126. Optimizing autonomous robot movement for data collection in precision horticulture

B. Ambrus*, G. Teschner, A.J. Kovács, M. Neményi and A. Nyéki
*Széchenyi István University, Albert Kázmér Faculty of Mosonmagyaróvár, Department of Biosystems and Precision Technology, Vár 2., Mosonmagyaróvár 9200, Hungary; * ambrus.balint@sze.hu*

Abstract

This study investigates the autonomous navigation capabilities of a custom-designed agricultural robot, focusing on its path planning, motion control, and performance evaluation. The robot integrates advanced systems, including a GNSS receiver with RTK corrections, Cube Orange autopilot and PID (proportional integral derivative) controllers for precise navigation. Field tests involved trajectory mapping between defined points, with data analysis revealing the relationships between angular deviations, motor speed differentials and positional accuracy. Results demonstrated strong correlations between angular velocity and deviations, highlighting areas for control system optimization. The findings aim to improve autonomous navigation frameworks in agricultural applications, promoting enhanced efficiency and precision in field operations.

Keywords: agricultural-robot development, Cube Orange, GNSS RTK, IMU, PID control

Introduction

Intelligent robots are becoming increasingly prevalent in agriculture, operating in diverse and dynamic environments to perform tasks such as data collection and intervention. To achieve autonomous functionality, mobile platforms must seamlessly execute critical tasks such as localization, routing, motion control and mapping (Ambrus *et al.*, 2024). While area coverage planning in agriculture is less advanced than point-to-point route planning, agricultural applications typically require full area coverage, whereas precision agriculture demands point-to-point navigation (Chakraborty *et al.*, 2022). The successful implementation of precision agriculture (PA) heavily relies on unmanned ground vehicles (UGVs), which play a crucial role in enhancing agricultural efficiency. UGVs help optimize fertilizer usage and enable precise weed control, contributing to more effective and sustainable farming (Kim *et al.*, 2019). The productivity of farming and crop yields per unit area are improving due to the division of labor and cooperation among multi-robot systems (MRS). Path planning is one of the most vital and integrated tasks for navigating UGVs. The vehicle or robot must generate a path between predefined target locations while avoiding collisions with obstacles in order to navigate autonomously (Ghaleb *et al.*, 2017).
This task of path planning can be divided into two main categories:

1. global route planning and local trajectory planning depending on the environmental data used to compute the optimal path. Global path planning focuses on identifying the most efficient route using a global geographical map,
2. local trajectory planning uses real-time sensor data from the environment to generate a collision-free path (Peralta *et al.*, 2020).

Once the path has been determined, the robot follows the course generated by the path planning algorithm. The Cube Orange flight controller is an excellent tool for motion planning in robotics. This advanced open-source autopilot system is designed for a range of vehicles, including drones, ground robots and aquatic vehicles. The Cube Orange is equipped with a powerful Cortex H7 processor and triple-redundant IMUs (inertial measurement units), ensuring reliability and stability during autonomous operations (Meier *et al.*, 2015). The controller mentioned above was used by Eredits *et*

al. (2023), who implemented UAV and UGV cooperation in the application, with the Cube Orange also handling the control of the UGV.
The authors developed the UGV motion control based on the Cube Orange controller, complemented by a PD controller designed to regulate the motor speeds depending on the target position to be reached. This article illustrates the accuracy of the motion and the deviations from the intended path based on this control strategy. The goal is to integrate and examine the well-established control strategy in the case of a custom-designed agricultural robot.

Materials and methods

As part of the development process, a global navigation satellite system (GNSS) receiver with real-time kinematic (RTK) corrections was integrated using a compact, energy-efficient mobile base station. This setup combined precise GNSS-based localization with RTK correction, advanced motion and trajectory planning, and state-of-the-art proportional-integral-derivative (PID) controllers, enabling the robot to navigate autonomously in complex and dynamic outdoor environments. The system is capable of real-time adjustments, ensuring that the robot remains on course even in the face of unexpected obstacles or terrain variations. Additionally, the integration of remote communication via 5G networks, MQTT messaging, and MAVLink protocols, enhances the system's adaptability and operational efficiency.

