RESEARCH ON AUTONOMOUS PATH PLANNING FOR VINEYARD ROBOTS BASED ON LASER SLAM COMBINED WITH THE ROBUST TIME ELASTIC BAND ALGORITHM
/
基于激光 SLAM 与 R-TEB 算法结合的葡萄园机器人自主路径规划研究

Pengcheng LV [1], Jinhong ZHANG [2], Wei CHANG [3,*], Wensheng WU [4]
[1] Ocean University of China, College of Engineering, Qingdao, China
[2] Qingdao Municipal Construction Development Co., Ltd., Qingdao, China
[3] Shanghai Lixin University of Accounting and Finance, School of Finance, Shanghai, China
[4] Hefei University of Technology, School of Economics, Hefei, China
Tel: +86 18901783506; E-mail: changwei0307@163.com
Corresponding author: Wei Chang

DOI: https://doi.org/10.35633/inmateh-77-80

Keywords: *Vineyard robots; SLAM; Robust-Time Elastic Band; Path planning*

ABSTRACT
In recent years, the development of vineyard robots has emerged as a significant development in agricultural equipment, playing an increasingly vital role in precision agriculture and intelligent operations. These robots are capable of precise navigation, obstacle avoidance, and real-time path planning within agricultural settings. The paper employs laser Simultaneous Localization and Mapping (SLAM) technology as the primary method for achieving real-time, accurate positioning of the robot, thereby providing reliable environmental perception capabilities and prior map information for the vineyard robot. The Robust-Time Elastic Band (R-TEB) local planning algorithm developed in this study automatically generates a smooth, continuous inspection path within the operational area. This objective is pursued by the consideration of parameters such as the robot's working width, minimum turning radius, and operational strip width, with the aim of achieving a minimization of energy consumption. Utilizing the Root Mean Square Error (RMSE) metric to gauge prediction accuracy, the R-TEB algorithm yielded values ranging from 0.016 to 0.022 meters, while the TEB algorithm produced values between 0.012 and 0.025 meters. The findings indicate that the R-TEB algorithm optimizes trajectory quality in vineyard environments, thereby enhancing the robot's autonomous navigation capabilities and obstacle avoidance efficiency.

摘要
近年来，葡萄园机器人的研发已成为农业设备领域的重要突破，在精准农业与智能化作业中发挥着日益关键的作用。这类机器人能够在农业场景中实现精准导航、障碍物规避及实时路径规划。本文采用激光同步定位与建图（SLAM）技术作为核心方法，实现机器人实时精准定位，从而为葡萄园机器人提供可靠的环境感知能力和预先地图信息。研究开发的鲁棒时间弹性带（R-TEB）局部规划算法，能在作业区域内自动生成平滑连续的巡检路径。该算法通过综合考虑机器人作业宽度、最小转弯半径及作业带宽度等参数，实现能耗最小化目标。采用均方根误差（RMSE）指标评估预测精度时，R-TEB 算法得出 0.016 至 0.022 米的误差值，而 TEB 算法误差值介于 0.012 至 0.025 米之间。研究表明，R-TEB 算法能优化葡萄园环境中的轨迹质量，从而提升机器人的自主导航能力和避障效率。

INTRODUCTION

Multi-object obstacle avoidance and path planning represent core challenges in the realm of autonomous navigation for vineyard robots. The primary task of the system is to generate collision-free and efficient motion trajectories. This is particularly challenging in complex agricultural environments such as vineyards, where the system must navigate regular row spacing, dense obstacles, and sparse point cloud data *(Fox et al., 1997)*.

Autonomous positioning and navigation technology for vineyard robots serves as the core foundational architecture within smart agriculture systems *(Qiu et al., 2020)*. Its performance directly determines the operational autonomy and control robustness of agricultural robots in unstructured agricultural environments.

SLAM-based navigation systems construct high-precision 3D point cloud maps using LiDAR point cloud data, providing mobile robots with centimeter-level positioning accuracy and dynamic environmental perception capabilities.

Through the combined use of front-end odometry and back-end optimization, real-time pose estimation and incremental map updating can be achieved *(Zhou et al., 2021)*. *Gao et al. (2021)* developed a tightly coupled factor-graph optimization framework that integrates LiDAR, IMU, and GNSS data, where loop-closure detection and motion constraints are introduced as nonlinear factors in the back-end optimization to support 3D reconstruction. *Nie et al. (2021)* proposed a lightweight SLAM system that builds a sparse tree-trunk feature map and suppresses branch- and leaf-induced occlusion by analyzing point cloud density gradients. Furthermore, a multi-criteria trunk detection algorithm was developed that combines curvature features with reflectance-intensity thresholds to enable accurate trunk identification.

