agriculture

MDPI

Article

Design, Optimization, and Field Evaluation of an Automatic Steering System for Agricultural Tractors Using Metaheuristic PID Tuning

Ali Karamolachab [1], Saman Abdanan Mehdizadeh [1,*] and Yiannis Ampatzidis [2,*]

[1] Department of Mechanics of Biosystems Engineering, Faculty of Agricultural and Rural Development, Agricultural Sciences and Natural Resources University of Khuzestan, Ahvaz 63417-73637, Iran; ali7chab@gmail.com

[2] Agricultural and Biological Engineering Department, Southwest Florida Research and Education Center, IFAS, University of Florida, 2685 SR 29 North, Immokalee, FL 34142, USA

* Correspondence: s.abdanan@asnrukh.ac.ir (S.A.M.); i.ampatzidis@ufl.edu (Y.A.)

Abstract

This paper presents the design and field evaluation of a low-cost automatic steering system for agricultural tractors. The system employs a PID controller whose gains are tuned using a metaheuristic optimization method. Core hardware includes an ESP32 microcontroller, an MPU9250 inertial measurement unit, a GPS module, and a servo motor for closed-loop yaw angle control, with a complementary filter fusing gyroscope and magnetometer data for robust heading estimation. Nine optimization algorithms were systematically compared: Grid Search, Random Search, Bayesian Optimization, Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), Moth-Flame Optimization (MFO), Sine Cosine Algorithm (SCA), Whale Optimization Algorithm (WOA), and Salp Swarm Algorithm (SSA). A cost function combining overshoot and settling time was used. Step response analysis showed that WOA achieved the best performance, with an integral absolute error of 6.31°·s, a settling time of 2.15 s, and a minimal overshoot of 0.08°. In field tests on asphalt and farmland, the WOA-tuned system reduced lateral deviation by 69% (from 12.4 cm to 3.8 cm) and 67% (from 18.7 cm to 6.2 cm), respectively, compared to manual steering. Repeated-measures ANOVA and paired *t*-tests confirmed statistically significant improvements ($p < 0.001$) with large effect sizes (Cohen's d > 2.7). The core components cost under $150 USD. The study offers a reproducible pipeline for comparative metaheuristic evaluation in agricultural vehicle guidance.

Keywords: yaw angle control; sensor fusion; precision agriculture; PID tuning; metaheuristic optimization

Check for updates

Academic Editors: Bruno Franceschetti and Marco Bietresato

Received: 11 April 2026
Revised: 29 April 2026
Accepted: 30 April 2026
Published: 3 May 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Precision agriculture has emerged as a critical paradigm for enhancing agricultural productivity while minimizing input costs and environmental impacts. By leveraging advanced sensing, control, and automation technologies, precision farming enables optimized resource utilization and improved operational efficiency. Among the key technologies enabling this transformation, automatic steering systems for agricultural vehicles have gained particular attention due to their potential to reduce operator fatigue, improve field efficiency, and enhance the accuracy of planting, spraying, and harvesting operations [1].

The ability to maintain straight trajectories with minimal lateral deviation directly translates to reduced input overlap, fuel savings, and optimized crop yield. Field studies have demonstrated that automatic steering systems can achieve lateral errors below 5 cm under favorable conditions, with substantial reductions in operator workload and input waste [2].

The development of effective automatic steering systems presents several technical challenges. Accurate heading estimation typically requires the fusion of multiple sensors, including gyroscopes, accelerometers, magnetometers, and Global Positioning System (GPS) receivers. An Attitude Heading Reference System (AHRS) provides three-dimensional orientation (roll, pitch, and yaw) of a moving vehicle by integrating data from these inertial sensors. Gyroscopes provide high bandwidth angular rate measurements but suffer from integration drift over time due to residual bias and noise. This drift accumulation can cause significant heading errors if not properly compensated. Magnetometers measure the Earth's magnetic field and offer an absolute heading reference relative to magnetic north without long-term drift; however, they are susceptible to local magnetic disturbances and require careful calibration [3]. These complementary characteristics motivate the use of sensor fusion algorithms, such as complementary filters or Kalman filters, to obtain a responsive and drift-free heading estimate. The complementary filter, in particular, combines high-pass-filtered gyroscope measurements with low-pass-filtered magnetometer and accelerometer data, with the filter gain determining the trade-off between short-term responsiveness and long-term stability. For agricultural tractors, accurate heading estimation is crucial for autonomous navigation, especially in low-speed farmland operations where GPS signals may be degraded [4]. Furthermore, the steering control system must reject disturbances arising from variable soil conditions, terrain irregularities, and tire–soil interactions while maintaining responsive and stable closed-loop behavior.

Proportional Integral Derivative (PID) controllers remain the industry standard for steering control applications due to their simplicity, robustness, and interpretability. The PID control approach offers a control action determined as the weighted superposition of an error signal, its temporal integral, and its derivative, providing sufficient flexibility to achieve satisfactory performance when properly tuned [5,6]. However, the performance of a PID controller critically depends on the appropriate selection of its three gains (Kp, Ki, Kd). Conventional tuning methods, such as Ziegler–Nichols or trial-and-error manual tuning, often yield suboptimal performance, particularly under varying operating conditions. This limitation has motivated extensive research into automated PID tuning using optimization algorithms. Field tests of PID-based automatic steering systems for tractors have shown promising results, with offset errors below 6 cm at travel speeds of 1.0 m/s, confirming the feasibility of PID control for agricultural guidance applications [7,8].

Metaheuristic optimization algorithms have demonstrated considerable promise for PID tuning tasks. These algorithms are designed to efficiently explore complex, multimodal search spaces and identify near-optimal parameter combinations without requiring gradient information. A comprehensive survey of metaheuristic algorithms for PID controller tuning highlights the growing interest in applying nature-inspired optimization methods to control system design, with researchers increasingly comparing multiple algorithms under consistent objective functions [9]. Particle Swarm Optimization (PSO), inspired by the social behavior of bird flocking, has been widely applied due to its simplicity and effectiveness. The Grey Wolf Optimizer (GWO) mimics wolf-pack hunting behavior and has demonstrated competitive performance in controller tuning. Moth Flame Optimization (MFO) employs logarithmic spiral navigation mechanisms, while the Sine Cosine Algorithm (SCA) uses trigonometric functions for search space exploration. The Salp Swarm Algorithm (SSA) models chain-like swarm behavior inspired by the navigation of salps in oceans. Bayesian optimization offers a probabilistic, surrogate-based approach that is particularly

sample-efficient, building a Gaussian process model of the objective function to guide the search. The Whale Optimization Algorithm (WOA) is inspired by the bubble-net hunting strategy of humpback whales. WOA combines three main mechanisms: encircling prey, bubble-net attack (spiral updating), and prey search (random exploration), with a probability parameter that switches between strategies to maintain search diversity [5,10].

Despite these advances, several research gaps remain. First, comparative studies of multiple optimization algorithms specifically for agricultural tractor steering PID tuning are limited. While individual algorithms have been evaluated separately, systematic comparisons under identical experimental conditions are scarce. Second, most existing work evaluates controller performance solely through simulation or bench testing, with limited field validation under actual farming conditions. The dynamic disturbances present in agricultural environments, including variable soil resistance, wheel slip, and terrain irregularities, cannot be fully replicated in simulation. Third, the specific challenges of yaw control on variable soil surfaces, including the effects of wheel slip, soil resistance, and tire sinkage, are rarely addressed in the optimization-focused literature.