Navigation system overview
The robot uses a Cube Orange controller that supports both ArduPilot and PX4 firmware, offering robust functionality for controlling various types of vehicles, including VTOL drones, helicopters, multicopters and also UGVs (robots). The Cube Orange processes data from these sensors to calculate the robot position and heading, enabling precise navigation. The GNSS system is enhanced with RTK technology, which significantly improves positioning accuracy. The RTK system consists of two key modules: a base station that provides positioning corrections and a rover mounted on the robot. The base station collects real-time positioning data and sends correction signals (RTCM) to the rover, allowing it to correct its position with high precision. The rover has two RTK states: RTK FIXED and RTK FLOAT. RTK FLOAT is less accurate than RTK FIXED and not suitable for centimeter-level positioning. By default, the algorithm always starts in the FLOAT state, but under the right conditions and time, the RTK FIXED state can be reached. The RTK FIXED state indicates that the RTK algorithm has calculated the rover's precise position using the real-time correction messages, version 3 (RTCM3) correction data received from the base station. The data collected by the rover is sent to the robot platform control unit via the PX4 autopilot system (Moeller *et al.*, 2020). The navigation system is supported by a custom-built mobile base station. The base station is equipped with a battery pack, so no external power source is required. This device connects to its own virtual private network (VPN) server (Ezra *et al.*, 2022). The receiver is connected to a Raspberry Pi 4, which runs the software controlling the robot movement on the Raspbian operating system.

Communication system
The Cube Orange controller is the central hub for managing position sensor data, while a Raspberry-based motor controller handles the moving. Communication between the Cube Orange and the robot's software is facilitated by MAVLink messages, a communication protocol commonly used in unmanned vehicles. The system also supports remote monitoring via QGroundControl, a map-based interface that communicates with the Cube Orange. Data from the RTK base station is also transmitted using MAVLink messages, allowing real-time position updates.

Trajectory and motion planning

Autonomous navigation involves both trajectory planning and motion planning. Trajectory planning focuses on defining the optimal route from the start to the target, considering known obstacles. Motion planning deals with controlling the robot's movement along this path, adjusting speed and steering in real time. It is divided into two main parts:

1. Global Trajectory Planning: Defines the overall path from start to target, considering known obstacles and navigable routes.
2. Local Trajectory Planning: Deals with unexpected obstacles that appear during the robot movement, adjusting the path in real-time.

PID motion planner

The robot uses a PID controller to regulate motion and ensure accurate navigation.

The motion planner of the custom-designed agricultural robot was developed in Python language and runs on a Raspberry Pi 4 computer, which controls the device.

The system continuously adjusts the robot's speed and steering based on two main inputs;

1. current position and orientation: provided by the localization module,
2. trajectory planner: supplies the next target position and desired orientation.

Using these inputs, the motion planner calculates the necessary linear and angular velocities to minimize errors and keep the robot on track (Figure 1).

The controller adjusts both linear velocity (speed) and angular velocity (turning rate) to ensure smooth and accurate navigation, with specific focus on minimizing deviations from the planned trajectory. As the robot navigates through varied terrain, it must constantly adjust its movement to stay on course.

- Linear velocity control: determines the robot's forward speed based on the distance to the target. The robot reduces its speed as it approaches the target to avoid overshooting.
- Angular velocity control: regulates the robot's turning speed to ensure it aligns with the target orientation. This component uses a PD controller to minimize errors in angular orientation, preventing overshooting or oscillation as the robot adjusts its heading.

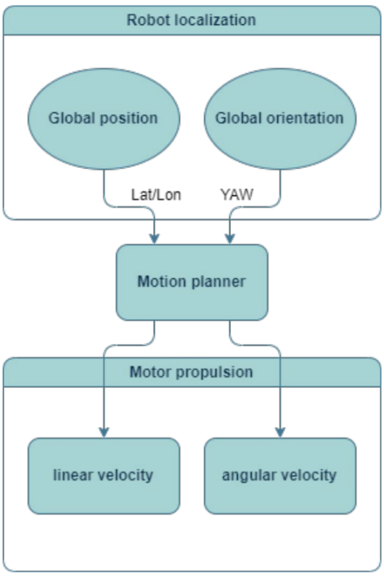

Figure 1. Localization system.

