International Core Journal of Engineering Volume 10 Issue 6, 2024
ISSN: 2414-1895 DOI: 10.6919/ICJE.202406_10(6).0025

Control System for Outdoor Robot on Basis of Multi-Sensor Fusion

Tianbao Yuan[1, a], Jun Zhao[2, b], and Jing Hou[2, c]

[1] Sichuan University of Science and Engineering, Automation and Information Engineering, Yibin 644000, P. R. China

[2] Artificial Intelligence Key Laboratory of Sichuan Province, Yibin 644000, P.R. China

[a]yTianBao@foxmail.com, [b]zhaojun@suse.edu.cn, [c]houjin828@sina.com

Abstract

Real-time kinematic (RTK) positioning, odometry, and data fusion of trajectory tracking are pivotal in automated navigation and positioning systems within control engineering. RTK leverages the Global Navigation Satellite System to offer high-precision positioning on a global scale, odometry provides incremental movement information of the vehicle, and trajectory tracking ensures precise adherence to a predefined navigational path. This study aims to investigate how to effectively integrate these various data sources to enhance the accuracy, robustness, and trajectory tracking performance of navigation systems.

Keywords

Trajectory Tracking; Data Fusion; Unscented Kalman Filter.

1. Introduction

When robots operate outdoors, precise positioning and trajectory tracking are essential. Navigation systems rely on continuous and accurate location data to guide robot movement, with trajectory tracking ensuring strict adherence to predetermined paths, and minimizing deviation. Currently, most studies use multi-sensor fusion or optimized vehicle position estimation[1], GNSS is usually used in combination with INS[3]. Facing the limitations of RTK-GPS in obstacle-dense environments, integrating wheel odometers and Inertial Measurement Units (IMU) with Global Navigation Satellite System (GNSS) data establishes a more stable and precise robot positioning mechanism. This fusion method, often nonlinear, employs the Unscented Kalman Filter[7] in robot position estimation. The performance evaluation method can be compared statistically by using mean absolute error and root mean square (RMS) error on the travel path, as described in[8]. Considering the practical limitations of continuous-time systems, especially in state sampling, this study adopts a discrete-time model. A novel control strategy was developed by discretizing the continuous-time kinematic equations and applying linear approximation. The strategy ensures that the system's tracking error approaches zero. Finally, the algorithm was applied to practical engineering through the ROS[9,10] system to verify its feasibility.

2. Structure of Control System

2.1 Hardware Implement Overview

The hardware structure is shown in Fig. 1, The framework proposed in this paper is validated using data gathered from RTK, IMU, and odometry.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202406_10(6).0025

Components are interconnected using a CAN bus for reliable data transfer, with USB interfaces employed for debugging and manual control. Sensor arrays are configured for optimal coverage and precision.

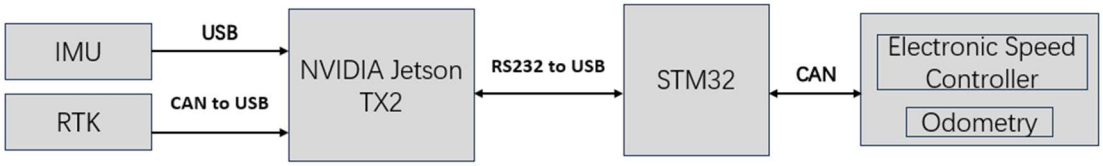


Fig. 1 system hardware structure diagram

2.2 Software Implement Overview

The software structure is shown in Fig. 2, When conducting experiments, ensure that the GPS device is initialized. Once the RTK correction signal is received and configured, the robot can commence operation. The entire system is composed of UKF position estimator and controller design. This work uses the Robot Operating System (ROS) system to develop all programs.