In the context of local path planning, the TEB algorithm has been demonstrated to exhibit exceptional performance by generating smooth trajectories through the optimization of time elastic bands. However, in sparse point cloud environments, its trajectory optimization efficiency is relatively low, and it lacks a dynamic adjustment mechanism for static obstacles. Subsequent researchers proposed a velocity-constrained TEB variant, which improved path smoothness but still exhibited path jittering in narrow-channel scenarios. *Zhou et al., 2014,* innovatively applied the ant colony optimization algorithm to solve the travelling salesman problem. The experimental data demonstrate that this method significantly enhances the prediction accuracy of agricultural machinery movement trajectories in complex obstacle environments. Addressing the specific requirements of omnidirectional mobile agricultural platforms, *Zhang et al., 2024,* deeply integrated the FMT algorithm with an improved artificial potential field method. This was then combined with B-spline curve path smoothing technology and a gravity adjustment mechanism based on relative distance was introduced. The findings indicate that, in comparison with FMT, RRT, and Informed-RR algorithms, this methodology results in a reduction of search time by over 45%, thereby significantly enhancing the operational efficiency and safety of agricultural machinery in unstructured farmland.

The present paper puts forward a local path planning algorithm for vineyard robots that is based on laser SLAM as prior information. The system framework is illustrated in Figure 1. Lidar is utilized as the visual perception sensor to map the entire vineyard through the SLAM system. The resulting three-dimensional map facilitates positioning and navigation, while the two-dimensional map aids path planning. The positioning data reception and conversion system provides positioning information to the path planning system. The robot relies on prior map information to perform real-time local planning and obstacle avoidance.

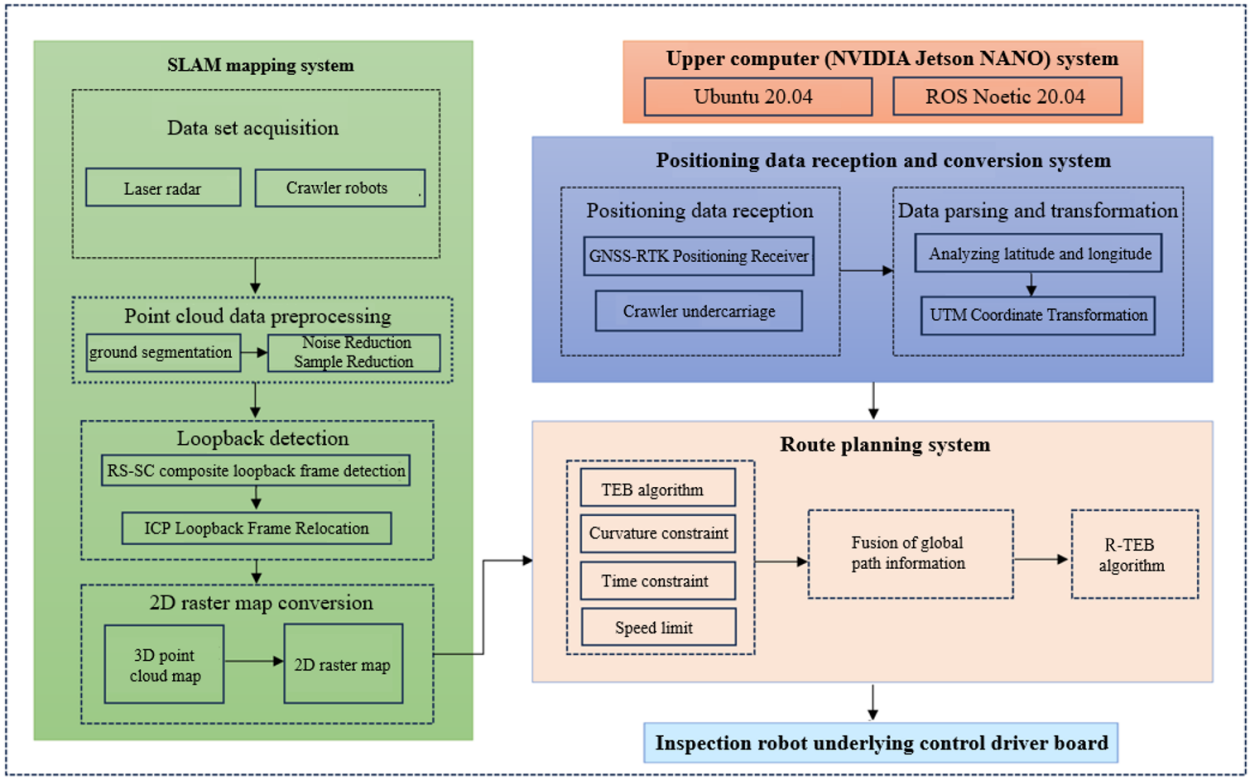

Fig. 1 – System framework

MATERIALS AND METHODS

Vineyard Raw Data Collection

The site selection for data collection by vineyard robots must ensure open space to guarantee stable positioning signals. This site is located at the Thousand-Mu Garden Grape Planting Base in Huantai County, Zibo City, Shandong Province. The standardized planting pattern of 3-meter row spacing and 1.5-meter plant spacing provides an ideal environment for robotic data collection and algorithm testing.

Fig. 2 – The Thousand-Mu Garden

As shown in Figure 3, the vineyard robot is equipped with a LiDAR device mounted on an aluminum profile bracket positioned 1 meter above the ground. For precise positioning, two sets of GNSS antennas—one front and one rear—are secured via powerful magnetic suction cups. This enables the GNSS-RTK system to provide the robot with real-time positioning data accurate to the centimeter level. Additionally, the robot incorporates an artificial intelligence processor. Multiple cables are used to establish system communication: a serial cable connects the control computer to the robot chassis drive board, a USB cable interfaces with the GNSS-RTK positioning receiver, and an Ethernet cable links the system to the LiDAR adapter box. The tracked chassis of the orchard robot is powered through a buck-converter module, which steps down the 48 V supply to 24 V for the LiDAR and further reduces it to 5 V for the AI processor.