A key aspect of the system is to minimize errors between the robot orientation (measured by the IMU) and the target heading, which may be slightly different from the GNSS-based heading. The robot compensates for these discrepancies by applying an offset value derived from previous position measurements.

Field testing configuration
The field measurement simulates navigation between plant rows, during which two RTK-accurate points were recorded. The designed system was tested with 10 repetitions between these two points, collecting data on the route. This data included GNSS points recorded during movement, angular deviations calculated from the target coordinates and the robot's orientation coordinates, as well as angular velocity determined by the PD motion planner and the motor rotation speeds set by these values.

Results

The data collected during the measurements were analyzed using QGIS and SPSS software. This facilitated the visual representation of localization points characterizing the robot movements on a map, alongside statistical evaluations and correlation analysis of the recorded parameters. Approximately 16,000 records were gathered across 10 repetitions (Figure 2).

Distance errors and angular deviations
Analysis revealed the distribution of distance errors along both the outbound and return paths. Distance errors were similarly distributed but more pronounced on the outbound path. The mean error angle for the outbound path was –0.15° with a standard deviation of 4.22°, while the return path exhibited a mean error angle of –1.41° and a larger standard deviation of 18.67°, suggesting higher variability during the return journey.
Figure 3 demonstrates the distribution of error angles, highlighting that the characteristic values remained within a range of a few degrees without significant outliers. Larger deviations, observed at the start of the motion, were linked to the robot's initial rotation toward its target.

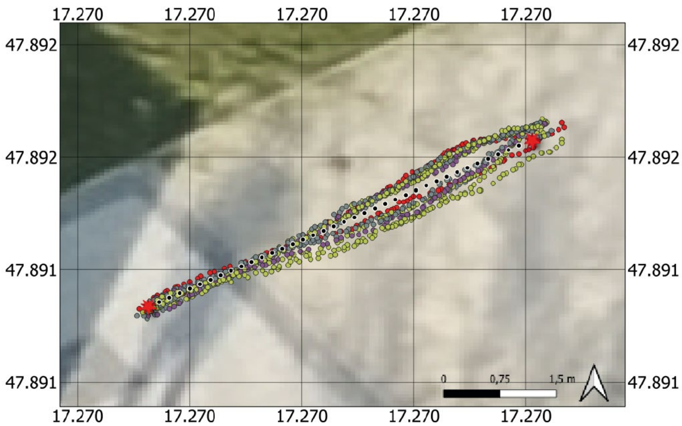

Figure 2. The visualization of motion based on ideal (black dots) and measured data (color dots), in each iteration.

Relationship between angular velocity and error angle
Figure 4 represents a strong positive correlation (R^2=0.94) between angular velocity and angular deviation. This indicates that as the angular deviation increased, the control system responded with proportionally higher angular velocity corrections.

Motor speed difference and angle deviation
Motor speeds, regulated on a 0–100 scale, were used to compute the speed difference parameter. As shown in Figure 5, a strong negative correlation (R^2=–0.844) exists between the motor speed difference and angular deviation. This suggests that larger speed differences correspond to reduced angular deviations, and vice versa. However, the negative correlation highlights potential control discrepancies affecting motor synchronization.

Statistical deviations from the ideal trajectory
Using QGIS, deviations from the ideal straight-line trajectory between two points were computed. The connecting line was segmented into 20 equal parts, and the distances of actual path points from these references were calculated using the least squares method. Additionally, the statistical indicators of deviations from the ideal (straight-line) movement between the two points were determined using QGIS software. To achieve this, the line connecting the starting and endpoint

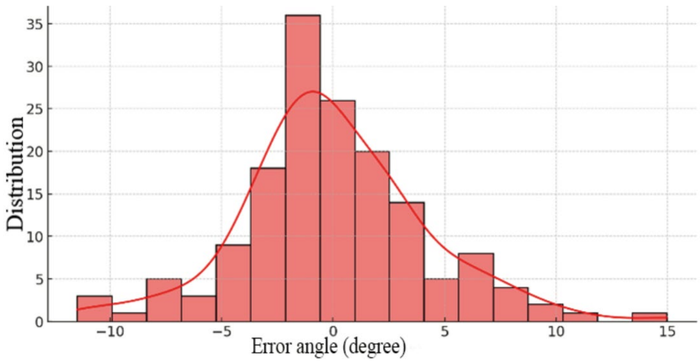

Figure 3. The distribution of the error angle.

