Analysis of Position and State Estimation of Quadruped Robot Dog Based on Invariant Extended Kalman Filter

Haiyan Shao[1,*], Qingshuai Zhao[1], Bin Chen[2], Xiankun Liu[3] and Zhiquan Feng[4]

[1]Haiyan Shao and Qingshuai Zhao are with the School of Mechanical Engineering, University of Jinan, Jinan, Shandong, 250022, China

[2]Bin Chen is with the Shandong Youbaote Intelligent Robotics Co., Ltd of Jinan, Jinan, Shandong, 250098, China

[3]Xiankun Liu is with the Jinneng Science & Technology Co., Ltd of Qihe, Dezhou, Shandong, 251100, China

[4]Zhiquan Feng is with the School of Information Science and Engineering, University of Jinan, Jinan, Shandong, 250022, China

Abstract: Compared with the state estimation of quadruped robots based on external sensors such as camera and lidar, the state estimation based on body sensors can provide high-frequency and stable odometer estimation. By analyzing the state estimation methods of the legged robot based on the body sensor, the invariant extended Kalman filter (IEKF) based on the body sensor is determined to conduct the state estimation analysis of the quadruped robot. Through various path tracking experiments in simulation and real environment, the influence of travel speed, travel distance and different steering angles on the position state estimation results was analyzed, and the IEKF model was optimized by compensating the angular velocity. Experiments show that within the set speed range, after adding angular velocity compensation, the position estimation accuracy error of the robot dog is well controlled and is less than 1%.

Keywords: Position state estimation, Invariant extended Kalman filter, Quadruped robot dog, Angular velocity compensation.

1. STATE ESTIMATION METHOD FOR QUADRUPED ROBOT

Invariant Extended Kalman Filter [1] defines the robot state on the Lie Group. If the dynamics satisfy the specific "affine group" characteristic, the symmetry will make the estimation error satisfy the "logarithmic linear" autonomous differential equation on the Lie Algebra. The linear system can be used to accurately recover the estimated state of the nonlinear system when it evolves over the group [2]. To sum up, the theoretical basis of the design of the invariant observer is that the estimation error is invariant under the action of the matrix Lie Group, and the error transfer matrix is independent of the state estimation value. The IEKF algorithm effectively solves the filter divergence problem of the EKF (Extended Kalman Filter, EKF) algorithm when noise is introduced and the inconsistency between observation and state estimation [3]. It is an international frontier method for state estimation of legged robots.

In 2013, M. Reinstein and M. Hoffmann [4] statistically trained a regression function on angular information and robot stride (relative position increment within one gait cycle) and produced a legged odometer. The data fusion of leg odometer and IMU is realized by using EKF. The method can handle the sliding of the foot well, and the position error is between 1% and 1.5% of the total path length. The disadvantage is that retraining is required when changing the test scene.

The StarlETH robot [5] designed by ETH Zurich has truly realized the fusion algorithm of inertial measurement unit and joint encoder in the experiment, which is used to predict the motion state of the robot. Bloesch M *et al*. [6] achieved state estimation for unknown terrain and arbitrary motion gaits by utilizing EKF framework, fusing joint encoder data and airborne IMU measurements, and incorporating all foothold positions into the estimation process. In the static walking small disturbance experiment, the root mean square errors (RMSE) of the velocity estimates in the three-axis directions are: 0.0111m/s, 0.0153m/s and 0.0126m/s, respectively. Unscented Kalman Filter (UKF) [7] was used in dynamic motion experiments. The UKF algorithm was mounted on the StarlETH robot to perform dynamic motion experiments on uneven terrain. The RMSE of the velocity estimates in the three-axis directions were: 0.0546m/s, 0.0406m/s and 0.0348m/s. Due to the different dimensions of the state vector selected by UKF, the number of sigma points required is also different. When calculating the Kalman

[*] Address correspondence to this author at the Haiyan Shao and Qingshuai Zhao are with the School of Mechanical Engineering, University of Jinan, Jinan, Shandong, 250022, China; Tel: 86-15215310628;
E-mail: me_shaohy@ujn.edu.cn