This paper addresses these gaps by presenting a complete automatic steering system for agricultural tractors, from hardware design and sensor fusion to control implementation and field evaluation. The specific contributions of this work are: (1) the design and construction of a low cost (<$150 USD for core electronic and mechanical components) automatic steering system using an ESP32 microcontroller, Micro-Electro-Mechanical Systems (MEMS) sensors, and servo motor actuation, with a belt and pulley transmission mechanism that preserves manual control capability; (2) the implementation of a complementary filter for robust yaw angle estimation, fusing gyroscope and magnetometer data to achieve both responsiveness and drift free operation; (3) a systematic comparison of nine optimization methods for PID gain tuning, including exhaustive search, Bayesian optimization, and six metaheuristic algorithms (PSO, GWO, MFO, SCA, WOA, SSA), using a composite cost function that balances overshoot and settling time; and (4) comprehensive field validation on both asphalt and farmland surfaces, with appropriate repeated-measures statistical analysis of lateral deviation reduction compared to manual steering, including effect size calculations (Cohen's d) to quantify practical significance. While metaheuristic algorithms have been applied to PID tuning in other domains, this work represents their first application to agricultural tractor yaw control with full hardware-in-the-loop validation.

2. Materials and Methods

2.1. Design and Construction of an Automatic Steering System

A schematic diagram of the proposed system is presented in Figure 1, showing its three core components, mechanical, electromechanical, and electrical, along with the subsystems.

The mechanical components of the automatic tractor steering system include the main steering wheel, the steering actuator motor, the motion transmission mechanism, and the belt-and-pulley assembly. In this design, the main steering wheel serves as the conventional interface between the driver and the steering mechanism. This system retains manual control capability for the driver even in automatic mode. This feature is not only crucial for the driver in essential situations, such as navigating obstacles or during emergencies, but also enables the combined use of automatic and manual modes and smooth transitions between them. The steering shaft, which is directly connected to the steering wheel, represents the initial point for torque application, whether by the automatic mechanism or the operator. The core of this system is the steering actuator motor, which generates the torque to rotate the steering mechanism. In this design, a servo motor (JX Servo, B70, Shantou Jixian Electronic Technology Co., Ltd., Shantou, China) is employed as the source of this force. To transmit the motor torque to the main steering shaft at an appropriate

transmission ratio, a belt-and-pulley mechanism (ratio 1:2.5) is used. The servo motor is mounted to the steering column housing using a custom aluminum bracket (6061-T6, 3 mm thickness) secured with M6 bolts. The larger pulley is attached to the steering shaft via a keyed hub with set screws to prevent slip. Belt tension is adjusted via a slotted motor mount with locking bolts, maintaining ~5 mm deflection under moderate finger pressure. The ESP32/IMU/GPS enclosure is mounted on the dashboard using vibration-damping rubber grommets. All wiring is routed through protective conduit and secured with cable ties to prevent entanglement. The emergency stop circuit uses a normally-closed pushbutton wired in series with the servo power relay, ensuring hardware-level cutoff independent of software state. Mechanical component dimensions are reported in millimeters (mm) where precision is required.

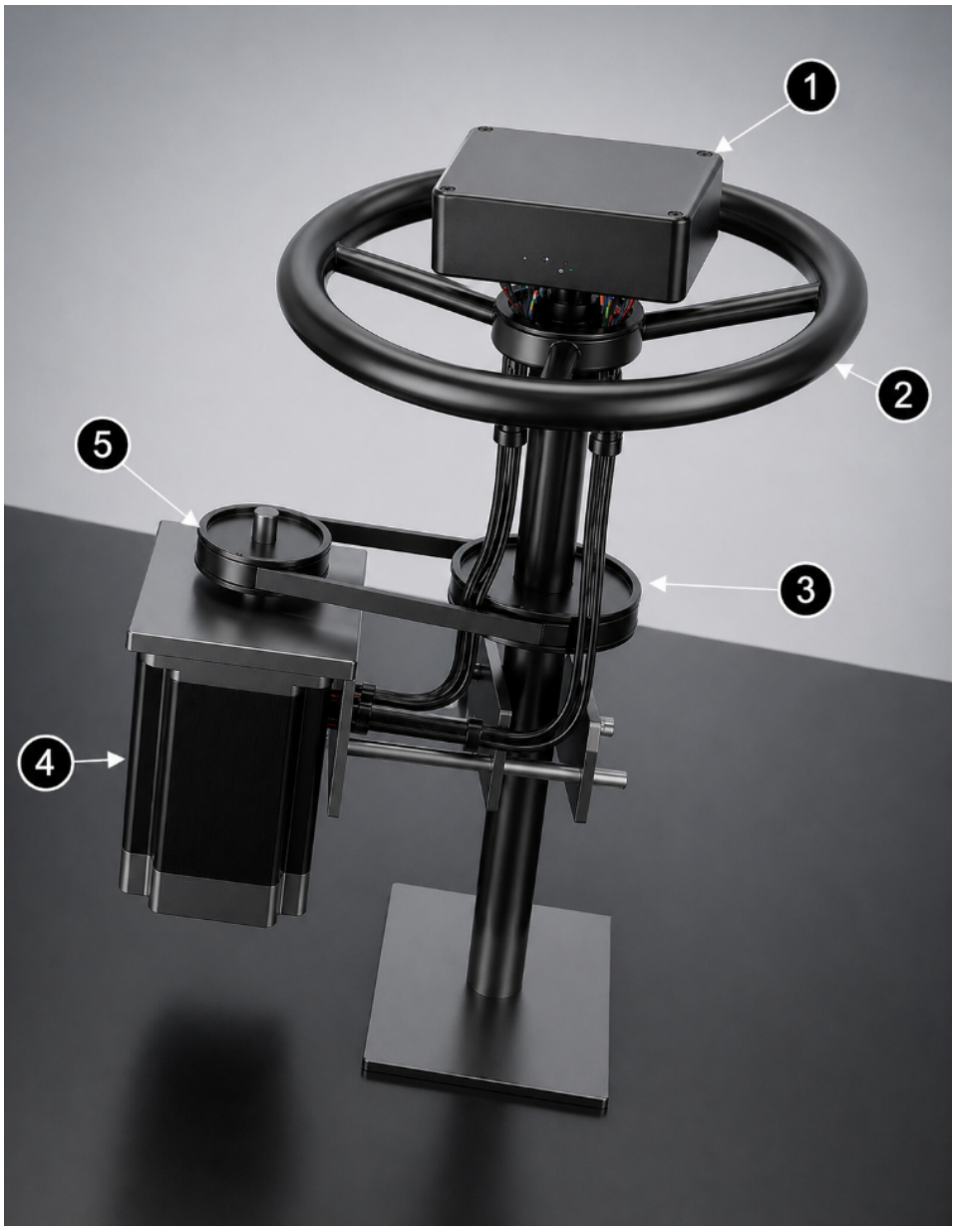


Figure 1. Schematic diagram of the automatic tractor guidance system: (1) controller box, (2) main steering wheel of the tractor, (3) belt, (4) servomotor, (5) pulley. The belt-and-pulley transmission (ratio 1:2.5) provides vibration damping, slip-limited overload protection, and modular retrofitting capability while preserving manual override (<0.8 Nm backdrive torque).

In this mechanism, a small pulley is mounted on the motor output shaft to provide the initial torque. A larger pulley is placed on an intermediate shaft, and the final force transmission is accomplished through the rotation of the belt between these two pulleys. Key advantages of this approach in agricultural contexts are: (1) inherent vibration damping reduces transmission of soil-induced shocks to the servo motor, extending actuator lifespan; (2) slip-limited torque transmission provides passive overload protection during unexpected obstacles, preventing gear damage; and (3) modular attachment enables retrofitting to existing tractors without permanent modification. Unlike direct-drive implementations, this configuration preserves the operator's ability to manually override the steering wheel with minimal backdrive torque (<0.8 Nm), a critical safety feature for field operations. The servo motor (JX B70, 46 kg·cm stall torque) was selected based on measured steering torque requirements representative of operating conditions, providing a 3.2× safety margin under worst-case soil resistance. Table 1 summarizes the torque characterization across conditions, confirming that the actuator output (servo stall torque × pulley ratio) exceeds peak field demands while maintaining low thermal load during continuous operation.

Table 1. Steering torque characterization and actuator safety margins under representative operating conditions.