Fig. 3 –Crawler-Type Vineyard Robot

1. External Monitor; 2. Artificial Intelligence (AI) Processors;
3. Crawler-Type Mobile Chassis; 4. Dual-Antenna GNSS-RTK; 5. 64-Line 3D LiDAR.

Figure 4 shows the hardware connection diagram of the robot.

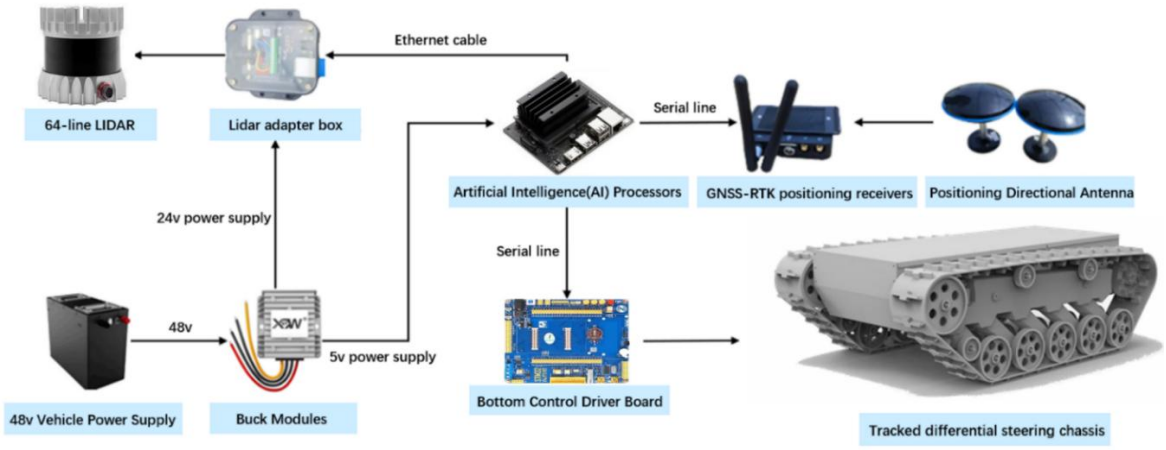

Fig. 4 – Hardware Connection Diagram of Vineyard Robot

SLAM System and R-TEB Algorithm Details

In outdoor orchard environments, the position of the sun continuously shifts from sunrise to sunset, and the intensity of sunlight fluctuates over time. These natural conditions can easily cause visual sensors to misidentify objects or experience positioning errors. In contrast, laser sensors do not rely on external light sources. By emitting their own laser beams for detection, they effectively mitigate interference from changing illumination and maintain stable sensing performance *(Xue et al., 2023)*.

The SC-LeGO-LOAM algorithm was adopted as the SLAM framework in this study. SC-LeGO-LOAM is based on LeGO-LOAM and incorporates loop closure detection using scan context. This improves detection speed compared to LeGO-LOAM *(Shan et al., 2018)*. The system primarily consists of four components: point cloud segmentation, feature extraction, radar odometry and map construction *(Lv et al., 2025)*. As illustrated in Figure 5, point cloud ground segmentation, denoising and downsampling are common pre-processing methods. The effectiveness of these methods was validated through before-and-after image comparisons in our self-created vineyard dataset. The loop closure detection module aligns with the SC-LeGO-LOAM algorithm framework. Odometry calculations use a feature-point-based front-end registration method integrated with the loop closure detection module for collaborative processing *(Qin et al., 2024)*.

Finally, the optimized pose and map data are fed into the graph optimization back-end to enhance orchard mapping accuracy. The resulting 3D map facilitates navigation, while the 2D map supports path planning. The 3D map is converted to 2D in real time, with the current pose serving as input to the path planning system *(Zhang et al., 2014)*.