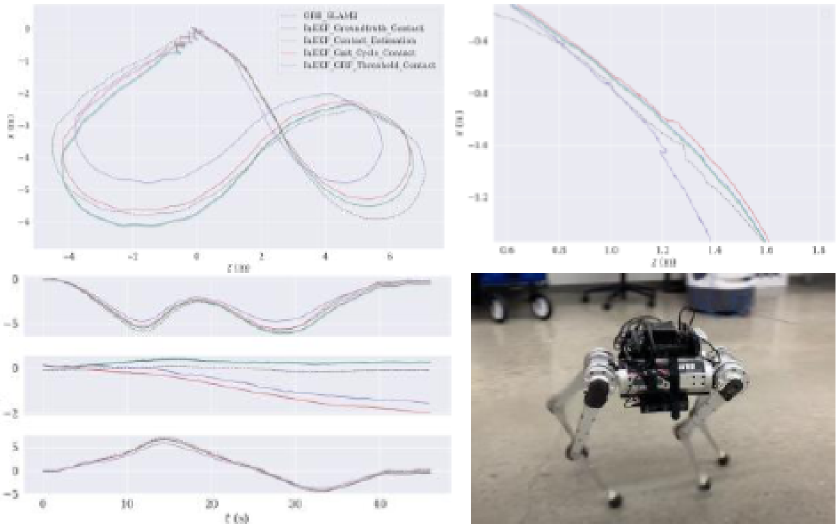


Figure 1: Experiment on walking path estimation of quadruped robot dog.

gain at the end, it is difficult to solve the high-dimensional matrix.

In 2016, A. Barrau and S. Bonnabel [8] introduced the IEKF based on Lie Group and verified that IEKF has better local convergence than EKF in experiments. Under challenging terrain, the divergence of EKF can be effectively avoided. In 2019, Hartley R *et al*. [9] applied IEKF to the bipedal robot CASSISE for the first time. In the free walking experiment on a 15M path, the position biases were less than 5%. Lin T-Y *et al*. [10] used IEKF and deep learning-based contact data for state estimation of a quadruped robot, as shown in Figure **1**, the generated odometry trajectories were comparable to ORB-SLAM2.

The quadruped robot dog has attracted widespread attention because of its good traffic capacity, and accurate positioning is one of the key prerequisites for its walking and navigation on complex roads. This paper mainly analyzes the position and state estimation algorithm of quadruped robot from four parts. The first part is the research status of the state estimation method of the legged robot, the second part mainly expounds the principle of the invariant extended Kalman filter and its improvement and optimization, the third part is the experiment, and the last part is the conclusion.

2. PRINCIPLE OF INVARIANT EXTENDED KALMAN FILTER AND IMPROVEMENT AND OPTIMIZATION

2.1. Filter Model Optimization and Design

In the framework of EKF, to estimate the posture direction of the robot body (IMU) in the world coordinate system at t time, the quaterniont q_t is generally selected as the state quantity. In the framework of IEKF, different from EKF, the corresponding rotation matrix is selected as the state quantity, that is, $R_t \in SO(3)$ is assumed. Similarly, in IEKF, the speed and position state quantities of the robot are w_t^d and $p_t \in \mathbb{R}^3$. The set of the above state variables forms a matrix Lie Group $X_t \in SE_{N+2}$,Where N stands for contact point.

$$X_t \triangleq \begin{pmatrix} R_{WB}(t) & v_B^W(t) & P_{WB}^W(t) & P_{WC_1}^W(t) & \cdots & P_{WC_N}^W(t) \\ 0_{1,3} & 1 & 0 & 0 & \cdots & 0 \\ 0_{1,3} & 0 & 1 & 0 & \cdots & 0 \\ 0_{1,3} & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0_{1,3} & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \tag{1}$$

Where $R_{WB}(t)$ is the attitude matrix of the robot dog, $v_B^W(t)$ is the velocity vector of the robot dog in the world coordinate system, $P_{WB}^W(t)$ is the position vector in the world coordinate system, and $P_{WC_N}^W(t)$ is the position vector of the contact point of the foot end in the world coordinate system. Because the contact process and measurement model of each contact point of the robot in Trot gait are the same and without loss of generality, the state matrix can be simplified as $X_t = \begin{pmatrix} R_t & v_t & p_t & d_t \end{pmatrix}$, where d_t represents the position vector of the foot contact point in the simplified world coordinate system.

References [9] and [10] consider the interference of noise on the IMU data, but do not consider the Yaw angular velocity drift error of the IMU rotating around the Z axis. The IMU's Roll and Pitch angles can be corrected using gravity as a reference, and the Yaw angle's reference is geomagnetism. Because the internal magnetic field of the robot is complex and the outside is wrapped by an iron shell, which seriously affects the accuracy of the geomagnetic information received by the IMU, it is necessary to perform angular velocity compensation on the measurement information of the IMU in the Yaw direction, to offset the Yaw angular velocity drift of the IMU.