Operating Condition	Measured Steering Torque (Nm)	Actuator Output Torque (Nm) *	Safety Margin
Asphalt, stationary	1.2 ± 0.3	11.3	9.4×
Farmland, dry soil	2.1 ± 0.5	11.3	5.4×
Farmland, wet soil	3.5 ± 0.8	11.3	3.2×
Obstacle encounter	4.1 ± 1.1	11.3	2.8×

* Note: Actuator output torque = servo stall torque (46 kg·cm ≈ 4.5 Nm) × belt–pulley ratio (2.5:1). Wet soil represents the design worst-case condition, yielding the governing 3.2× safety margin. Thermal testing over 45-min continuous runs at maximum load showed <12 °C temperature rise, confirming safe operating margins.

2.2. Hardware Components

The autonomous vehicle prototype (Figure 2) integrates an ESP32-WROOM-32D (Espressif Systems, Shanghai, China) microcontroller, an MPU9250 (TDK InvenSense, San Jose, CA, USA) inertial measurement unit (IMU), a GPS module, and a servo motor for steering actuation. The ESP32 serves as the central processing unit, handling sensor data acquisition, executing control algorithms, and communicating with a host computer for monitoring and parameter tuning, if needed. The MPU9250 is a nine-axis MEMS sensor that combines a three-axis gyroscope, three-axis accelerometer, and a three-axis magnetometer (AK8963) in a compact 3 mm × 3 mm package. It communicates via I^2C at 400 kHz. The integrated magnetometer provides an absolute heading reference using Earth's magnetic field, which is essential for bounding yaw drift over extended periods (e.g., 5+ hours of field operation). The gyroscope offers a programmable full-scale range (up to ±2000°/s) and low noise (3 mdps/$\sqrt{\text{Hz}}$), while the accelerometer (up to ±16 g) and magnetometer (up to ±4800 μT) support accurate attitude estimation.

A GPS module (NEO-6M, u-blox, Thalwil, Switzerland; sourced from China) is connected via a dedicated UART and provides position, velocity, and course over ground (COG) information. The GPS course serves as an additional absolute heading reference when the vehicle speed exceeds 1 m/s. Steering is accomplished by a servo motor (JX Servo, B70, Shantou Jixian Electronic Technology Co., Ltd., Shantou, China) controlled through a PWM signal from the ESP32. The servo's neutral position (90°) corresponds to straight motion; control signals are constrained to ±30° to prevent mechanical over-travel. Figure 2 shows the schematic diagram of the electrical circuit.

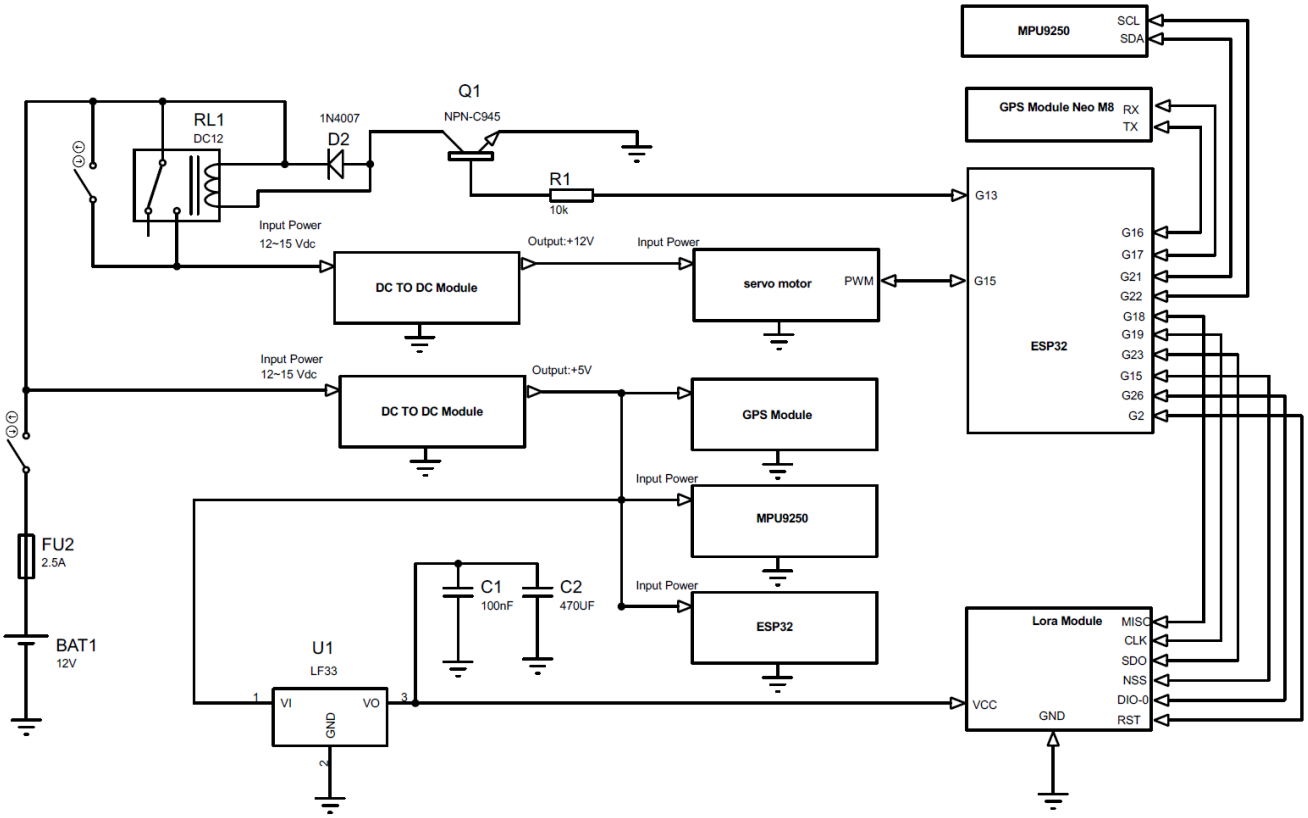


Figure 2. Schematic diagram of the control unit electrical circuit for the automatic steering system.

2.3. Yaw Angle Estimation

The vehicle operates in a line-based manner. At the beginning of each work line, the operator defines a desired heading (e.g., to follow a specific furrow or row). This desired yaw, ψ_{ref}, is set via the mobile app or a physical button and becomes the control setpoint. The vehicle then maintains this heading throughout the line, with the controller acting on the error between the desired and estimated yaw. Each line typically lasts less than one hour, after which the operator may set a new desired heading for the next line. This operational pattern bounds long-term drift because the setpoint is frequently re-initialized, but accurate heading estimation during each line remains critical for precise guidance.

2.3.1. Gyroscope-Based Integration

The primary sensor for tracking angular motion is the gyroscope's z-axis. After initial bias removal (averaging the first 100 samples), the angular rate is integrated to obtain a relative yaw angle (Equation (1)):

$$\psi_{gyro}(t) = \psi_{gyro}(t-1) + (\omega_z - bias)\Delta t \quad (1)$$

where $\psi_{\text{gyro}}(t)$ is the gyroscope-derived yaw angle (°), $\omega_z(\tau)$ is the angular rate about the vehicle's vertical (z) axis (°/s), Δt is the sampling interval (s), and bias is the estimated gyro bias (°/s) obtained by averaging the first 100 stationary samples.

The gyroscope provides excellent short-term accuracy and high-bandwidth response, making it ideal for capturing vehicle dynamics. However, integration drift accumulates over time due to residual bias and noise. Typical MEMS gyroscopes drift at rates of 1–5°/hour, which could cause noticeable heading errors by the end of a one-hour line if used alone.