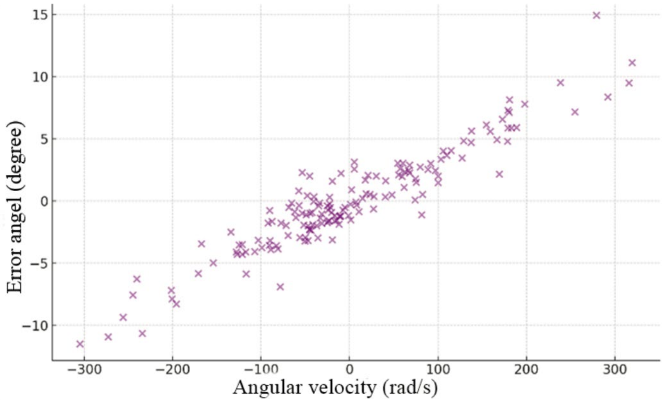

Figure 4. The relationship between angular velocity and error angle.

was divided into 37 equal segments, and the geographic coordinates for these division points were created. The results, summarized in Table 1, show average deviations and their standard deviations for the 10 iteration. While deviations were generally minor, variations indicate the need for further tuning of motion control systems.

Discussion

This study provided an in-depth analysis of the robot movement and orientation patterns, focusing on the relationship between angular deviations and motor speeds to evaluate navigation and control accuracy. The findings revealed that distance errors and angular deviations varied between the outbound and return paths. Specifically, the outbound path displayed larger errors, likely due to initial inaccuracies in motor control and trajectory planning. These errors propagated along the return path, although to a lesser extent, as the robot did not return exactly to the starting point.

The distribution and scatter plots indicate that significant angular errors were more frequent at certain intervals, suggesting potential motor control issues, such as track misalignment, improper PID control calibration, or wheel slippage. The stronger deviations observed during the outbound journey may have compounded due to challenges during initial rotation and trajectory alignment. Conversely, smaller errors on the return path might indicate compensatory adjustments made by the control system after the initial errors were introduced.

A strong positive correlation (R^2=0.94) between angular velocity and angular deviation demonstrates the robot›s ability to dynamically respond to deviations by increasing corrective angular velocity. However, the strong negative correlation (R^2=–0.844) between motor speed differences and

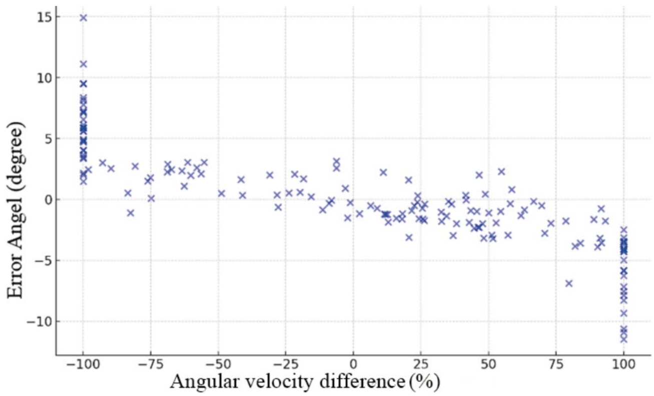

Figure 5. The relationship between speed difference and error angle.

Table 1. Statistical summary of deviations from the ideal trajectory.