Assuming that the measured value of the IMU is affected by Gaussian white noise, and the Yaw angle drift error of the IMU caused by factors such as magnetic field is w_t^{d} (*i.e.*, angular velocity compensation). The value is always greater than or less than 0 (related to the direction of the magnetic field), which is different from Gaussian noise and more like the measurement error of numerical stability, therefore:

$$\tilde{w}_t = w_t + w_t^g + w_t^{\mathrm{d}},\ w_t^g \sim N(0_{3,1}, \sum\nolimits^g \delta(t-t')) \tag{2}$$

$$\tilde{a}_t = a_t + w_t^a,\ w_t^a \sim N(0_{3,1}, \sum\nolimits^a \delta(t-t')) \tag{3}$$

Where N represents Gaussian process and $\delta(t-t')$ represents δ function, $\tilde{w}_t$ and $\tilde{a}_t$ represent the measured values of angular velocity and linear acceleration, w_t and a_t represent the actual angular velocity and linear acceleration of the robot dog, w_t^g and w_t^{a} represent the Gaussian white noise with a mean value of 0, w_t^{d} represents the yaw angle drift error caused by factors such as magnetic fields, and its value is assumed to be 0 in the Roll and Pitch directions.

It is assumed that the position of the contact point between the foot end of the robot dog and the ground remains fixed in the world coordinate system, that is, the measured speed of the contact point of the foot end is zero. To accommodate the potential small sliding of the foot end, the measured speed is assumed to be the actual speed plus Gaussian white noise, that is:

$$\tilde{v}_C^W = v_C^C + w_t^v,\ w_t^v \sim N(0_{3,1}, \sum\nolimits^v \delta(t-t')) \tag{4}$$

Where $\tilde{v}_C^W$ represents the measured value of the speed of the contact point of the foot end v_C^C represents the actual speed of the contact point between the foot end of the robot dog and the ground, and w_t^v represents the Gaussian white noise with a mean value of 0.

Using the inertial measurement unit and foot contact measurement, the system dynamics model can be expressed as [10].

$$\frac{d}{dt}R_t = R_t\left(\tilde{w}_t - w_t^g \text{-} w_t^{\mathrm{d}}\right)_\times \tag{5}$$

$$\frac{\mathrm{d}}{dt}v_t = R_t(\tilde{a}_t - w_t^a) + g \tag{6}$$

$$\frac{d}{dt}p_t = v_t \tag{7}$$

$$\frac{\mathrm{d}}{dt}d_t = R_t h_R\left(\tilde{\alpha}_t\right)\left(-w_t^v\right) \tag{8}$$

Where $(\bullet)_\times$ means 3 × 3 skew symmetry matrix, g represents the gravity acceleration vector, $\tilde{\alpha}_t$ represents the measured value of the encoder, $h_R\left(\tilde{\alpha}_t\right)$ is the measurement direction of the contact point coordinate system (measured by the encoder and calculated by forward kinematics) relative to the IMU coordinate system, and $R_t h_R\left(\tilde{\alpha}_t\right)$ is the rotation matrix that transforms the vector from the foot contact coordinate system to the world coordinate system.

The speed and position information of the robot dog obtained by the state estimation algorithm can meet the basic motion control requirements, but if you want the motion control of the robot to be more stable and accurate, or to be used for SLAM mapping and navigation [11], then the speed and position calibration is necessary. If the actual running speed of the robot dog is v, and the estimated speed obtained by the state estimation algorithm is v_t, a scale factor A is required to satisfy (9). The same applies to position calibration.

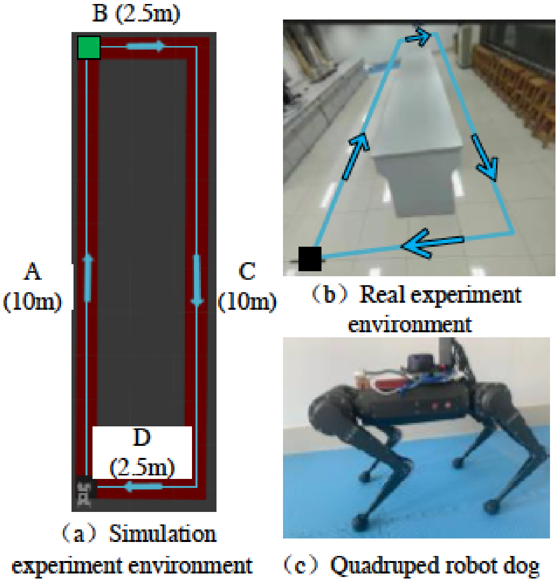
$$v = A \times v_t \tag{9}$$

$$\mathrm{p} = A \times p_t \tag{10}$$

Where, A is the scale coefficient, which is a fixed constant value.

2.2. Simulation Experiment Verification

To verify the effectiveness and accuracy of the improved state estimation algorithm, an experimental environment is set up (Figure **2**). In the Gazebo simulation environment, control the quadruped robot dog to move in a Trot gait, with a walking speed of 0-0.2m/s, and record its real trajectory and estimated trajectory. In Figure **2**, the black square in section A is the starting point. The robot dog walks clockwise along the track line and returns to the end position of section A (coinciding with the starting point) to end.