2.3.2. Magnetometer Heading

The magnetometer measures the Earth's magnetic field and yields an absolute heading relative to magnetic north. Magnetometer calibration was performed using a two-step procedure: (i) a six-position hard-iron calibration to estimate bias offsets, followed by (ii) ellipsoid fitting for soft-iron correction using 2000 samples collected in a magnetically clean environment. The resulting calibration matrix is applied in real time. After hard-iron and soft-iron calibration, the heading is as follows (Equation (2)):

$$\psi_{mag}(t) = atan2(m_y, m_x) \quad (2)$$

where ψ_{mag} is the magnetometer-derived heading relative to magnetic north (°), *mx* and *my* are the calibrated horizontal magnetic components. The heading is adjusted for magnetic declination at the test location. A magnetic declination of +2.1° (east) for the test location (48.7176° E, 31.8202° N) is added to the magnetometer heading.

The magnetometer has no drift—it always provides the same absolute reference. However, it is susceptible to noise and local magnetic disturbances and cannot be used directly for feedback control without heavy filtering, which would introduce lag.

2.3.3. Sensor Fusion: Gyroscope + Magnetometer

The complementary filter (Equation (3)) fuses the gyroscope and magnetometer to estimate primary heading. GPS course-over-ground (COG) is incorporated as a secondary correction when vehicle speed > 1.0 m/s and GPS fix quality ≥ 3 (per NMEA standards), using a separate low-gain complementary stage (α_GPS = 0.01) to avoid introducing high-frequency noise. This hierarchical fusion strategy prioritizes short-term responsiveness (gyro) with medium-term drift correction (magnetometer) and long-term absolute reference (GPS), validated in preliminary tests showing < 0.5° RMS error versus RTK-GNSS reference:

$$\psi_{fused}(t) = \alpha \cdot \psi_{mag}(t) - (1 - \alpha)\psi_{gyro}(t) \quad (3)$$

where ψ_{fused} is the complementary filter fused yaw estimate, and α is the complementary filter gain (gyro weight). The coefficient α (typically 0.02–0.05) determines the extent to which the magnetometer corrects the gyro drift. The complementary filter gain $\alpha = 0.035$ was selected via a grid search over $\alpha \in [0.01, 0.10]$, minimizing the root-mean-square heading error against an RTK-GNSS reference during preliminary field tests. A small value ensures that the filter relies primarily on the smooth gyro prediction, while the magnetometer slowly pulls the estimate toward the true heading, eliminating accumulated error. This formulation makes explicit that the gyroscope drives the short-term dynamics via Equation (1), and the magnetometer provides long-term stability through Equation (2). The fused estimate $\psi_{fused}(t)$ is used as the feedback signal for the controllers described in Section 2.3.

2.4. Control System Architecture

A closed-loop feedback control system regulates the steering servo to maintain a desired yaw angle ψ_{ref}. The error signal is defined as follows (Equation (4)):

$$e(t) = \psi_{ref} - \psi_{fused}(t) \quad (4)$$

Then $e(t)$ is normalized to $[-180, 180)$ degrees. The controller computes a corrective action $u(t)$ which is then added to the servo's neutral position (90°) and constrained to the safe range [0, 180]. A discrete-time PID controller with anti-windup and derivative filtering was implemented for yaw angle regulation.

2.4.1. PID Controller

The discrete-time PID controller with anti-windup and derivative filtering is given by Equation (5) [11]:

$$u(t) = K_p e(t) + K_i \sum e(t)\Delta t + K_d LPF\left(\frac{e(t) - e(t-1)}{\Delta t}\right) \quad (5)$$

The integral term is clamped to $[-30/Ki\ 30/Ki]$ to prevent integral windup. The derivative term is passed through a first-order low-pass filter (LPF) with a coefficient of 0.9 to attenuate high-frequency noise [12]. The gains *Kp*, *Ki*, and *Kd* are the parameters to be tuned.

2.4.2. Timing and Actuation Constraints

The MPU9250 inertial measurement unit is sampled at 100 Hz (10 ms period) via I^2C, while the GPS module provides course-over-ground updates at 5 Hz via UART. The PID control law is executed at 100 Hz, synchronized with the IMU sampling interrupt; the computed control output is held constant between updates and applied to the servo via a 50 Hz PWM signal (20 ms period), which is the native update rate of the JX Servo B70. End-to-end latency—measured by toggling a GPIO pin at the start of the IMU read routine and capturing the corresponding PWM edge on a digital oscilloscope—was 14.2 ± 1.8 ms (mean ± SD, n = 50 trials), encompassing IMU read, complementary filtering, PID computation, and PWM update. This latency is well within the servo's 20 ms frame. It corresponds to a phase margin of >45° at the crossover frequency observed in bench frequency-response tests, confirming that timing delays do not compromise closed-loop stability.

The servo control signal is constrained to ±30° from the neutral position (PWM pulse widths of 0.5 ms to 2.5 ms; neutral at 1.5 ms) to prevent mechanical over-travel. During aggressive step responses (e.g., 0° → 30° yaw command), the controller briefly commanded the saturation limit; however, the servo's internal speed specification (0.10 s per 60° at 6 V) provides inherent rate limiting that smooths torque transients. No additional software rate limiting was implemented, as empirical testing showed that the native servo dynamics, combined with the derivative term in the PID controller, adequately suppressed high-frequency oscillations without introducing lag.

2.5. Optimization Methods

To obtain the best controller gains, multiple optimization strategies were employed: grid search, random search, Bayesian optimization, and six metaheuristic algorithms (PSO, GWO, MFO, SCA, WOA, SSA). All methods share the same cost function (see Section 2.6) and are executed offline by a Python 3.10.9 script that communicates with the ESP32 via a serial link. Each candidate's gain set was evaluated using a 10-s step response (0°→30°), with performance metrics computed over the full duration. Settling time definitions follow standard control practice: time to enter and remain within ±2% of the final value. Table 2 summarizes the hyperparameter settings, evaluation budgets, and protocol details, ensuring a fair comparison across all nine optimization methods.

To maintain experimental consistency, all algorithms searched within identical parameter bounds: proportional gain $K_p \in [0.5, 3.0]$, integral gain $K_i \in [0.1, 3.0]$, and derivative gain $K_d \in [0, 0.5]$. These bounds were established through prelimnary system identification and Ziegler–Nichols critical gain estimation ($K_{cr} \approx 2.1$), with pilot tests confirming that values outside these ranges resulted in either instability ($K_p > 3.0$, $K_i > 3.0$) or sluggish, underdamped behavior ($K_p < 0.5$, $K_i < 0.1$). The upper limit for K_d was set to 0.5 to prevent amplification of high-frequency sensor noise while still providing sufficient damping; this corresponds to a derivative time constant $T_d = K_d/K_p \leq 1.0$ s, aligning with the dominant

mechanical time constant of the steering actuator (~0.8 s). Stochastic algorithms were initialized with 10 independent random seeds to assess outcome variability, and all methods were constrained to an identical evaluation budget of 200 function calls to ensure computational fairness. Deterministic methods (Grid Search) were run to completion.

Table 2. Optimization algorithm hyperparameters and experimental protocol.

Method	Population/Points	Max Iterations/Evals	Stopping Criteria	Random Seeds
Grid Search	11^3 = 1331	1331	Exhaustive	1
Random Search	200	200	Budget-limited	10
Bayesian Opt.	50 initial + 150 adaptive	200	Max evaluations	10
PSO	30 particles	50 iterations	Max iterations	10
GWO	30 wolves	50 iterations	Max iterations	10
MFO	30 moths	50 iterations	Max iterations	10
SCA	30 agents	50 iterations	Max iterations	10
WOA	30 whales	50 iterations	Max iterations	10
SSA	30 salps	50 iterations	Max iterations	10

Note: All metaheuristics used tournament selection (size = 3), elitism (top 10% preserved), and identical cost function J = Mp + 0.5·ts. Random seeds: 10 independent runs per algorithm; results report mean ± SD. Hardware: ESP32 (240 MHz) with Python host via serial (115,200 baud). Fairness ensured by an identical evaluation budget (200 function evaluations) for stochastic methods; deterministic methods ran to completion.