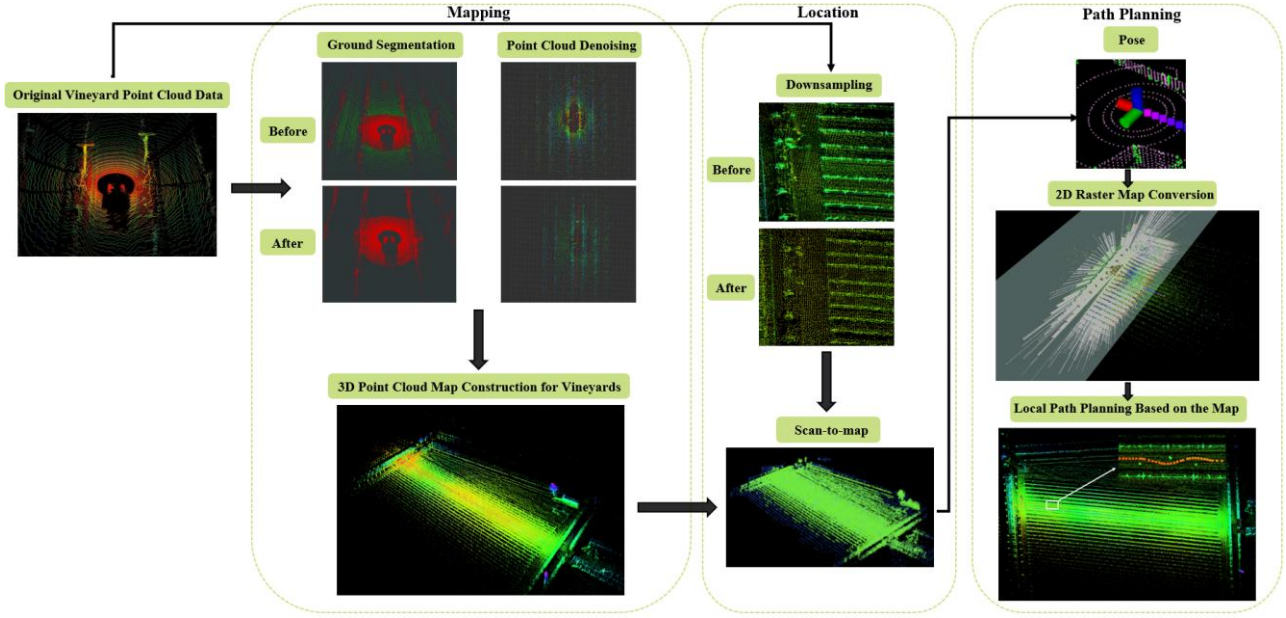

Fig. 5 –Autonomous Navigation Schematic Diagram

In practical engineering applications, to adapt to diverse complex environments and achieve superior obstacle avoidance, the classic "Elastic Band" algorithm describes the motion state of mobile robots through a sequence of poses $X_i = (x_i, y_i, \beta_i)^T \in \mathrm{R}^2 \times S^1$. However, this approach fails to account for temporal constraints, potentially rendering path planning ineffective in dynamic real-world scenarios *(Pire et al., 2017*. To address this issue, *Rösmann et al., 2015,* proposed the TEB algorithm. Building upon the Elastic Band framework, it incorporates temporal interval information by introducing a time difference ΔT_j between consecutive pose sequences. This enables path planning to not only optimize spatial trajectories but also dynamically adjust temporal allocation, thereby enhancing the feasibility of trajectory tracking.

The TEB path can be represented as:

$$B := (Q, \tau) = (X_i, \Delta T_j)_{i=0\ldots n, j=0\ldots n-1} \tag{1}$$

Where: Q denotes the state trajectory; τ denotes the time interval sequence; B is the data structure composed of the state trajectory and the corresponding time interval sequence *(Wu et al., 2021)*.

During the TEB path optimization process, it is essential to ensure that the robot satisfies the Ackerman kinematic model constraint, meaning its turning radius must not be less than the minimum turning radius:

$$P_{\min} = \frac{d_k}{\Delta\beta_k} \tag{2}$$

where: d_k is the Euclidean distance between adjacent trajectory points; $\Delta\beta_k$ is the change in steering angle between adjacent trajectory points. When the robot moves along the trajectory, its velocity v_k and acceleration a_k must satisfy the physical constraints *(Yang et al., 2022)*:

$$v_k = \frac{d_k}{\Delta T_k}, \quad a_k = \frac{v_{k+1} - v_k}{\Delta T_k} \tag{3}$$

Although the TEB algorithm demonstrates strong trajectory optimization capabilities in dynamic environments, it still suffers from insufficient path smoothness. Since TEB primarily focuses on the joint optimization of time and space, the path optimization objective often prioritizes obstacle avoidance and time constraints. This can result in trajectories with numerous abrupt turns and significant curvature changes. Such trajectories not only compromise the stability of robotic motion but also increase the difficulty of execution control. Particularly in narrow or complex terrains, this tendency can lead to increased path tracking errors *(Zhang et al., 2022)*.

To address this issue, this paper proposes an improved R-TEB algorithm. While preserving TEB's computational efficiency and obstacle avoidance capabilities, it further optimizes trajectory smoothness, enhancing the stability and feasibility of robot path tracking *(Zhong et al., 2020)*.

1) Introducing curvature optimization constraints:

B-spline curves possess higher-order continuity, enabling secondary optimization of paths generated by the TEB algorithm to achieve smoother trajectories while preserving their curvature. Trajectory points produced by the TEB algorithm often exhibit a step-like appearance, which is detrimental to smooth tracking. Therefore, B-splines can be employed to refine these paths. The specific process is as follows: Extract key control points from the TEB trajectory. Let the trajectory points generated by TEB be denoted as $X = \{X_0, X_1, \ldots, X_n\}$, where each point X_i contains position information (x_i, y_i) and time interval ΔT_i. Select a subset of key trajectory points P_i as control points for the B-spline to reduce computational complexity while ensuring global consistency of the trajectory.

Construct a B-spline curve using a cubic B-spline (k=3) fit:

$$P(u) = \sum_{i=0}^{n} N_{i,3}(u) P_i \tag{4}$$

Compute the basis function $N_{i,3}(u)$ to ensure the trajectory satisfies smoothness.