(a) Simulation experiment environment (b) Real experiment environment (c) Quadruped robot dog

Figure 2: Experiment environment and quadruped robot dog.

Figure **3** shows the estimated trajectory and walking trajectory of the robot dog in the Gazebo simulation environment. The blue line indicates the estimated trajectory of the robot dog based on the IEKF algorithm, and the red line indicates that the walking trajectory of the robot dog which is also the reference trajectory. After the simulation, the data package is imported into EVO, and the average position bias is 3.2cm, and the RMSE is 3.7cm. The maximum cumulative bias of the trajectory appears at the end position, which is 7.9cm. The simulation experiment preliminarily proves the feasibility of the research on the state estimation of the quadruped robot dog based on IEKF.

3. EXPERIMENT

The main purpose of the experiment is to compare the IEKF algorithm with the EKF algorithm, as well as the two algorithms before and after adding angular velocity compensation, to verify the feasibility and reliability of the improved algorithm based on IEKF. Specifically, the remote-control handle controls the robot dog to reach the preset end position according to the preset trajectory, and then analyzes the accuracy of the position estimation of the robot dog by comparing the estimated trajectory and the set trajectory. In the experiment, the maximum moving speed was set to 0.2m/s and 0.3m/s. The experimental environment and quadruped robot dog are shown in Figure **2**. The size of the experimental environment is the same as that of the simulation environment. The total length of the path is 25m, of which the lengths of sections A and C are 10m, and the lengths of sections B and D are 2.5m. The position of the black square where the robot dog in segment A is located is the starting point, and the end point is set according to different experiments. The robot dog is equipped with a 3DM-GX3®-25 IMU (Table **1**).

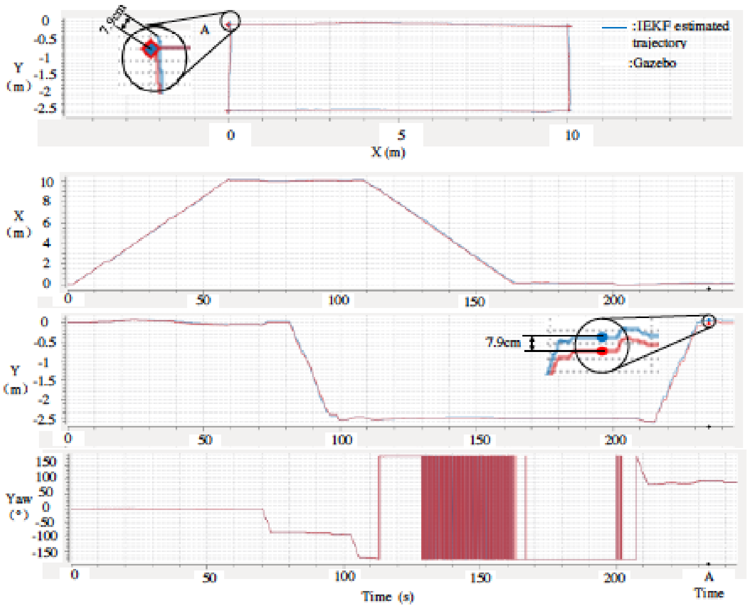


Figure 3: The estimated trajectory and walking trajectory of the robot dog in the simulation environment.

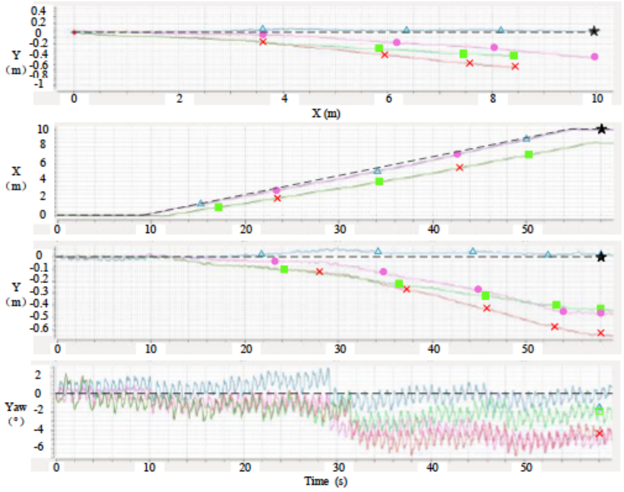
Table 1: 3DM-GX3® -25 IMU Performance Parameters

Physical Dimensions and Electrical Characteristics				
Input voltage	**Interface**	**Dimensions**	**Weight**	**Power Consumption**
4~36V DC	USB2.0/ RS232	44*24* 11mm	18g	400mW
Performance				
Roll/Pitch/Heading Accuracy	**Attitude resolution**	**Attitude Repeatability**	**Filter output frequency**	**Vibration limit**
±2°	＜0.01°	0.2°	1-1000Hz	500g