2.5.1. Grid Search

Grid Search is a deterministic brute-force method. In this method, the parameter space is discretized into a predefined number of steps for each gain. All combinations are tested exhaustively. While this guarantees finding the optimum within the grid, the number of experiments grows exponentially with the number of parameters. The domain of each parameter x_i is discretized into N_i equally spaced values (Equation (6)) [13]:

$$\mathrm{x}_i^{(j)} = \mathrm{x}_i^{min} + j \cdot \frac{\mathrm{x}_i^{max} - \mathrm{x}_i^{min}}{N_i - 1}, \qquad \mathrm{j} = 0, 1, \ldots, N_I - 1. \tag{6}$$

The Cartesian product of these sets forms a finite grid of candidate points. The algorithm evaluates $J(\mathbf{x})$ at every grid point and selects the one with the smallest cost (Equation (7)) [13]:

$$\mathrm{x}^* = arg_{x \in Grid} \min J(x). \tag{7}$$

2.5.2. Random Search

Random Search samples N candidate solutions independently from a uniform distribution over the hyper-rectangle defined by the parameter bounds. A user-specified number of gain sets is drawn uniformly at random from the specified ranges. This method is considerably faster than grid search and often identifies near-optimal solutions with far fewer tests. For each parameter x_i (Equation (8)) [14], the following applies:

$$x_i^{(t)} \sim u(x_i^{min}, x_i^{max}), \qquad \mathrm{t} = 1,\ 2,\ \ldots, \mathrm{N}. \tag{8}$$

After all samples are evaluated, the best-found solution is as follows (Equation (9)) [14]:

$$\mathrm{x}^* = arg_{t=1\ldots N} \min J\left(x^t\right) \tag{9}$$

2.5.3. Bayesian Optimization

Bayesian optimization builds a probabilistic surrogate model (Gaussian process) of the unknown cost function $f(x) = J(x)$ and uses an acquisition function to select the next most promising gain set. The algorithm balances exploration (testing unknown regions) and exploitation (focusing on regions with low predicted cost). It consists of the following processes (Equations (10)–(12)) [15,16]:

- Gaussian Process (GP) surrogate: After observing data $D_{1:t} = \{ x_i, y_i \}_{i=1}^{t} = (with\, y_i = f(x_i)$, possibly noisy), a GP with mean function $\mu(x)$ and covariance kernel $k(x, x')$ provides the following posterior predictions:

$$\hat{f}(x) \sim GP(\mu_t(x), \sigma_t^2(x)) \tag{10}$$

  where $\mu_t(x)$ and σ_t^2 (x) are the posterior mean and variance.

- Acquisition Function (Expected Improvement):

$$\alpha_{EI}(x) = E\left[max\left(f_{best} - \hat{f}(x), 0\right)\right] = (f_{best} - \mu_t(x) - \xi)\Phi(z) + \sigma_t(x)\Phi(Z), \tag{11}$$

With $Z = \frac{F_{BEST} - \mu_t(X) - \xi}{\sigma_t(X)}$, where Φ and ϕ are the CDF and PDF of the standard normal, f_{best} is the current best observed value, and ξ balances exploration/exploitation. The next evaluation point is chosen by maximizing α_{EI}:

$$x_{t+1} = \arg max \alpha_{EI}(x) \tag{12}$$

2.5.4. Particle Swarm Optimization (PSO)

PSO is a population-based stochastic optimization technique inspired by the social behavior of bird flocking or fish schooling. A swarm of particles moves through the parameter space, each tracking its personal best position and the global best position found by the swarm. PSO maintains a swarm of $\boldsymbol{N}$ particles. Each particle i has a position $X_i(t)$ and a velocity v_i(t). It also remembers its personal best position $\boldsymbol{p}_i$ and the global best position **g** found by the entire swarm.

The velocity update combines three components: inertia, cognitive (pull toward personal best), and social (pull toward global best) (Equation (13)) [17]:

$$v_i(t+1) = wv_i(t) + c_1 r_1 (p_i - x_i(t)) + c_2 r_2 (g - x_i\,(t)) \tag{13}$$

where w is the inertia weight, c_1 and c_2 are acceleration coefficients, r_1 and r_2 [0, 1] are random numbers.

The position is then updated (Equation (14)) [18]:

$$\boldsymbol{x_i}(t+1) = \boldsymbol{x_i}(t) + \boldsymbol{v_i}(t+1) \tag{14}$$

Personal and global bests are updated whenever a better cost is found.

2.5.5. Grey Wolf Optimizer (Gwo)

GWO mimics the leadership hierarchy and hunting mechanism of grey wolves in nature. The algorithm maintains a population of wolves classified into four types: α (the best solution), β (the second-best), δ (the third-best), and ω (the remaining solutions). Hunting is guided by alpha, beta, and delta, with wolves updating their positions based on the locations of these leaders. The search process includes encircling, hunting, and attacking prey, with a control parameter **a** that decreases linearly as it transitions from exploration to exploitation. The three best solutions are designated as α, β, and δ. Other

wolves (ω) update their positions based on these three leaders. The encircling behavior is modeled as follows (Equation (15)) [19]:

$$D_\alpha = |C_1 \cdot X_\alpha(t) - X(t)|, \quad D_\beta = |C_2 \cdot X_\beta(t) - X(t)|, \quad D_\delta = |C_3 \cdot X_\delta(t) - X(t)|, \tag{15}$$

where $C_j = 2r_2$ with r_2 ∈ [0, 1] as random.

Then the position update for a wolf X is as follows (Equations (16) and (17)) [19]:

$$X_1 = X_\alpha - A_1 \cdot D_\alpha, \quad X_2 = X_\beta - A_2 \cdot D_\beta \quad , \quad X_3 = X_\delta - A_3 \cdot D_\delta \tag{16}$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \tag{17}$$

Here, $A_j = 2a \cdot r_1 - a$, where the components of a decrease linearly from 2 to 0 over iterations, and $r_1 \in [0, 1]$ is random. The vectors **A** and **C** help the wolves to explore and exploit.

2.5.6. Moth-Flame Optimization (MFO)

MFO is inspired by the navigation method of moths, which fly in a straight line by maintaining a fixed angle with respect to the moon (transverse orientation). In the algorithm, moths are candidate solutions that spiral around flames (accordingly, the best positions). Each moth updates its position using a logarithmic spiral function relative to a flame, with the number of flames decreasing over iterations to balance exploration and exploitation. This mechanism allows moths to explore the search space thoroughly while converging toward promising regions. Each moth M_i updates its position using a logarithmic spiral with respect to a flame F_j (Equation (18)) [20]:

$$M_i = F_j + D_{ij} \cdot e^{bt} \cdot cos(2\pi t) \tag{18}$$

where $D_{ij} = |F_j - M_i|$ is the distance, b is a constant defining the spiral shape, $t \in [r, 1]$ is a random number, with Y linearly decreasing from -1 to -2 over iterations.

The number of flames decreases adaptively (Equation (19)) [21]:

$$Flame_no = round\left(N - t \cdot \frac{N-1}{T}\right) \tag{19}$$

where N is the number of moths, and T is the maximum iterations. This mechanism forces moths to focus on the best flames as optimization progresses.

2.5.7. Sine Cosine Algorithm (SCA)

SCA creates multiple initial candidate solutions and fluctuates them outward or toward the best solution using mathematical models based on sine and cosine functions. The algorithm employs a set of random parameters to balance exploration and exploitation. For each search agent $\boldsymbol{X_i}$, the update rule is as follows (Equation (20)) [22]:

$$\boldsymbol{X_i}(t+1) = \begin{cases} X_i(t) + r_1 \sin(r_2) \cdot |r_3 P_i(t) - X_i(t)|, & r_4 < 0.5, \\ X_i(t) + r_1 \cos(r_2) \cdot |r_3 P_i(t) - X_i(t)|, & r_4 \geq 0.5, \end{cases} \tag{20}$$

where $P_i(t)$ is the best solution found (destination point), r_1 controls the next position's region (decreases linearly from a constant to 0), $r_2 \in [0, 2\pi]$ defines how far the movement should be towards or outwards from the destination, $r_3 \in [0, 2]$ gives a random weight to the destination, $r_4 \in [0, 1]$ switches between sine and cosine components.

