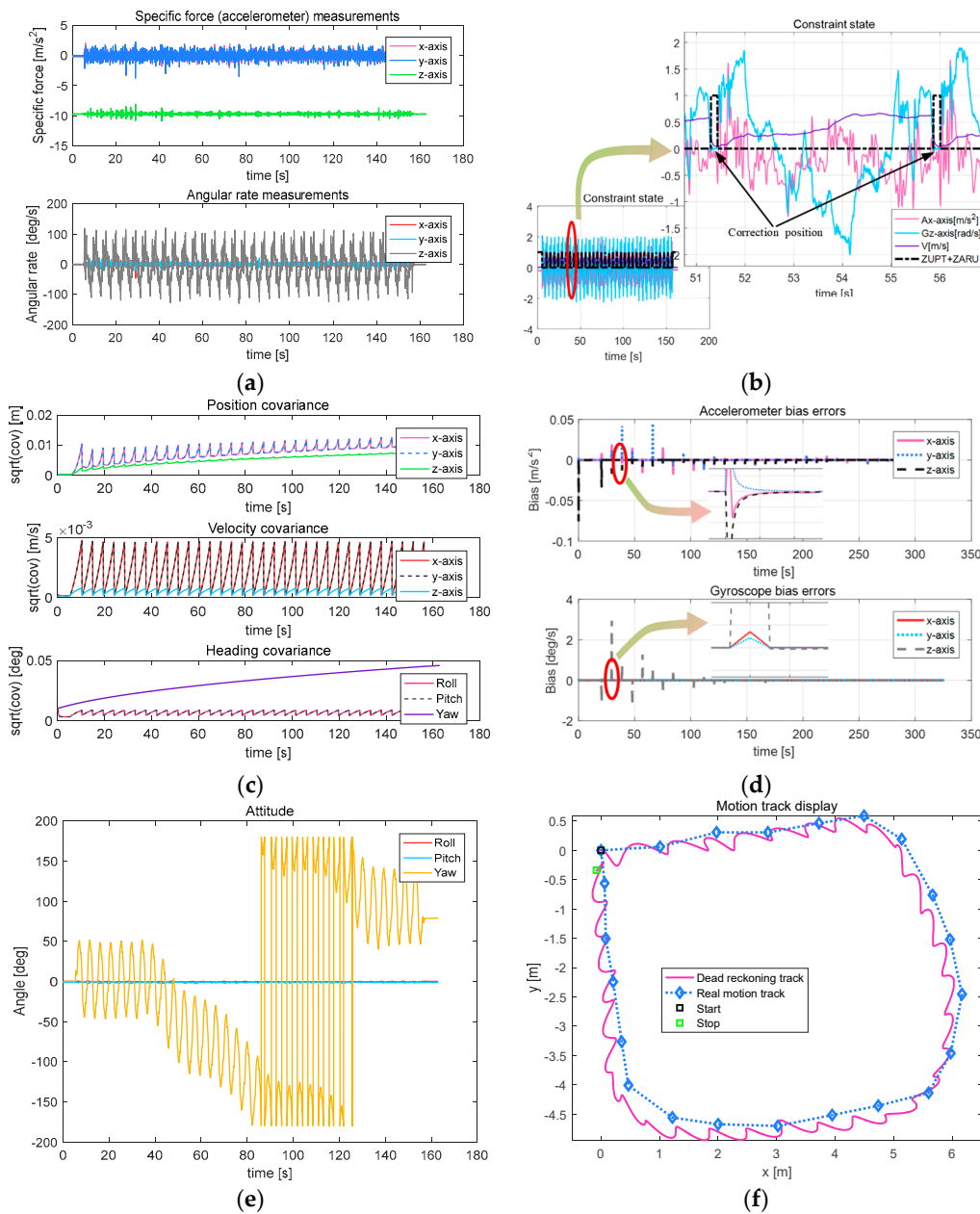


Figure 17. Parameters related to circular motion. (a) Output result of the MEMS IMU; (b) Constraint state; (c) Result of the covariance; (d) Bias errors; (e) Attitude of the snake robot; (f) Moving path of the snake robot.

Figure 17a–d shows that the convergence of data, including accelerometer and gyroscope data, constraint cyclicity, the covariance of position, velocity and attitude, and bias errors of the accelerometer and gyroscope, are consistent with those of linear, turn and turn-back movements.

Figure 17e,f shows that the snake robot's heading axis changed three times in all, basically returning to origin, eventually; however, the heading has changed 90 degrees from the starting moment. The other two axes are still approximately zero. Figure 17b,f shows that the algorithm can effectively and accurately detect the zero-velocity moment and constrain the divergent serious inertial navigation positioning data. It can achieve a trajectory curve that is quite smooth and has highly repetitive start and end points. The simulation trajectory is consistent with the actual trajectory. With respect to the resulting trajectory curve, its horizontal error is 0.35 m, the actual walking distance is 19.4 m, 1.80% of TTD (the results of the other three tests are: 2.52%, 2.67%, 1.75%).