Smooth Path Optimization Calculates Curvature $\kappa(u)$ to Evaluate Path Smoothness:

$$\kappa(u) = \frac{x'(u)y''(u) - y'(u)x''(u)}{(x'(u)^2 + y'(u)^2)^{\frac{3}{2}}} \tag{5}$$

By adjusting control points to minimize the curvature change rate $\kappa(u)$, the feasibility of the robot's trajectory is ensured.

2) Introducing curvature optimization constraints:

By optimizing the time interval ΔT_k, the TEB algorithm reduces abrupt velocity changes while ensuring path feasibility, thereby enhancing trajectory traceability. To ensure safety, the TEB algorithm incorporates obstacle distance constraints during optimization, guaranteeing the robot maintains a safe distance from obstacles *(Zhou et al., 2020)*:

$$e_{\Gamma} = S\left(\frac{d_{\min,j} - r_{\text{omin}}}{r_{\text{pmin}}}\right) \tag{6}$$

where: $d_{\min,j}$ represents the minimum distance between the robot and an obstacle or target point; r_{omin} denotes the obstacle avoidance safety distance; r_{pmin} indicates the maximum permissible deviation when reaching the target point. To enhance the robot's task execution efficiency, the TEB algorithm minimizes the total travel time:

$$f_{\text{time}} = \sum_k \Delta T_k \tag{7}$$

By optimizing path time allocation, the TEB algorithm ensures that the robot finds the optimal trajectory while satisfying all constraints.

The core of the TEB algorithm is to optimize a weighted multi-objective function, ensuring that the generated path satisfies kinematic constraints while adapting to complex environments. Its objective function is as follows:

$$J_{\text{TEB}} = \sum_k \gamma_k f_k(B) \tag{8}$$

where: $f_k(B)$ represents the constraint function in path planning, related to robot motion constraints, obstacle avoidance, and path smoothness; γ_k denotes the weighting of each constraint, determining the importance of different optimization objectives.

3) Introducing velocity smoothing constraints

In path planning and trajectory optimization, the smoothness of a robot's velocity directly impacts the stability and execution efficiency of its motion. To enhance path tracking performance in dynamic environments, this paper introduces a velocity smoothing constraint. This constraint aims to prevent unreasonable velocity fluctuations during path execution, ensuring smooth motion.

In traditional path optimization algorithms, particularly the TEB algorithm, velocity control often lacks smoothness. This can lead to excessive acceleration or abrupt deceleration during execution, increasing control difficulty and potentially causing system oscillations. To address this, the following constraint is proposed by optimizing the smoothness of velocity changes *(Jiang et al., 2022)*:

$$J_{\text{smoothness}} = \sum_{i=1}^{N} (V_i - V_{i-1})^2 \tag{9}$$

Here, V_i and V_{i-1} denote the linear velocities at trajectory points P_i and P_{i-1}, respectively. By minimizing the velocity difference between adjacent trajectory points, smooth velocity transitions are maintained, preventing large velocity discontinuities and reducing sudden acceleration or deceleration events.

Additionally, to achieve finer control over velocity smoothness, velocity smoothing constraints can be integrated with other trajectory optimization constraints—such as time interval optimization and obstacle avoidance optimization—to work synergistically throughout the overall optimization process. This approach ensures obstacle avoidance and trajectory smoothness while further guaranteeing more gradual velocity changes during path tracking. It reduces requirements on the dynamics model and enhances both the precision and stability of trajectory execution.

4) Integrate global path information

In dynamic environments, relying solely on local path optimization often leads to getting stuck in local optima, especially when obstacles are unevenly distributed or path changes are significant. To enhance the global consistency of path planning and prevent robots from deviating from global objectives during local planning, this paper proposes integrating global path information into the local path optimization process to improve the overall feasibility and stability of the path. In this algorithm, local path planning is optimized using the TEB algorithm. However, a global path deviation constraint is introduced during this process to ensure consistency between the local trajectory and the global path. This constraint helps the robot maintain the correct direction of travel, thereby preventing path deviation or deadlock issues arising from the local optimization process *(Yang et al., 2015)*.

To ensure consistency between local paths and global paths, the global path deviation constraint is defined as:

$$J_{\text{global}} = \sum_{i=1}^{N} || \, p_i - p_i^{\text{global}} \, ||^2 \tag{10}$$

In the actual optimization process, global path information does not directly influence local path optimization but serves as a guiding factor. After integrating global path information, the robot considers the direction and trend of the global path during local path optimization, preventing deviation from the overall objective in complex environments. This approach not only ensures path feasibility but also enhances path coherence. Particularly in multi-obstacle and dynamic environments, the robot can perform path tracking more effectively. Simultaneously, leveraging global path information to guide local optimization prevents getting stuck in local optima. This enables the robot to maintain effective motion planning in dynamic environments, ultimately boosting the overall efficiency and success rate of path planning.

RESULTS

Vineyard Field Test

The experimental design developed multiple test protocols based on the actual conditions of the Thousand-Mu Garden grape experimental field, aiming to comprehensively validate the application effectiveness of the pose correction algorithm and trajectory tracking control algorithm in a vineyard environment. The experiments encompassed key aspects such as multi-object obstacle detection, path planning, and path tracking, ensuring a realistic reflection of the robot's performance in complex environments. By collecting actual measurement data, this study conducted a detailed assessment of the robot's path tracking accuracy and system robustness under various operational conditions. Data analysis results will validate whether the path planning algorithm generates feasible operational paths in the field and further assess the trajectory tracking algorithm's performance in practical applications.