The parameter $r_1 = 2 - 2t/T$ ensures a transition from exploration (large movement) to exploitation (fine-tuning). A parameter r_1 decreases linearly to guide the search from

global to local, while r_2, r_3, and r_4 introduce stochastic behavior through sine/cosine functions. The simple mathematical formulation makes SCA computationally efficient while maintaining good search capabilities [22].

2.5.8. Whale Optimization Algorithm (WOA)

WOA simulates the bubble-net hunting strategy of humpback whales. The algorithm comprises three main phases: encircling prey (moving toward the best solution), bubble-net attack (spiral updating), and prey search (random exploration). A probability parameter p switches between shrinking encircling and spiral models, while the control parameter A determines whether whales explore ($|A| \geq 1$) or exploit ($|A| < 1$). This combination of strategies enables effective global search and local refinement. The three mechanisms are as follows [23]:

- Encircling prey (when $p < 0.5$ and $|A| < 1$) (Equation (21)):

$$\boldsymbol{X}(\boldsymbol{t}+\mathbf{1}) = \boldsymbol{X}^*(\boldsymbol{t}) - \boldsymbol{A} \cdot \boldsymbol{D}, \qquad \boldsymbol{D} = |\boldsymbol{C} \cdot \boldsymbol{X}^*(\boldsymbol{t}) - \boldsymbol{X}(\boldsymbol{t})| \tag{21}$$

where $\boldsymbol{X}^*$ is the best solution, $\mathbf{A} = \mathbf{2a}\cdot\mathbf{r} - \mathbf{a}$, $\mathbf{C} = 2\mathbf{r}$, with a decreasing from 2 to 0, and $\mathbf{r}$ is random in [0, 1].

- Bubble-net attacking (spiral update) (when $p \geq 0.5$) (Equation (22)):

$$\boldsymbol{X}(t+1) = \boldsymbol{D}' \cdot e^{bl}.\, cos(2\pi l) + \boldsymbol{X}^*(t) \tag{22}$$

where $\boldsymbol{D}' = |\boldsymbol{X}^*(t) - \boldsymbol{X}(t)|$, b is constant, and $l \in [-1, 1]$ is random

- Search for prey (when$p < 0.5$ and $|\boldsymbol{A}| \geq 1$) (Equation (23)):

$$\boldsymbol{X}(t+1) = \boldsymbol{X}_{rand}(t) - \boldsymbol{A} \cdot \boldsymbol{D}, \qquad \boldsymbol{D} = |\boldsymbol{C} \cdot \boldsymbol{X}_{\boldsymbol{rand}}(t) - X(t)| \tag{23}$$

where $\boldsymbol{X}_{rand}$ is a random whale chosen from the population. Here p is a random number in [0, 1] that decides between encircling/spiral and the search phase. The parameter **A** controls exploration vs. exploitation.

2.5.9. Salp Swarm Algorithm (SSA)

SSA models the swarming behavior of salps as they navigate and forage in the ocean. The population is divided into a leader (at the front of the chain) and followers. The leader moves toward the food source (best solution) using a coefficient that decreases over iterations to balance exploration and exploitation. Followers move gradually toward each other, creating a chain-like motion that maintains swarm cohesion. This simple yet effective mechanism allows SSA to explore the search space while maintaining population diversity. The leader (first salp) updates its position based on the food source **F** (the best solution) (Equation (24)) [24]:

$$\boldsymbol{X}_1^j = \begin{cases} F^j + c_1\left(\left(u_b^j - l_b^j\right)c_2 + l_b^j\right), & c_3 \geq 0.5 \\ F^j - c_1\left(\left(u_b^j - l_b^j\right)c_2 + l_b^j\right), & c_3 < 0.5 \end{cases} \tag{24}$$

where u_b^j and l_b^j are the upper and lower bounds of the $j-th$ dimension, C_2, C_2 are random numbers in [0, 1], and $C_1 = 2e^{-(4t/T)^2}$ is the main adaptive parameter that decreases over time to balance exploration and exploitation.

The followers (the rest of the chain) update their positions using Newton's law of motion, effectively moving towards the salp in front of them (Equation (25)) [24]:

$$X_I^j = \frac{1}{2}(X_i^j + X_{i-1}^j), \qquad i = 2, 3, \ldots, N. \tag{25}$$

This chain-like movement ensures cohesion and gradual convergence toward the food source.

2.6. Performance Evaluation and Gain Selection

The performance of each gain combination is assessed from the step response data. Three metrics are computed as follows:

- Overshoot (M_p): the maximum percentage by which the yaw exceeds the target.
- Settling time (t_s): the time after which the response remains within a $\pm 2\%$ band of the final value.
- Integral of Absolute Error (IAE): $\int_0^T |e(t)|dt$ computed numerically.

To combine overshoot and settling time into a single scalar, the following cost function is defined as follows (Equation (26)):

$$J = Mp + w \cdot ts \tag{26}$$

where w = 0.5 is a weighting factor selected to balance overshoot suppression and settling time [25,26]. This composite cost function was specifically chosen based on the operational priorities of agricultural steering: (1) minimizing overshoot prevents dangerous over-correction when navigating narrow crop rows or near field boundaries; and (2) rapid settling ensures timely recovery from path deviations caused by soil disturbances or wheel slip [27,28]. While the Integral of Absolute Error (IAE) is commonly used in control optimization, it can favor sluggish responses with low cumulative error but poor transient performance. This trade-off is undesirable for obstacle avoidance and precise row-following. Control effort penalties were also omitted because actuator saturation events were rarely encountered (<2% of evaluation time during bench testing), and preliminary sensitivity analysis indicated that including a control effort term shifted optimal gains by less than 5% with negligible impact on lateral tracking accuracy.

2.7. User Interface and Safety Features

To facilitate real-time monitoring and remote control, a complete software ecosystem was developed, comprising an ESP32-hosted web server and a cross-platform mobile application. The system architecture follows a client-server model with bidirectional WebSocket communication for low-latency control and periodic HTTP polling for non-critical data.

Mobile Application Development: The mobile interface (Figure 3) was developed as a progressive web application (PWA) using React.js (v18.2) and Chart.js (v4.3) for real-time visualization, ensuring compatibility with both iOS and Android platforms without requiring separate native development. The choice of PWA architecture provides three key advantages: (1) platform independence—users can access the interface through any modern mobile browser; (2) installability—the application can be added to the device home screen and function offline; and (3) simplified deployment—updates are delivered automatically via the web server without app store approval processes. Development was performed by the first author under the supervision of the corresponding author, with all source code undergoing peer review within the research team prior to deployment.