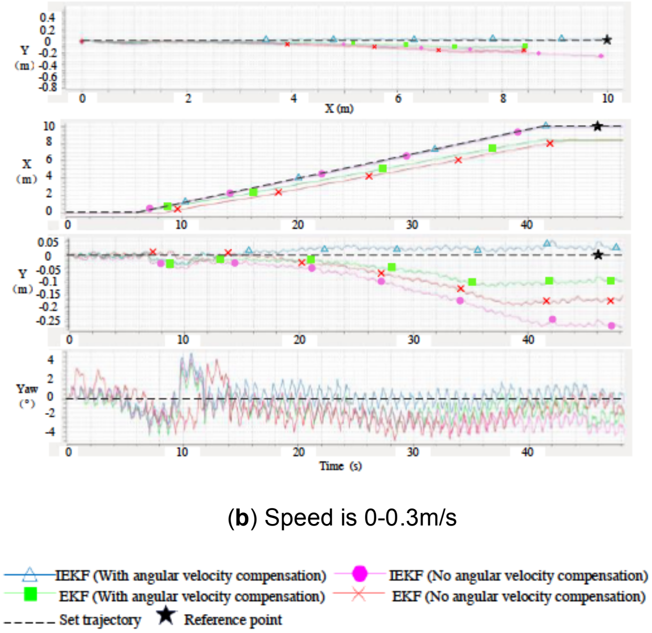
There are three types of experiments: (1) Straight-through experiments. Measure the error between the estimated end position and the preset end position, and obtain the position estimation accuracy of the robot dog on a single axis. (2) Experiment around the rectangular field one time. Measure the error between the estimated end position and the starting position, and indirectly obtain the estimated accuracy of the angular velocity. (3) Experiment around the rectangular field three times. The error between the estimated end position and the starting position is measured to further verify the reliability of the algorithm. For these experiments, if the position estimation error is less than 1%, the algorithm is reliable, otherwise it is unreliable.

3.1. Straight-Through Experiment

Set the starting and ending points of section A, the black square in Figure **2** is the starting point, the green square at the other end of section A is the end point, and the entire distance is 10m. At the beginning of the experiment, the two front feet of the quadruped robot dog stepped on the starting position. After starting, the robot dog moves forward at two speeds of 0-0.2m/s and 0-0.3m/s respectively. Stop when both front feet are at the end.



(**a**) Speed is 0-0.2m/s



(**b**) Speed is 0-0.3m/s

Figure 4: The experiment of the robot dog moving forward 10m in a straight line at different speeds.

The results of the straight-through experiment are shown in Figure **4**. The set trajectory is represented by a black dotted line, which is a straight line along the X-axis for 10m and the steering angle is 0. Different position state estimation algorithms are represented by solid lines with different colors and with different markers. In Figure **4**, from top to bottom, the displacements of different position state estimation algorithms on the X-Y plane, the displacement components on the X axis, the displacement components on the Y axis, and the yaw angle of the trajectory of different position state estimation algorithms are listed. Based on the set trajectory and reference point, the displacement components of different algorithms in the X-axis are counted, and the position estimation error with the reference point of the X-axis component is calculated.

A total of 11 groups of experiments were carried out in this part, and the results are shown in Table **2**. The

Table 2: X-axis Position Estimation Error Summary Table when Traveling in a Straight Line (cm)

Groups Parameters		1	2	3	4	5	6	7	8	9	10	11	RMSE
EKF (No angular velocity compensation)	0-0.2m/s	-159.3	-172.3	-150.2	-132.6	-145.6	-142.6	-168.1	-145.5	-162.0	-168.8	-170.3	156.6
	0-0.3m/s	-160.7	-178.9	-163.5	-155.3	-167.8	-152.9	-179.9	-154.6	-177.1	-170.5	-166.8	166.4
EKF (With angular velocity compensation)	0-0.2m/s	-158.0	-172.3	-150.0	-132.2	-146.6	-142.3	-167.5	-145.1	-161.6	-169.0	-170.5	156.4
	0-0.3m/s	-160.1	-178.9	-163.1	-156.2	-167.0	-152.5	-177.9	-153.1	-176.5	-170.0	-166.3	165.9
IEKF (No angular velocity compensation)	0-0.2m/s	-6.6	-13.8	-11.2	-9.1	-11.8	-9.6	-10.3	-12.5	-11.7	-10.6	-13.2	11.3
	0-0.3m/s	-13.8	-12.3	-10.8	-9.6	-14.2	-11.8	-11.1	-12.3	-12.2	-13.2	-13.5	12.3
IEKF (With angular velocity compensation)	0-0.2m/s	1.1	1.6	1.1	0.2	-2.1	1.7	-1.5	2.2	1.1	-1.6	2.8	1.7
	0-0.3m/s	1.1	2.5	1.2	0.8	2.3	-1.3	2.1	2.2	1.7	2.0	2.1	1.8