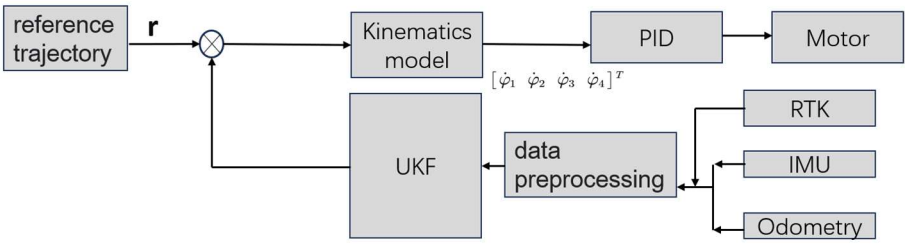


Fig. 2 system software structure diagra

When conducting experiments, we need to ensure that the GPS device is initialized. When the RTK correction signal is received and set, the robot can start working. After loading the trajectory, the robot starts the navigation task at any location near the start of the path.

3. Implement of Pose Estimation

The position estimation process involves two key stages: data filtering and the fusion of data from multiple sensors. Due to the different sampling frequencies of RTK and odometer sensors, this results in orientation jumps during the positioning process.

Therefore, the introduction of Mahalanobis distance, denoted as D_p and D_φ, where D_p and D_φ represent measurement errors. These errors must be kept below the predetermined thresholds of $D_{p,t}$ and $D_{\varphi,t}$. In this study, the threshold values for the robot model are set as $D_{p,t}=3.68$ and $D_{\varphi,t}=0.4$. These parameters are defined based on empirical knowledge embedded in the model. If $D_p<D_{p,t}$ and $D_\varphi<D_{\varphi,t}$, the correction set will simultaneously consider position and orientation data from the odometry. If $D_p<D_{p,t}$ and $D_\varphi \geq D_{\varphi,t}$, the correction set will only consider position data from the odometry. If $D_p<D_{p,t}$ and $D_\varphi<D_{\varphi,t}$ the correction set will only consider orientation data from the odometry. Through the aforementioned process, pre-filtered data points can be obtained, and they are used as observations in the UKF framework.

The method based on the Unscented Kalman Filter (UKF) utilizes Sigma points to describe nonlinear systems, preserving higher-order terms with greater accuracy without distortion resulting from simplification. Assume there is a discrete-time nonlinear system, as shown in equation (1).

ISSN: 2414-1895 DOI: 10.6919/ICJE.202406_10(6).0025

$$X_{k+1} = \Upsilon(X_k, k, Q_k); Y_k = G(X_k, R_k) \tag{1}$$

Where Υ and G represent the dynamic model of the nonlinear system, and it is assumed that they are known. X_k is the system's unobserved state, k is the system input, Y_k is the unique observed state, Q_k is the process noise, and R_k is the measurement noise of the system observation. When a variable X with dimension L passes through a nonlinear system $Y = G(X)$, where X has a mean $\overline{X}$ and covariance P_X, a matrix called the sigma matrix is formed, which has a size of $2L+1$. The sigma points χ_i and the corresponding weights W_i can be defined as follows:

$$\lambda = \alpha^2(L+\kappa) - L; \chi_k^{(0)} = \overline{X} \tag{2}$$

$$\chi_k^{(i)} = \overline{X} - \left(\sqrt{(L+\lambda)P_X}\right)_{i-L}, i = L+1, ..., 2L \tag{3}$$

$$W_i^{(m)} = W_i^{(c)} = \frac{1}{2(L+\lambda)}, i = 1, 2, ..., 2L \tag{4}$$

Where λ is the scaling parameter, α is used to determine the distribution of sigma points around X and is typically set to a small positive value, such as 0.01. β is used to incorporate prior knowledge about the distribution of X, κ is a secondary scaling parameter that is set to 0, and $\sqrt{(L+\lambda)P_X}$ is the i-th row of the matrix square root, with sigma points predicted through the transformation matrix Υ. Based on the weights of each sigma point, the predicted mean $\overline{X}_{k|k}^{-}$ and the predicted covariance matrix $\overline{P}_{k|k}^{-}$ can be obtained, which include the process noise Q_k.

$$\overline{X}_{k|k}^{-} = \sum_{i=0}^{2N} W_i^{(m)} \chi_{k|k-1}^{(i)} \tag{5}$$

$$P_{k|X}^{-} = \sum_{i=0}^{2N} W_i^{(m)} \left(\chi_{k|k-1}^{(i)} - \overline{X}_{k|k}^{-}\right)\left(\chi_{k|k-1}^{(i)} - \overline{X}_{k|k}^{-}\right)^T + Q_k \tag{6}$$