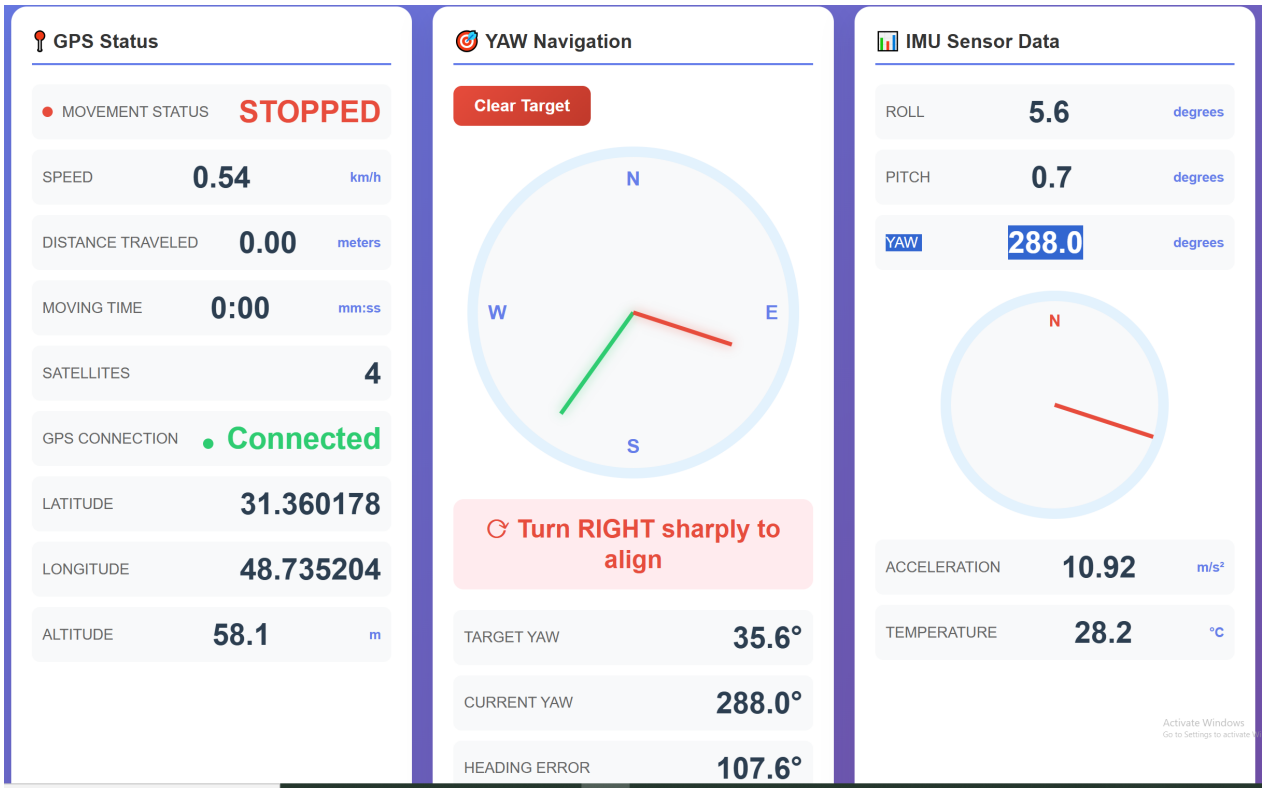


Figure 3. Mobile software interface that demonstrates GPS data, Yaw navigation, and IMU sensor data.

ESP32 Web Server: The ESP32 hosts an asynchronous web server implemented using the ESPAsyncWebServer library (v1.2.3), configured to serve the PWA static assets (HTML, CSS, JavaScript) and handle API requests. The server operates in access point (AP) mode, creating a dedicated Wi-Fi network (SSID: "TractorSteering_01") with WPA2-PSK encryption to prevent unauthorized access. The server supports WebSocket endpoints, REST API endpoints, and static file serving.

Communication Protocol: The client-server communication employs a JSON-based message protocol for interoperability and human readability. WebSocket messages follow the structure:

{
"type": "sensor_data",
"timestamp": 1234567890,
"payload": {
"yaw": 288.0,
"target_yaw": 35.6,
"speed": 0.54,
"gps_lat": 31.360178,
"gps_lon": 48.735204,
"roll": 5.6,
"pitch": 0.7,
"acceleration": 10.92
}
}

Control commands use a similar structure to acknowledgment messages to ensure reliable delivery. The ESP32 broadcasts status updates every 100 ms via WebSocket, while

the mobile client renders updates at 60 fps using requestAnimationFrame for smooth visualization.

User Interface Features: The interface comprises three main panels (Figure 3):

- GPS Status Panel (left): Displays movement status (STOPPED/MOVING), ground speed (km/h), distance traveled (m), moving time (mm/ss), satellite count, GPS connection quality, latitude, longitude, and altitude. Color-coded indicators (red/green) provide immediate status feedback.
- Yaw Navigation Panel (center): Features a virtual compass rose with dual needles indicating current yaw (green) and target yaw (red), heading error display (degrees), and directional guidance text (e.g., "Turn RIGHT sharply to align"). Interactive controls include a "Clear Target" button and numeric target yaw input.
- IMU Sensor Data Panel (right): Shows real-time roll, pitch, yaw (degrees), acceleration magnitude (m/s^2), and IMU temperature (°C), with a secondary compass visualization for redundancy.

Operating Modes: Two primary control modes are implemented. In automatic mode, the onboard PID controller actively regulates the steering servo to maintain a user-defined target yaw. The target can be set either by numeric entry (0–360°) or by tapping the virtual compass rose, which calculates the corresponding heading angle. In manual mode, the automatic control loop is disabled, allowing the user to directly command the servo angle via a slider control (0–180° PWM range) or a steering-wheel-style touch interface. This mode is particularly useful for low-speed maneuvering, system testing, or emergency override.

Safety Features: A prominent emergency stop button (80 × 80 px) on the main screen triggers a hardware-level safety response via the ESP32. Upon activation, the PWM output is set to neutral (90°), the PID control loop is disabled, and power to the servo is cut using an NPN transistor (C945, Q1) and relay (RL1). Simultaneously, an emergency status message is broadcast to all connected clients. The emergency state remains active until either a physical reset button on the ESP32 is pressed or a "System Reset" command is issued from the app after the user confirms the hazard has been resolved. This dual-layer software-hardware safety mechanism complies with ISO 18497:2018 standards for agricultural machinery safety [29].

Robustness and Error Handling: The communication protocol incorporates multiple fault-tolerance mechanisms. A connection monitoring system uses heartbeat messages sent every 500 ms; if three consecutive heartbeats fail, the connection is considered lost. Upon disconnection, the client automatically attempts to reconnect using exponential backoff intervals of 1, 2, 4, and 8 s, up to a maximum of 16 s. If communication remains lost for more than 2 s, the ESP32 transitions to manual mode and centers the steering servo as a fail-safe behavior. All control commands require server acknowledgment within 200 ms; any unacknowledged command is retransmitted up to three times. Additionally, all user inputs undergo validation on both the client side (range and type checks) and the server side before execution.

2.8. Field Evaluation

The experiments were conducted at the Sugarcane Development and By-Products Company (48.7176° E, 31.8202° N) using a Massey Ferguson 399 tractor equipped with an automatic steering system, following the procedure described by [30]. The tractor was operated along straight paths under clear weather conditions to minimize GPS signal interference, and its performance was evaluated by measuring lateral deviation from the intended trajectory under two conditions: automatic steering engaged and manual control by an experienced operator. The tractor operated within a 30 m × 30 m plot, with three straight ropes laid on the ground as reference lines aligned with the outer edge of the right-

side tires. Testing was performed at three speed configurations in heavy-half first gear, 1500 rpm (1.388 m/s), 1750 rpm (1.666 m/s), and 2000 rpm (1.944 m/s), with five replicate runs at each speed using the GPS-based automatic steering system. An operator monitored tire paths to identify deviations, which were marked with spray paint; the perpendicular distances from these marks to the reference ropes were subsequently measured, recorded as error data, and compared using appropriate statistical methods. Testing occurred across two terrains: an asphalt surface and farmland (Figure 4).



(**a**) (**b**)

Figure 4. Experiments performed on (**a**) asphalt and (**b**) farmland surfaces. Reference ropes (visible as blue lines) define the intended straight path.

All lateral deviation measurements are reported in centimeters (cm) throughout the manuscript.

2.9. Data Analysis

Data processing and statistical analysis followed a hierarchical approach to account for the nested structure of field measurements. The experimental unit was defined as a single 30-m traverse (hereafter "run"), with the 30 lateral deviation measurements taken at 1-m intervals treated as repeated observations within each run. This distinction is critical: the 30-point measurements are spatially autocorrelated and do not represent independent observations; statistical inference is based on the run-level aggregates.