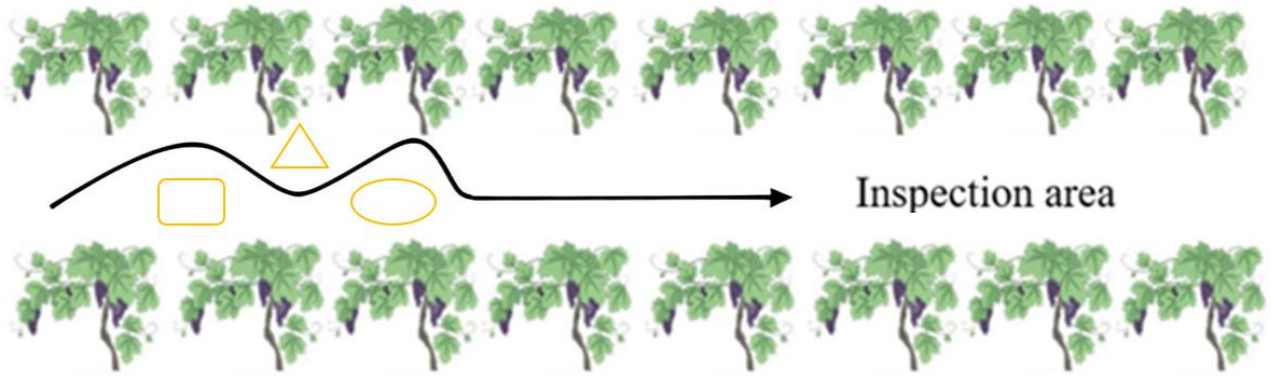

Fig. 6 –Schematic Diagram of Inspection Area

As shown in Figures 6 and 7, the robot can achieve autonomous navigation within a vineyard environment. Under this path planning scenario, the orchard robot can smoothly navigate around obstacles and continue its inspection operations.

Fig. 7 – Robot Obstacle Avoidance Diagram

Test Result Analysis

Based on the recorded data, differences between the R-TEB and TEB algorithms were analyzed with respect to path smoothness, obstacle avoidance capability, and tracking accuracy, enabling a comprehensive assessment of the applicability of the R-TEB algorithm in complex vineyard environments. During testing, the robot traveled along an optimized path, with the R-TEB algorithm adjusting the path based on real-time sensor data to ensure smooth obstacle avoidance and precise path tracking. Data was recorded throughout the entire process to analyze the stability and real-time dynamic obstacle avoidance capabilities of the R-TEB algorithm.

During the test, the robot traveled along the predetermined target path while continuously recording changes in its linear and angular velocities. The robot maintained a relatively stable linear velocity during linear path segments. However, during obstacle avoidance or turning maneuvers, the angular velocity adjusted accordingly to ensure smooth path transitions.

As shown in Figures 8 and 9, compared to the traditional TEB algorithm, the R-TEB algorithm effectively reduces abrupt changes in linear and angular velocities by introducing curvature optimization and velocity smoothing constraints. This results in smoother velocity adjustments, thereby enhancing the stability of trajectory tracking.