following conclusions are obtained through experiments: the error of the IEKF algorithm is much smaller than that of the EKF algorithm, and the errors of the two algorithms after adding angular velocity compensation are smaller than those before adding. The accuracy error of the IEKF algorithm with angular velocity compensation is less than 1%. The IEKF algorithm obtains a higher accuracy than the EKF algorithm because the speed and position are calibrated at the time of design. The RMSE of the improved IEKF algorithm with angular velocity compensation is reduced by 84.9% and 85.4%, respectively, compared with that without angular velocity compensation.

In the straight-line traveling experiment, the estimated position of the robot dog does not coincide with the estimated traveling trajectory, that is, there is a bias in the Y-axis. This is evident in both algorithms without angular velocity compensation. The bias is mainly caused by the following two points: (1) the initial angle of the robot dog does not coincide with the estimated trajectory. (2) There is a cumulative drift error in the yaw angle of the IMU. Since the initial angle biases must exist, the error value of each experiment has obvious and irregular changes, as shown in Table **2**. In this paper, the cumulative drift error of the IMU yaw angle is optimized by adding angular velocity compensation. The specific operation steps are through multiple in-situ rotation experiments, count the difference between the actual rotation angle and the estimated angle of the robot dog, and then obtain the compensation angular velocity w_t^d through inverse integral conversion, and finally compensate it into the IMU data. The angular velocity compensation value of the IMU sensor used in this experiment is $1.05 \times 10^{-3}\text{rad/s}$. The accuracy and anti-electromagnetic interference ability of different inertial sensors are different, and the specific compensation value needs to be measured by your self. To further verify the reliability of the angular velocity compensation algorithm, the following motion experiments around the rectangular field are carried out.

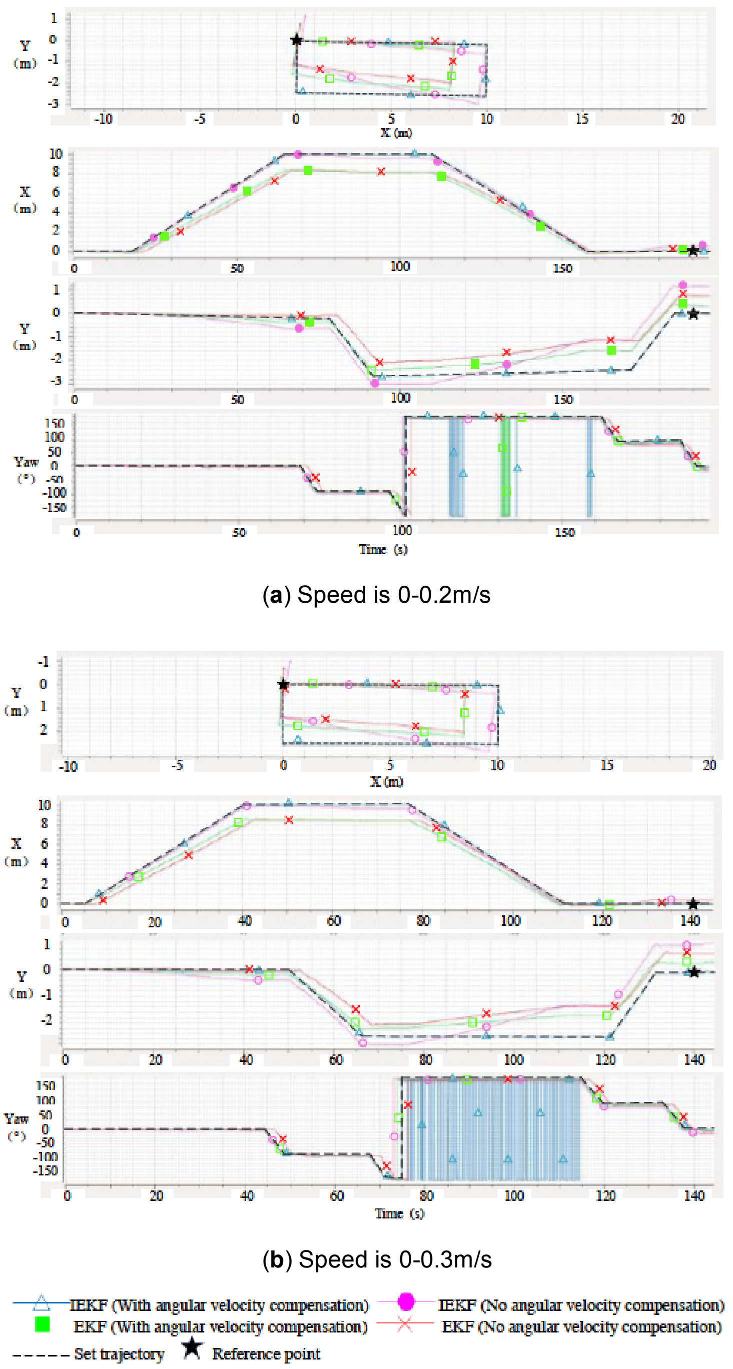
3.2. Experiment Around the Rectangular Field One Time