In summary, its location precision is within 5% of TTD in regular motion types using the proposed navigation method based on the characteristics of the snake robot motion constraints. This is able to satisfy the requirement of autonomous navigation and positioning for the snake robot in traditional applications in short distance running situations.

5. Conclusions

This paper described a proposed autonomous navigation method based on the characteristics of a snake robot's motion constraints. The proposed method realized the snake robot's autonomous navigation and positioning with non-nodes and non-assistant using the installation of IMU on the robot. By studying the motion characteristics of the snake robot, it establishes its kinematics model, analyzes the movement constraint characteristics and movement error propagation characteristics, explores the snake robot navigation arrangement, proposes constraint criteria and fixed relationships, and satisfies the zero-state constraint with the motion feature and control modes of the snake robot. Finally, it studies the snake robot's EKF position estimation method under motion characteristics restriction, realizing the robot's autonomous navigation positioning. The tests of linear, turn, turn-back, and circular motion with the self-developed snake robot show that its comprehensive location accuracy was less than 5% of TDD, and proved that in short run situations, the method can meet the requirements of autonomous navigation and positioning for snake robots.

Although this method has been applied to the field of vehicle and pedestrian navigation for a long time, it has not been applied to snake robots because of the uniqueness of the movement characteristics of snake robots. The novelty of this paper is in applying this technology to the snake robot's navigation by studying the motion model of the snake robot and to study the feasibility of its short-distance navigation. The future work is to continue improving the key technology of long-distance long-time autonomous navigation, and to apply this work to the robot SLAM system to improve the accuracy of positioning and mapping. Finally, a snake robot with high-precision navigation can be widely used in disaster rescue, underwater survey, industrial testing and other special environments that traditional robots or humans cannot enter.

Acknowledgments: This work is supported by the National Natural Science Foundation of China (Grant No. 61771059); the Beijing Natural Science Foundation (Grant No. 4172022); and the Beijing Science and Technology Project (Grant No. Z161100005016109). Supported by Beijing Key Laboratory of High Dynamic Navigation Technology.

Author Contributions: Lihua Dou and Xu Zhao conceived and designed this work. Xu Zhao designed and implemented the acquisition system, analyzed the data and drafted the work. Zhong Su and Ning Liu performed the experiments and helped with data analysis. Lihua Dou, Zhong Su and Ning Liu revised the work. All the authors have approved the submitted version of the manuscript, have agreed to be personally accountable for their own contributions in the literature.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hirose, S. *Biologically Inspired Robots: Snake Locomotors and Manipulators*; Oxford University Press: Oxford, UK, 1993.
2. Liljebäck, P.; Pettersen, K.Y.; Stavadahl, Ø.; Gravdahl, J.T. A review on modelling, implementation, and control of snake robots. *Robot. Auton. Syst.* **2012**, *60*, 29–40. [[CrossRef](#)]
3. Sanfilippo, F.; Azpiazu, J.; Marafioti, G.; Transeth, A.A.; Stavadahl, Ø.; Liljebäck, P. Perception-Driven Obstacle-Aided Locomotion for Snake Robots: The State of the Art, Challenges and Possibilities. *Appl. Sci.* **2017**, *7*, 336. [[CrossRef](#)]
4. Gray, J. The Mechanism of Locomotion in Snakes. *J. Exp. Biol.* **1946**, *23*, 101–124. [[PubMed](#)]
5. Liljebäck, P.; Pettersen, K.; Stavadahl, Ø. Modelling and Control of Obstacle-Aided Snake Robot Locomotion Based on Jam Resolution. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 3807–3814.
6. Liljebäck, P.; Pettersen, K.Y.; Stavadahl, Ø.; Gravdahl, J.T. Compliant control of the body shape of snake robots. In Proceedings of the 2004 IEEE International Conference on Robotics and Automation, Hong Kong, China, 31 May–7 June 2004; pp. 4548–4555.
7. Yamada, H.; Chigisaki, S.; Mori, M.; Takita, K.; Ogami, K.; Hirose, S. Development of amphibious snake-like robot ACM-R5. In Proceedings of the 36th International Symposium on Robots, Tokyo, Japan, 29 November–1 December 2005.
8. Wright, C.; Buchan, A.; Brown, B.; Geist, J.; Schwerin, M.; Rollinson, D.; Tesch, M.; Choset, H. Design and architecture of the unified modular snake robot. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, St. Paul, MN, USA, 14–18 May 2012; pp. 4347–4354.
9. Liljebäck, P.; Stavadahl, Ø.; Beitnes, A. Snake Fighter-Development of a Water Hydraulic Fire Fighting Snake Robot. In Proceedings of the 9th International Conference on Control, Automation, Robotics and Vision, Singapore, 5–8 December 2006; pp. 1–6.
10. Liljebäck, P.; Pettersen, K.; Stavadahl, Ø. A Snake Robot with a Contact Force Measurement System for Obstacle-aided Locomotion. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 683–690.
11. Tanaka, M.; Kon, K.; Tanaka, K. Range-Sensor-Based Semiautonomous Whole-Body Collision Avoidance of a Snake Robot. *IEEE Trans. Control Syst. Technol.* **2015**, *23*, 1927–1934. [[CrossRef](#)]
12. Tian, Y.; Gomez, V.; Ma, S. Influence of two SLAM algorithms using serpentine locomotion in a featureless environment. In Proceedings of the IEEE International Conference on Robotics and Biomimetics, Qingdao, China, 3–7 December 2016; pp. 182–187.
13. Chavan, P.; Murugan, M.; Unnikannan, E.V.; Singh, A.; Phadatar, P. Modular Snake Robot with Mapping and Navigation: Urban Search and Rescue (USAR) Robot. In Proceedings of the 2015 International Conference on Computing Communication Control and Automation, Pune, India, 26–27 February 2015; pp. 537–541.
14. Fu, X.; Wang, Y. Research of snake-like robot control system based on visual tracking. In Proceedings of the International Conference on Electronics, Communications and Control, Ningbo, China, 9–11 September 2011; pp. 399–402.
15. Xiao, X.; Cappel, E.; Zhen, W.; Dai, J.; Sun, K.; Gong, C.; Travers, M.J.; Choset, H. Locomotive reduction for snake robots. In Proceedings of the IEEE International Conference on Robotics and Automation, Seattle, WA, USA, 26–30 May 2015; pp. 3735–3740.
16. Ponte, H.; Queenan, M.; Gong, C.; Mertz, C.; Travers, M.; Enner, F.; Hebert, M.; Choset, H. Visual sensing for developing autonomous behavior in snake robots. In Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, China, 31 May–7 June 2014; pp. 2779–2784.
17. Ohno, K.; Nomura, T.; Tadokoro, S. Real-Time Robot Trajectory Estimation and 3D Map Construction using 3D Camera. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 5279–5285.
18. Billah, M.M.; Khan, M.R. Smart inertial sensor-based navigation system for flexible snake robot. In Proceedings of the IEEE International Conference on Smart Instrumentation, Measurement and Applications, Kuala Lumpur, Malaysia, 24–25 November 2014; pp. 1–5.