	Iter.1	Iter.2	Iter.3	Iter.4	Iter.5	Iter.6	Iter.7	Iter.8	Iter.9	Iter.10	Average
Average	0.090	0.125	0.077	0.091	0.117	0.093	0.092	0.086	0.085	0.131	0.099
SD	0.059	0.073	0.041	0.049	0.076	0.050	0.049	0.056	0.050	0.074	0.058
Min	0.011	0.004	0.006	0.003	0.006	0.007	0.006	0.001	0.003	0.010	0.006
Max	0.282	0.254	0.181	0.003	0.267	0.203	0.252	0.204	0.191	0.253	0.209
Median	0.072	0.122	0.074	0.096	0.125	0.087	0.087	0.084	0.086	0.144	0.098

angle deviations highlights the need for better calibration to minimize inconsistencies in motor synchronization. These findings emphasize the importance of refining motor control parameters to enhance trajectory accuracy and reduce deviations.
Future investigations should focus on addressing initial measurement errors and optimizing PID tuning to improve trajectory stability. Additionally, exploring advanced adaptive control strategies may help mitigate errors introduced during the robot's movement.

Conclusions

In this article, a simple and effective autonomous control method for UGVs is proposed. The position of the UAV is determined by an Orange Cube. The autonomous control system was implemented using Python programming. The simplicity of the proposed system allows the UGV to move autonomously along the designated path, demonstrating its efficiency in the application of robots in practical closed farming technologies. These findings contribute to the development of more precise navigation systems, supporting the creation of stable and accurate control frameworks for autonomous agricultural robots. The proposed system demonstrated promising performance in handling complex outdoor environments, providing a foundation for further refinements in trajectory control and motor synchronization to enhance operational efficiency and reliability.

Acknowledgements

The research was carried out by the "Precision Bioengineering Research Group", supported by the "Széchenyi István University Foundation". The research presented in this paper was funded by the "Piaci KFI (2020-1.1.2-PIACI-KFI-2021-00182)" National Research, Development and Innovation Office project.

References

Ambrus, B., Teschner, G., Kovács, A.J., Neményi, M., Helyes, L., Pék, Z., Takács, S., Alahmad, T., & Nyéki, A. (2024). Field-grown tomato yield estimation using point cloud segmentation with 3D shaping and RGB pictures from a field robot and digital single lens reflex cameras. Heliyon, 10(20), e37997. https://doi.org/10.1016/j.heliyon.2024.e37997

Eredits, D., Krecht, R., & Boros, N. (2023). GPS based navigation and mobile base station of a mobile robot platform. Chemical Engineering Transactions, 107, 385–390. https://doi.org/10.3303/CET23107065

Ezra, P.J., Misra ,S., Agrawal, A., Oluranti, J., Maskeliunas, R., & Damasevicius, R. (2022). Secured communication using virtual private network (VPN). In: K. Khanna, V.V. Estrela, & J.J.P.C. Rodrigues (Eds), Cyber Security and Digital Forensics. Lecture Notes on Data Engineering and Communications Technologies, 73. Springer, Singapore. https://doi.org/10.1007/978-981-16-3961-6_27

Chakraborty, S., Elangovan, D., Govindarajan, P.L., ELnaggar, M.F., Alrashed, M.M., & Kamel, S. (2022). A comprehensive review of path planning for agricultural ground robots. Sustainability, 14(15), 9156. https://doi.org/10.3390/su14159156

Kim, J., Kim, S., Ju, C., & Son, H.I. (2019). Unmanned aerial vehicles in agriculture: a review of perspective of platform, control, and applications, IEEE Access, 7, 105100–105115.

Ghaleb, F.A., Zainal, A., Rassam, M.A., & Abraham, A. (2017). Improved vehicle positioning algorithm using enhanced innovation-based adaptive kalman filter. Pervasive and Mobile Computing, 40, 139–155.

Moeller, R., Deemyad, T., & Sebastian, A., (2020). Autonomous navigation of an agricultural robot using RTK GPS and Pixhawk. In The 2020 Intermountain Engineering, Technology and Computing (IETC). IEEE, pp 1–6, https://doi.org/10.1109/IETC47856.2020.9249176.

Peralta, F., Arzamendia, M., Gregor, D., Reina, D.G., & Toral, S. (2020). A comparison of local path planning techniques of autonomous surface vehicles for monitoring applications: the Ypacarai Lake case-study, Sensors, 20, 1488.

Meier, L., Honegger, D., & Pollefeys, M. (2015). PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms, In IEEE International Conference on Robotics and Automation (ICRA), pp. 6235–6240.