The main purpose of this experiment is to study the estimation accuracy of different algorithms in the Yaw angle, and the feasibility and reliability of the angular velocity compensation algorithm.

Different from the straight-through experiment, the starting point and the ending point of this experiment coincide, and the robot dog starts from the starting point and returns to the starting point after a circle. When moving around A-D, the ideal trajectory should be a regular rectangle.

Based on the set trajectory and the reference point, the estimated end positions of different state estimation algorithms in the X-Y plane are counted, and the position estimation error with the reference point is calculated. When the speed is 0-0.2m/s and 0-0.3m/s, the experimental results of a circle around the rectangle are shown in Figure **5**.

A total of 11 groups of experiments were carried out in this part, and the results are shown in Table **3**. The following conclusions are obtained through experiments: the position estimation result of the robot dog has obvious right-turn offset when no angular velocity compensation is added. The error of the IEKF algorithm is smaller than that of the EKF algorithm. After adding the angular velocity compensation, the error is much smaller than that before the addition, and



(**a**) Speed is 0-0.2m/s

(**b**) Speed is 0-0.3m/s

Figure 5: The experiment of the robot dog moving around the rectangular field one time at different speeds.

the accuracy error is 0.15% and 0.17%, less than 1%. Several sets of experimental results effectively prove the effectiveness of adding angular velocity compensation for algorithm optimization.

3.3. Experiment Around the Rectangular Field Three Times

To further verify the feasibility of the algorithm, an experiment was carried out in which the robot dog moved around the rectangle field three times (Figure **6**), and the movement speed was 0-0.3m/s. The position error of the final state estimation is within 15cm, the accuracy error is 0.2%, and is less than 1%.

The above analysis and experiments show that the IEKF algorithm with angular velocity compensation can significantly optimize the position state estimation results of the robot dog.

Table 3: Summary of Position Estimation Errors Along A-D Motion for One Circle (cm)

Groups Parameters		1	2	3	4	5	6	7	8	9	10	11	RMSE
EKF (No angular velocity compensation)	0-0.2m/s	62.5	66.2	68.2	63.7	72.4	68.1	56.3	70.2	58.3	56.7	61.8	64.3
	0-0.3m/s	99.2	100.2	103.2	96.5	98.9	102.5	104.2	99.3	97.5	105.2	104.7	101.4
EKF (With angular velocity compensation)	0-0.2m/s	27.1	28.2	27.5	31.5	30.2	26.8	29.2	28.5	27.6	29.1	27.6	28.5
	0-0.3m/s	29.2	30.2	31.5	33.1	28.8	29.7	30.5	31.4	32.1	29.8	30.2	30.6
IEKF (No angular velocity compensation)	0-0.2m/s	51.5	50.3	52.7	54.3	58.1	49.5	55.7	53.6	57.2	53.0	52.6	53.6
	0-0.3m/s	97.3	96.3	95.6	89.2	88.7	100.7	95.6	87.3	84.2	88.3	100.8	93.1
IEKF (With angular velocity compensation)	0-0.2m/s	4	3.7	5.2	4.2	4.6	3.5	3.1	3.5	3.4	3.2	2.3	3.8
	0-0.3m/s	4.6	3.5	5.1	4.8	4.1	4.3	2.8	3.2	4.2	3.6	5.2	4.2