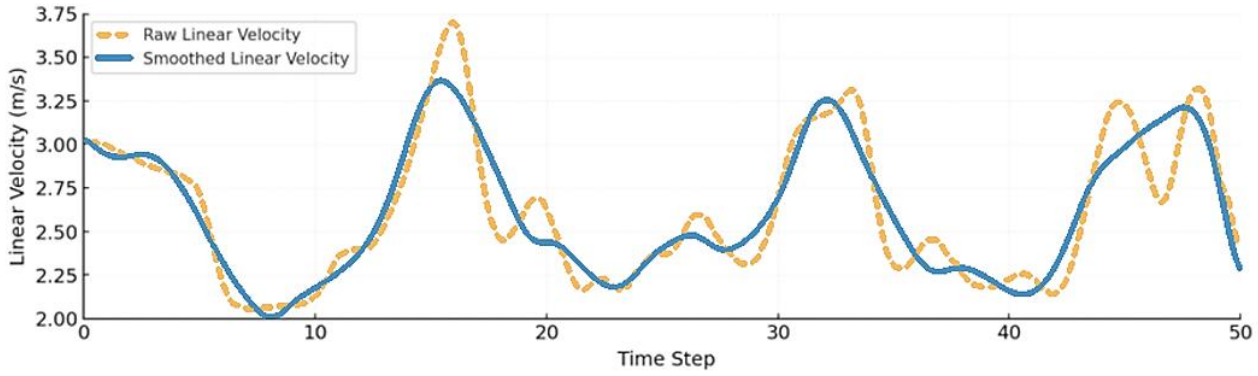

Fig. 8 –Linear Speed Comparison Between TEB and R-TEB Algorithms

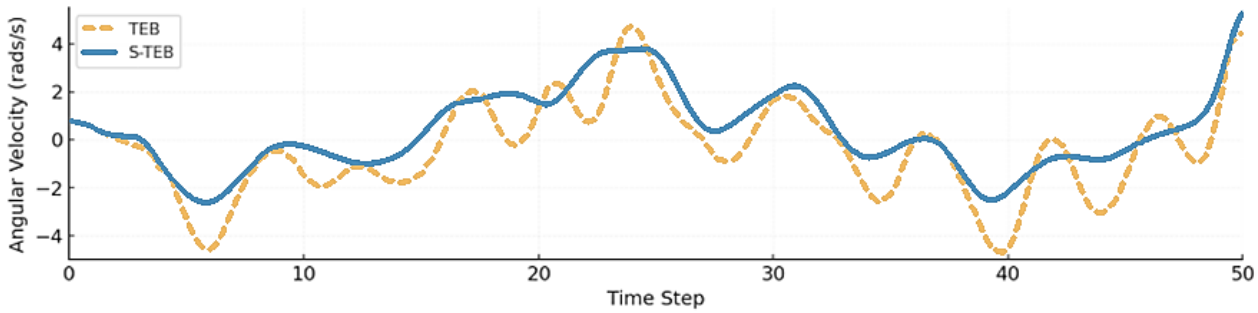

Fig. 9 –Comparison of Angular Velocity Between TEB and R-TEB Algorithms

As shown in Figure 10, under identical obstacle avoidance conditions, the robot's actual travel trajectories were recorded using RTK positioning while operating with the R-TEB and TEB algorithms. The experimental results reveal that the path planned by the TEB algorithm exhibits significant curvature changes, causing the robot to deviate from the planned trajectory during actual operation. This deviation is particularly pronounced at obstacle-avoidance turns. In contrast, the R-TEB algorithm, which optimizes curvature, enables the robot to follow the planned path more smoothly. The deviation between the actual trajectory and the theoretical path is smaller, resulting in superior path tracking performance.

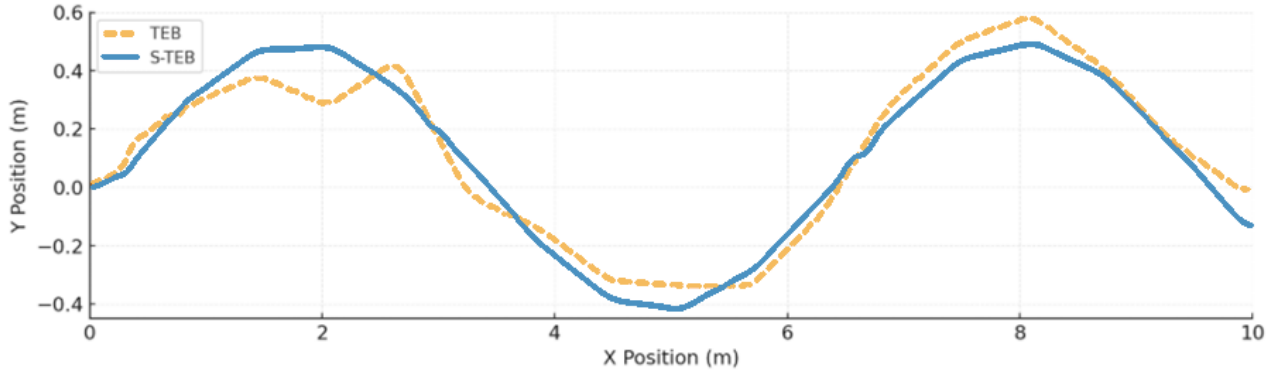

Fig. 10 –Actual Trajectories of TEB and R-TEB Algorithms

Table 1

Trajectory Tracking Test RTK Output Absolute Lateral Deviation Statistics (Unit: meters)

Test	TEB average	TEB Standard deviation	TEB RMSE	R-TEB average	R-TEB Standard deviation	R-TEB RMSE
1	0.017	0.018	0.017	0.019	0.015	0.020
2	0.019	0.019	0.019	0.018	0.016	0.019
3	0.020	0.022	0.022	0.019	0.018	0.019
4	0.021	0.018	0.018	0.017	0.017	0.016
5	0.018	0.021	0.016	0.018	0.018	0.020
6	0.017	0.018	0.019	0.019	0.015	0.017
7	0.019	0.021	0.025	0.017	0.011	0.016
8	0.019	0.019	0.019	0.019	0.018	0.020
9	0.024	0.019	0.023	0.018	0.019	0.022
10	0.021	0.020	0.012	0.018	0.016	0.020

The tracking performance of the two algorithms was evaluated by operating the robot at a constant speed of 10 km/h. For both trials, the initial position deviation and heading deviation were set to zero (with the initial position deviation limited to a maximum of 3 cm and the heading deviation restricted to a maximum of 1.5°). A GNSS-RTK device with a 10-Hz sampling frequency was employed to record the vehicle's position and heading information. The calculation of relative deviations was achieved by means of data analysis from multiple test runs.

Table 1 presents statistical results for absolute lateral deviations output by RTK, derived from repeated obstacle-avoidance tracking tests using multiple sets of R-TEB and TEB algorithms within the vineyard. Using RMSE (Root Mean Square Error) as the metric for prediction accuracy, the R-TEB algorithm yielded values between 0.016–0.022 m, while the TEB algorithm produced values between 0.012–0.025 m. This further demonstrates the R-TEB algorithm's superior tracking capability in practical applications.