19. Estébanez, J.G. *GPS-IMU Integration for a Snake Robot with Active Wheels*; Institutt for Teknisk Kybernetikk: Trondheim, Norway, 2009.
20. Yang, W.; Bajenov, A.; Shen, Y. Improving low-cost inertial-measurement-unit (IMU)-based motion tracking accuracy for a biomorphic hyper-redundant snake robot. *Robot. Biomim.* **2017**, *4*, 16. [[CrossRef](#)] [[PubMed](#)]
21. Baglietto, M.; Sgorbissa, A.; Verda, D.; Zaccaria, R. Human navigation and mapping with a 6DOF IMU and a laser scanner. *Robot. Auton. Syst.* **2011**, *59*, 1060–1069. [[CrossRef](#)]
22. Sato, M.; Fukaya, M.; Iwasaki, T. Serpentine locomotion with robotic snakes. *IEEE Control Syst.* **2002**, *22*, 64–81. [[CrossRef](#)]
23. Wiriyacharoensunthorn, P.; Laowattana, S. Analysis and design of a multi-link mobile robot (Serpentine). In Proceedings of the 2002 IEEE International Conference on Industrial Technology, Bangkok, Thailand, 11–14 December 2002; Volume 2, pp. 694–699.
24. Kelasidi, E.; Tzes, A. Serpentine motion control of snake robots for curvature and heading based trajectory-parameterization. In Proceedings of the 2012 20th Mediterranean Conference on Control & Automation, Barcelona, Spain, 3–6 July 2012; pp. 536–541.
25. Khatib, O. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *Int. J. Robot. Res.* **1986**, *5*, 90–98. [[CrossRef](#)]
26. Ye, C.; Hu, D.; Ma, S.; Li, H. Motion planning of a snake-like robot based on artificial potential method. In Proceedings of the 2010 IEEE International Conference on Robotics and Biomimetics, Tianjin, China, 14–18 December 2010; pp. 1496–1501.
27. Groves, P.D. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*; Artech House: Norwood, MA, USA, 2008.
28. Jahn, J.; Batzer, U.; Seitz, J.; Patino-Studencka, L.; Boronat, J.G. Comparison and evaluation of acceleration based step length estimators for handheld devices. In Proceedings of the 2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Zurich, Switzerland, 15–17 September 2010.
29. Retscher, G.; Fu, Q. An intelligent personal navigator integrating GNSS, RFID and INS for continuous position determination. *Boletim de Ciências Geodésicas* **2009**, *15*, 707–724.



© 2018 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) license (<http://creativecommons.org/licenses/by/4.0/>).