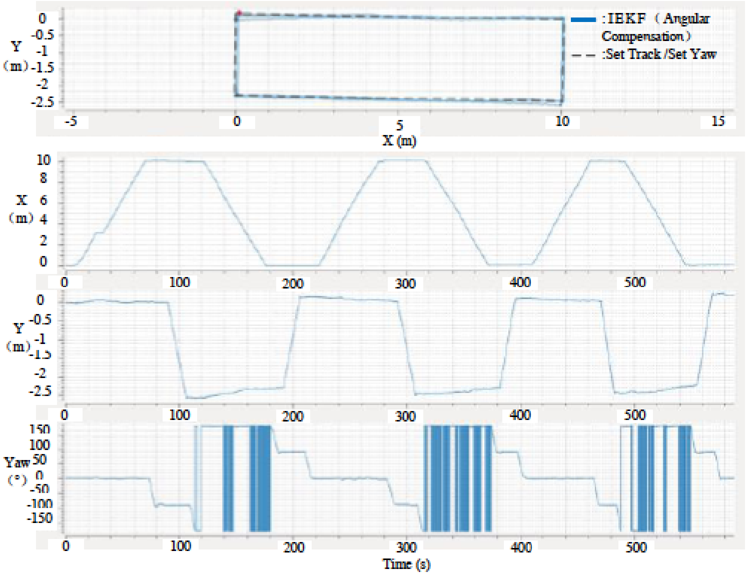


Figure 6: The experiment of the robot dog moving around the rectangular field three times at a speed of 0-0.3m/s.

4. CONCLUSION

Aiming at the problem of large error in the position state estimation of the quadruped robot dog, this paper adds angular velocity compensation to the IMU data based on the IEKF algorithm and solves the problem of Yaw angular velocity drift around the Z axis. The results of a series of multiple experiments with different travel speeds, different travel distances and different steering angles all show that after adding angular velocity compensation, the position state estimation accuracy of the robot dog is greatly improved, exceeding 84.9%, which effectively suppresses the deviation of the position and state estimation of the quadruped robot dog.

ACKNOWLEDGMENT

This work was supported by the Enterprise commissioned development project of Jinan Huibang Intelligent Technology Co., Ltd. (No. W2021293), the Enterprise commissioned development project of Jinan Huibang Automatic Control Co., Ltd. (No. W2020232) and the Independent Innovation Team Project of Jinan City (No.2019GXRC013).

REFERENCES

[1] S. Bonnabel, "Left-invariant extended Kalman filter and attitude estimation," 2007 46th IEEE Conference on Decision and Control, 2007; pp. 1027-1032. https://doi.org/10.1109/CDC.2007.4434662

[2] S. Bonnable, P. Martin and E. Salaün, "Invariant Extended Kalman Filter: theory and application to a velocity-aided

attitude estimation problem," Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference, 2009; pp. 1297-1304. https://doi.org/10.1109/CDC.2009.5400372

[3] Barrau A. Non-linear state error based extended Kalman filters with applications to navigation [D]. Mines Paristech 2015.

[4] M. Reinstein and M. Hoffmann, "Dead Reckoning in a Dynamic Quadruped Robot Based on Multimodal Proprioceptive Sensory Information," in IEEE Transactions on Robotics, 2013; 29(2): 563-571. https://doi.org/10.1109/TRO.2012.2228309

[5] Hutter M, Gehring C, Bloesch M, et al. StarlETH: A compliant quadrupedal robot for fast, efficient, and versatile locomotion[M]//Adaptive Mobile Robotics. 2012: 483-490. https://doi.org/10.1142/9789814415958_0062

[6] Bloesch M, Hutter M, Hoepflinger MA, et al. State estimation for legged robots-consistent fusion of leg kinematics and IMU [J]. Robotics, 2013; 17: 17-24. https://doi.org/10.15607/RSS.2012.VIII.003

[7] Bloesch M, Gehring C, Fankhauser P, Hutter M, Hoepflinger MA and Siegwart R. "State estimation for legged robots on unstable and slippery terrain," 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013; pp. 6058-6064. https://doi.org/10.1109/IROS.2013.6697236

[8] A. Barrau and S. Bonnabel, "The Invariant Extended Kalman Filter as a Stable Observer," in IEEE Transactions on Automatic Control, 2017; 62(4): 1797-1812. https://doi.org/10.1109/TAC.2016.2594085

[9] Hartley R, Ghaffari M, Eustice RM, Grizzle JW. Contact-aided invariant extended Kalman filtering for robot state estimation [J]. The International Journal of Robotics Research, 2020; 39(4): 402-430. https://doi.org/10.1177/0278364919894385

[10] Lin T-Y, Zhang R, Yu J, Ghaffari M. Legged Robot State Estimation using Invariant Kalman Filtering and Learned Contact Events[C]. 5th Annual Conference on Robot Learning 2021.

[11] MAO Jun, FU Hao, CHU Chaoqun, HE Xiaofeng, CHEN Changhao. A Review of Simultaneous Localization and Mapping Based on Inertial-visual-lidar Fusion [J/OL]. Navigation Positioning and Timing: 1-19[2022-07-14]. http://kns.cnki.net/kcms/detail/10.1226.V.20220611.1745.012.html

Received on 15-07-2022 Accepted on 20-08-2022 Published on 30-08-2022

DOI: https://doi.org/10.31875/2409-9694.2022.09.03