Additionally, the calculated sigma points are propagated through the nonlinear function . An approximation of the measurement mean based on the predicted state is shown as in equation (7).

$$\overline{Y}_{k|k}^{-} = \sum_{i=0}^{2N} W_i^{(m)} Y_{k|k-1}^{(i)} \tag{7}$$

We use the covariance and weighted average of the posterior sigma points to estimate the measurement covariance matrix $P_{\overline{Y}_k\overline{Y}_k}$,which includes measurement noise R_k, and the cross-correlation measurement covariance matrix $P_{X_kY_k}$ for Y. The specific methods are described in equations (8) and (9).

ISSN: 2414-1895 DOI: 10.6919/ICJE.202406_10(6).0025

$$P_{\bar{Y}_k\bar{Y}_k} = \sum_{i=0}^{2N} W_i^{(c)} \left(Y_{k|k-1}^{(i)} - \bar{Y}_{k|k}^{-}\right)\left(Y_{k|k-1}^{(i)} - \bar{Y}_{k|k}^{-}\right)^T + R_k \tag{8}$$

$$P_{X_kY_k} = \sum_{i=0}^{2N} W_i^{(c)} \left(\chi_{k|k-1}^{(i)} - \bar{X}_{k|k}^{-}\right)\left(Y_{k|k-1}^{(i)} - \bar{Y}_{k|k}^{-}\right)^T \tag{9}$$

Ultimately, update the mean and covariance matrix of the system's state and compute the Kalman gain K_k.

$$K_k = P_{X_kY_k} P_{\bar{Y}_k\bar{Y}_k}^{-1} \tag{10}$$

$$\bar{X}_{k|k} = \bar{X}_{k|k}^{-} + K_k\left(Y_k - \bar{Y}_{k|k}^{-}\right) \tag{11}$$

$$P_{k|X} = P_{k|X}^{-} - K_k P_{\bar{Y}_k\bar{Y}_k} K_k^T \tag{12}$$

4. Implement of Trajectory Tracking Control

4.1 Design of Control Law

Converting the continuous-time model of wheeled mobile robots to a discrete-time model accommodates high-frequency sampling and accuracy requirements. Through discretization and linearization of the kinematic model, and employing a zero-order hold method for control inputs, an accurate discretized model is achieved. R represents half the track width, and r represents the radius of the drive wheel.

$$Y(k+1) = Y(k) + \Phi(k)u(k) \tag{13}$$

$$\Gamma = \begin{bmatrix} \frac{r}{4} & \frac{r}{4} & \frac{r}{4} & \frac{r}{4} \\ \frac{r}{4R} & \frac{-r}{4R} & \frac{r}{4R} & \frac{-r}{4R} \end{bmatrix} \tag{14}$$

$$\mathcal{H}(k) = H(k)u(k) = \begin{bmatrix} \frac{r}{4R} & -\frac{r}{4R} & \frac{r}{4R} & -\frac{r}{4R} \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \varphi_4 \end{bmatrix} \tag{15}$$

$$\mathcal{L}(k) = G(k)u(k) = \begin{bmatrix} \frac{r}{4R} & \frac{r}{4R} & \frac{r}{4R} & \frac{r}{4R} \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \varphi_4 \end{bmatrix} \tag{16}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202406_10(6).0025

In the above equation (13), $\Phi(k)=\begin{bmatrix}R(k)\\0\quad T\end{bmatrix}_{3\times2}\Gamma$, where $R(k)=\beta(k)\Lambda(k)$. $\Lambda(k)=\begin{bmatrix}\cos\gamma(k) & -d\sin\gamma(k)\\ \sin\gamma(k) & d\cos\gamma(k)\end{bmatrix}$, $\beta(k)=\frac{2\sin\left(\frac{T}{2}\mathcal{H}(k)\right)}{\mathcal{H}(k)}$, $\gamma(k)=\theta(k)+\frac{T}{2}\mathcal{H}(k)$. $\mathcal{H}(k)$ and $\mathcal{L}(k)$ are respectively $H(k)u(k)$ and $G(k)u(k)$.