CONCLUSIONS

The experimental results demonstrate that this vineyard robot exhibits optimized path planning and tracking accuracy under various experimental conditions. The system has demonstrated its efficacy in effectively addressing challenges posed by complex terrain and dynamic environments, thereby validating its practical application potential in vineyard scenarios. This finding is of considerable significance for the vigorous advancement of smart agriculture and the empowerment of agricultural modernization in the future.

REFERENCES

[1] Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*. 4(1), 23-33.

[2] Gao P., Jiang J., Song J. (2021). Fruit Tree Model Reconstruction Method for Robotic Canopy Volume Measurement in Orchards. *IEEE Access.* vol. 9, pp: 156246 – 156259.

[3] Jiang H., Pi J., Li A., Yin C. (2022). Dynamic Local Path Planning for Intelligent Vehicles Based on Sampling Area Point Discrete and Quadratic Programming. *IEEE Access*, vol. 10, pp.70279-70294.

[4] Lv P., Li Z. (2025). Research on Yolov5-Based Visual Slam Optimisation Method in Farm Depot Environment. *INMATEH-Agricultural Engineering*. 75(1). 83-94. DOI: https://doi.org/10.35633/inmateh-75-07

[5] Nie F., Zhang W., Wang Yl. (2021). A forest 3-d lidar slam system for rubber-tapping robot based on trunk center atlas. *IEEE/ASME Trans. on Mechatronics*. 27(5): 2623-2633.

[6] Pire T., Fischer T., Castro G., De Cristóforis P., Civera J., Berlles J. (2017). S-PTAM: Stereo parallel tracking and mapping. *Robot. Auton. Syst*. 93, 27–42.

[7] Qin Z., Wang H., Lv P. (2024). Research on Simultaneous Localization and Mapping Method for Orchards Based on Scan Context and NDT-ICP Fusion Scheme. *INMATEH-Agricultural Engineering*. 73(2). 636-646. DOI: https://doi.org/10.35633/inmateh-73-54

[8] Qiu Z., Zhao N., Zhou L., Wang M., Yang L., Fang H., Liu Y. (2020). Vision-based moving obstacle detection and tracking in paddy field using improved yolov3 and deep SORT. *Sensors*, 20(15), 4082.

[9] Rösmann C., Hoffmann F., Bertram T. (2015). Timed-elastic-bands for time-optimal point-to-point nonlinear model predictive control. In *Proceedings of the 2015 European Control Conference (ECC)*, Linz, Austria, 15–17 July. 3352–3357.

[10] Shan T., Englot B. (2018). LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 4758-4765.

[11] Wu, J., Ma, X., Peng, T., Wang, H. (2021). An improved timed elastic band (TEB) algorithm of autonomous ground vehicle (AGV) in complex environment. *Sensors*, 21(24), 8312.

[12] Xue G., Li R., Zhang Z. (2023). Research Status and Development Trend of SLAM Algorithm Based on 3D LiDAR (基于 3D 激光雷达的 SLAM 算法研究现状与发展趋势). *Information and Control*, vol. 52, Issue (01): 18-36.

[13] Yang, L., Xu, Y., Liang, Y., Qin, J., Li, Y., Wang, X., Wu, C. (2022). Extraction of straight field roads between farmlands based on agricultural vehicle-mounted LiDAR. *International Journal of Agricultural and Biological Engineering*, 15(5), 155-162.

[14] Yang, J., Chung, S. J., Hutchinson, S., Johnson, D., Kise, M. (2015). Omnidirectional-vision-based estimation for containment detection of a robotic mower. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 6344-6351.

[15] Zhou K., Leck Jensen A., Sørensen C G l. (2014). Agricultural operations planning in fields with multiple obstacle areas. *Computers and Electronics in Agriculture*. 109, 12-22.

[16] Zhang Y., Mo Z., Tian H. (2024). Path planning algorithm of agricultural robot based on improved APF-FMT*. *Journal of South China Agricultural University*. 45(3): 408-415.

[17] Zhou Z., Cao J., Di S. (2021). Overview of 3D LiDAR SLAM algorithms (3D 激光雷达 SLAM 算法综述). *Chinese Journal of Scientific Instrument*, vol. 42, Issue (09): 13-27.

[18] Zhang J., Singh S. (2014) LOAM: Lidar odometry and mapping in real-time. Robotics: *Science and Systems Conference*. 2(9): 1-9.

[19] Zhang, X., Zhu, T., Du, L., Hu, Y., Liu, H. (2022). Local path planning of autonomous vehicle based on an improved heuristic Bi-RRT algorithm in dynamic obstacle avoidance environment. *Sensors*, 22(20), 7968.

[20] Zhong, X., Tian, J., Hu, H., Peng, X. (2020). Hybrid path planning based on safe A* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment. *Journal of Intelligent & Robotic Systems*, 99(1), 65-77.

[21] Zhou, X., Wang, Z., Ye, H., Xu, C., Gao, F. (2020). Ego-planner: An ESDF-free gradient-based local planner for quadrotors. *IEEE Robotics and Automation Letters*, 6(2), 478-485.