It is evident that when $\mathcal{H}(k)=0$, the equation becomes meaningless. We need to address the issue of control input coupling. Considering the robot's orientation θ as an uncontrolled state, and focusing solely on the robot's position coordinates, let the incremental change in the robot's state be $\Omega(k)=R(k)u(k)$. By performing the transformation, we obtain:

$$\Omega(k)=R(k)u(k)=\beta(k)\Lambda(k)\Gamma u(k) \tag{17}$$

Set $\eta(k)$ as a virtual control law, and perform the following transformation on the actual control law $u(t)$ and the virtual control law $\eta(k)$:

$$f(u(k))=\beta(k)u(k)=\eta(k),\ \eta(k)=\begin{bmatrix}\eta_1(k)\\ \eta_2(k)\end{bmatrix}=\begin{bmatrix}\beta(k)\mathcal{L}(k)\\ \beta(k)\mathcal{H}(k)\end{bmatrix} \tag{18}$$

Having obtained the virtual control law $\eta(k)$, the actual control law $u(k)$ can be derived through the calculation of the inverse function based on the defined function relationship. The transformation relation is as follows:

$$u(k)=f^{-1}(\eta(k))=\begin{cases}\frac{2}{T}\Gamma^{-1}[\eta_1(k)0]^T, & \eta_2(k)=0\\ \frac{2}{T}\Gamma^{-1}\begin{bmatrix}\frac{\eta_1(k)}{\eta_2(k)}\arcsin\frac{\eta_2(k)}{2}\\ \arcsin\frac{\eta_2(k)}{2}\end{bmatrix}, & \eta_2(k)\neq0\end{cases} \tag{19}$$

4.2 Stability Analysis

Trajectory tracking error:

$$\sqrt{e(k)=p(k)-r(k)} \tag{20}$$

When the robot's motion trajectory meets condition $e(k)\leqslant\delta$, it is considered that the robot has achieved tracking of the target trajectory, where δ is a real number near the origin.

The designed control protocol is shown in (21):

$$\eta(k)=\Lambda^{-1}(k)\left[(K-I)p(k)+r(k+1)-Kr(k)\right] \tag{21}$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202406_10(6).0025

If there exists a matrix k such that $\overline{K} = \| K \| - \| \tilde{I}(k) \| \| K - I \| < 1$, Then, the stability error of the system can converge to the neighborhood of the origin, where $I - \Lambda(k)\tilde{\Lambda}^{-1}(k) = \tilde{I}(k)$.
Substituting the control protocol (21) into the error equation (20) we can get:

$$\begin{aligned} & e(k+1) \\ &= p(k+1) - r(k+1) \\ &= p(k) + \Lambda(k)\eta(k) - r(k+1) \\ &= Kp(k) - (K-I)p(k) + \Lambda(k)\Lambda^{-1}(k)Kp(k) - \Lambda(k)\Lambda^{-1}(k)p(k) \\ &\quad + \Lambda(k)\Lambda^{-1}(k)r(k+1) - r(k+1) - \Lambda(k)\Lambda^{-1}(k)Kr(k) \\ &= Ke(k) - \tilde{I}(k)(K-I)e(k) - \tilde{I}(k)\Delta r(k) \end{aligned} \tag{22}$$

It can be seen from the definition of $\tilde{I}$ that there is a value δ_1 in the neighborhood of the origin such that: $\| \tilde{I} \| < \delta_1$. When the system's sampling period meets certain conditions, a value δ_2 exists in the neighborhood of the origin such that:

$$\| r(k+1) - r(t) \| = \| \Delta r(k) \| < \delta_2 \tag{23}$$

Taking the norm of (22) we can get:

$$\| e(k+1) \| \leqslant \| e(k) \| (\| K \| - \delta_1 \| K - I \|) + \delta_1\delta_2 \tag{24}$$

Iterating over (24) gives:

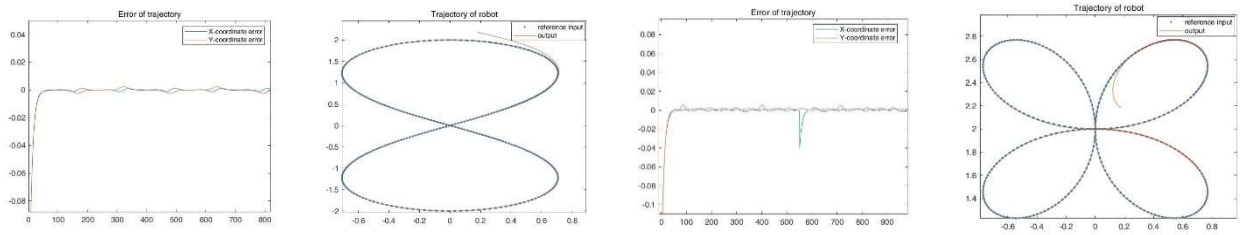
$$\begin{aligned} \| e(k) \| &\leqslant \| e(k-1) \| (\| K \| - \delta_1 \| K - I \|) + \delta_1\delta_2 \| e(1) \| \\ &\leqslant \| e(0) \| (\| K \| - \delta_1 \| K - I \|) + \delta_1\delta_2 \end{aligned} \tag{25}$$

Because $e(0) = p(0) - p(0) = 0$, $\overline{K} = \| K \| - \delta_1 \| K - I \| < 1$, By scaling and iterating we get:

$$\| e(k+1) \| \leqslant \| \frac{\delta_1\delta_2\left(1 - \overline{K}^k\right)}{1 - \overline{K}} \| \tag{26}$$

Therefore, the robot's consistency error will eventually converge to the neighborhood of the origin.
To verify the effectiveness of the trajectory tracking control protocol proposed in this section, two sets of simulations with different initial states were conducted using MATLAB, with figure-eight trajectories and clover as the reference trajectories, respectively. The results are shown in Fig. 3.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202406_10(6).0025



(a) (b) (c) (d)

Fig. 3 Trajectory Tracking and Tracking Error

5. Experimental Result

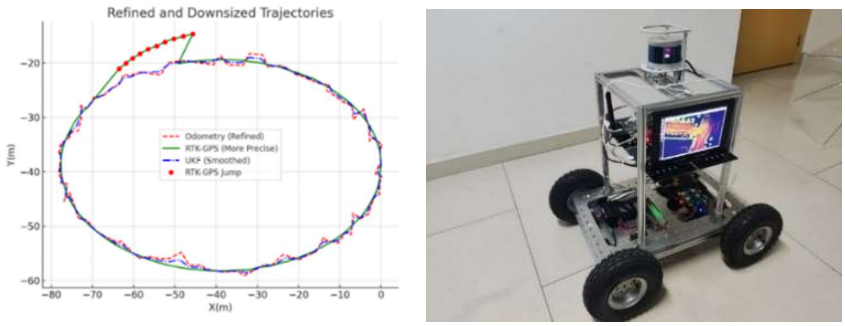
In this experiment, we tested the positioning system implemented through UKF technology within the campus, selecting an elliptical path in the campus as the test site. It should be noted that due to the indoor campus environment and specific speed and path restrictions, the experimental design was relatively simplified. Although the test scenario was basic, it has fully verified the practical operability of integrating the aforementioned technologies.

The experimental data is shown in Table 1, compared to the actual ground conditions, our UKF positioning system has an error rate of only 0.787%. As shown in Table 1, whether it's the minimum error, average error, or standard deviation, UKF's performance is superior to that of traditional odometers and RTK systems. The focus of this paper is whether the error percentage generated by UKF has a significant advantage over other technologies. Although the scope of the experiment and the path settings differ, the UKF method proposed in this paper still yields acceptable results in terms of average error and standard deviation.

Table 1. Distance Comparison Using Odometer, RTK GPS, and UKF

Methods	Ground Truth(m)	Ground Truth(m)	Error(%)	Mean Error(m)
Odometry	165.19	160.06	3.105	5.13
RTK-GPS	165.19	162.91	1.380	2.28
UKF	165.19	163.89	0.787	1.30

Ultimately, Fig. 4 displays the trajectory comparison among the odometer, RTK-GPS, and UKF methods. It's particularly noteworthy that in the top left corner of Fig. 4, a significant jump is observed with the RTK-GPS. In contrast, the UKF position estimator demonstrated higher stability. Simultaneously, the UKF position estimator also effectively reduced the errors that might accumulate when relying solely on the odometer.



(a) experimental result (b) Experimental Robot

Fig. 4 Comparison of Trajectories Using Odometer, RTK, and UKF and Experimental Robot

ISSN: 2414-1895 DOI: 10.6919/ICJE.202406_10(6).0025

6. Conclusion

This paper presents a model that has been discretized and subjected to linearization. A new control protocol was designed for the discretized kinematic model post-approximate linearization, allowing the system's tracking error to converge to a smaller value. The system incorporates data from RTK, odometer, and IMU sensors. The research utilized a robot autonomously assembled in the laboratory as the experimental subject to validate the effectiveness of the proposed UKF vehicle positioning and path tracking method. Over a path of 165.19 meters, the travel distance estimated by UKF exhibited an error of 0.787%.

Furthermore, this study is a key practical endeavor in outdoor environments for robot motion, encompassing robot positioning and path tracking. Such projects enable us to more conveniently learn, engage with, and practice the core technologies of autonomous driving. Due to the limited speed and path constraints within the campus, this study has only preliminarily demonstrated the successful integration of robot positioning and path tracking. Moving forward, deploying the vehicle in more complex testing scenarios becomes the next critical task.

Acknowledgments

We appreciate the valuable insights provided by Teacher Zhao jun and student Lin Zhijiang. We appreciate Artificial Intelligence Key Laboratory of Sichuan Province for providing essential resources, and our colleagues for their constructive feedback. Special gratitude to our families for their endless support and encouragement.

References

[1] Mohamed S A S, Haghbayan M H, Westerlund T, et al. A survey on odometry for autonomous navigation systems[J]. IEEE access, 2019, 7: 97466-97486.

[2] Mikov A, Panyov A, Kosyanchuk V, et al. Sensor fusion for land vehicle localization using inertial MEMS and odometry[C]//2019 IEEE International Symposium on Inertial Sensors and Systems (INERTIAL). IEEE, 2019: 1-2.

[3] Zhang C, Zhao X, Pang C, et al. Adaptive fault isolation and system reconfiguration method for GNSS/INS integration[J]. IEEE Access, 2020, 8: 17121-17133.

[4] Zhou J, Traugott J, Scherzinger B, et al. A New Integration Method for MEMS Based GNSS/INS Multi-sensor Systems[C]//Proceedings of the 28th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2015). 2015: 209-218.

[5] Fu L, Hu C, Kong L. A Sea-Sky Line Detection Aided GNSS/INS Integration Method for Unmanned Surface Vehicle Navigation[C]//Proceedings of the 30th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2017). 2017: 1809-1815.

[6] Lefferts E J, Markley F L, Shuster M D. Kalman filtering for spacecraft attitude estimation[J]. Journal of Guidance, control, and Dynamics, 1982, 5(5): 417-429.

[7] Wan E A, Van Der Merwe R. The unscented Kalman filter for nonlinear estimation[C]//Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373). Ieee, 2000: 153-158.

[8] Ng K M, Johari J, Abdullah S A C, et al. Performance evaluation of the RTK-GNSS navigating under different landscape[C]//2018 18th International Conference on Control, Automation and Systems (ICCAS). IEEE, 2018: 1424-1428.

[9] Quigley M, Gerkey B, Smart W D. Programming Robots with ROS: a practical introduction to the Robot Operating System[M]. " O'Reilly Media, Inc.", 2015.

[10] Shimchik I, Sagitov A, Afanasyev I, et al. Golf cart prototype development and navigation simulation using ROS and Gazebo[C]//MATEC Web of Conferences. EDP Sciences, 2016, 75: 09005.