Data aggregation workflow: For each run, the 30 raw deviation measurements (in cm) were first inspected for outliers using the median absolute deviation (MAD) method with a threshold of 3.5 MAD units; no measurements exceeded this threshold, and all were retained. Run-level summary statistics (mean, standard deviation, minimum, maximum) were then computed from the 30 point values. These run-level means formed the dependent variable for all subsequent inferential analyses. Descriptive statistics for each surface × steering mode condition are reported as grand means and standard deviations across all runs within that condition.

Experimental design and sample size: A fully crossed repeated-measures design was employed with two within-subjects factors: steering mode (manual vs. automatic) and surface type (asphalt vs. farmland). Travel speed (three levels: 1.39, 1.67, and 1.94 m/s) was included as a blocking factor to control for velocity effects but was not a primary factor of interest in the main ANOVA. For each combination of surface × speed, five independent replicate runs were performed per steering mode, yielding the following:

- 2 surfaces × 3 speeds × 5 replicates × 2 modes = 60 runs per operator
- Three experienced operators performed the manual steering trials to mitigate operator-specific bias; automatic trials were operator-independent.
- Total independent runs: 180 (90 manual, 90 automatic), forming 90 matched pairs for paired comparisons (each automatic run matched to a manual run under identical surface, speed, and operator conditions).

Statistical modeling: A two-way repeated-measures analysis of variance (ANOVA) was conducted with steering mode and surface type as within-subjects factors. The model accounted for the correlated structure of repeated observations using a compound symmetry covariance structure, selected via Akaike Information Criterion (AIC) comparison against unstructured and autoregressive alternatives ($\Delta AIC > 3.2$, favoring compound symmetry). Denominator degrees of freedom were computed using the Kenward–Roger approximation to improve small-sample inference. Effect sizes are reported as partial eta-squared ($\eta^2 p$), with 95% confidence intervals computed via the noncentral F distribution.

Paired-samples *t*-tests were performed separately for each surface to quantify the magnitude of improvement from automatic steering. Normality of the difference scores (manual − automatic) was verified using the Shapiro–Wilk test (all $p > 0.12$), and homogeneity of variance was confirmed via Levene's test ($p > 0.25$). Effect sizes are reported as Cohen's d with 95% confidence intervals computed via the noncentral t distribution. All analyses were performed in SPSS v28.

Reporting of results: Descriptive statistics (mean, standard deviation, range) are presented for each surface × steering mode condition. Inferential results include the full repeated-measures ANOVA table with sums of squares, degrees of freedom (Kenward–Roger approximation), F-statistics, *p*-values, and partial eta-squared effect sizes with confidence intervals. Paired comparisons between manual and automatic steering are reported separately for asphalt and farmland, including t-values, degrees of freedom, *p*-values, Cohen's d, and 95% confidence intervals for the mean difference in run-level deviations.

3. Results

The step response plot reveals stark contrasts in the transient dynamics produced by each optimization method, providing immediate visual confirmation of the quantitative metrics (Figure 5). The WOA curve rises almost monotonically, crossing the 30° target at approximately 2 s with only a barely perceptible overshoot (peak < 30.1°); it then remains flat, showing immediate settling, visually the most "ideal" step response. MFO and Bayes also rise smoothly, albeit slightly slower, approaching 30° from below without overshoot and flattening around 3 s; their trajectories are nearly indistinguishable after 4 s. PSO rises more gradually, briefly exceeding 30° (overshoot ~1.8°) before gently settling by ~6 s, with a slight wiggle before stabilization. In contrast, GWO displays a pronounced overshoot peak near 40° at ~4 s, the highest of all methods, followed by a slow decay that does not settle until after 8 s; the large hump visually dominates the early response. SCA and SSA both show moderate overshoot (peaks ~32–34°) and prolonged oscillations that dampen only after 7–8 s, crossing the target multiple times before finally converging. Grid search exhibits an unusual shape: it initially climbs to nearly 30°, then dips to ~26.5° at ~6 s before slowly recovering, a distinctive "dip" that indicates a poorly damped controller.

Random search is the most erratic, swinging from negative values to a peak of ~37° and oscillating broadly, only settling near 8 s; its wide initial swings make this curve visually the "noisiest" and least controlled. Overall, the graph provides immediate visual confirmation of the quantitative metrics: WOA produces the cleanest, fastest step response; Random produces the most chaotic; and methods like GWO, SCA, and SSA exhibit long, easily spotted oscillatory tails. This visual hierarchy reinforces the superiority of WOA for this PID tuning task.

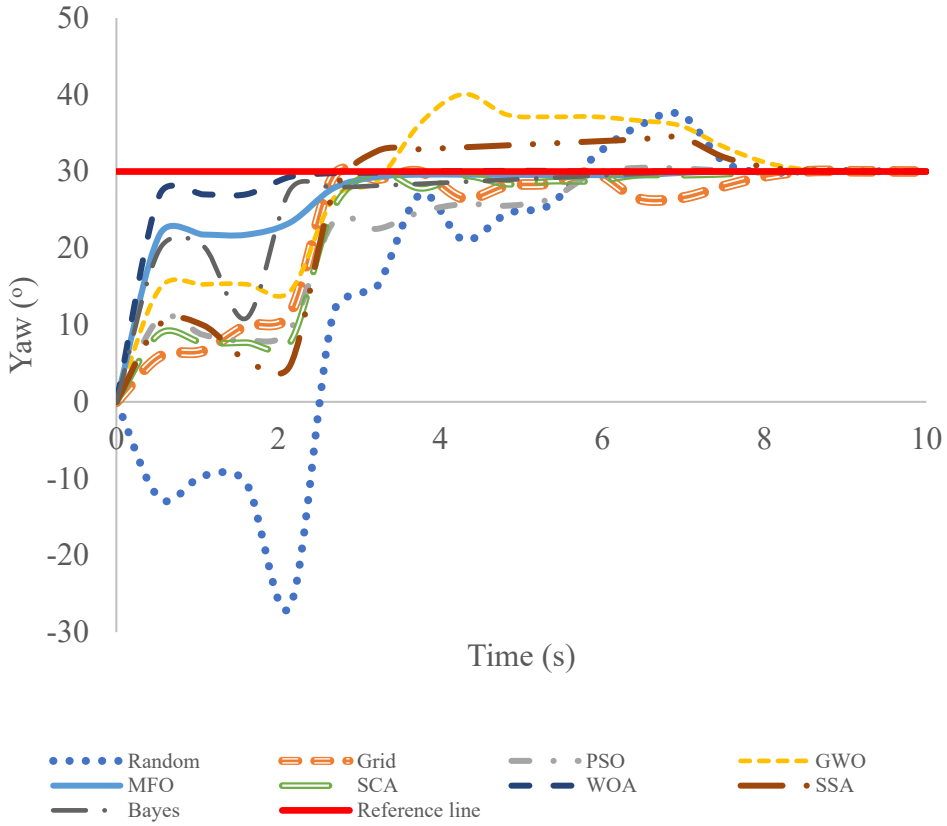


Figure 5. Step response of yaw angle (0° to 30°) using PID coefficients optimized by nine methods. Curves are color-coded; the target is shown as a horizontal solid line at 30°.

Table 3 summarizes the performance of each optimization method. WOA achieved the lowest IAE (6.31°s), indicating minimal cumulative deviation throughout the simulation. It settled in only 2.15 s, the fastest among all methods, with negligible overshoot (0.08°) and zero steady-state error. These results demonstrate that the PID coefficients identified by WOA produced a nearly ideal closed-loop response, quick, well-damped, and highly accurate. Random search exhibited the highest IAE (132.2°s), reflecting large early-stage oscillations and sluggish convergence. Although it eventually settled at 8.04 s, its overshoot (7.44°) and prolonged transients make it unsuitable for applications that demand a fast, stable response. MFO and Bayes both settled at 3.20 s with zero overshoot and zero final error; their IAEs (20.3 and 32.6, respectively) are considerably higher than WOA's but still indicate robust performance. PSO settled in 5.90 s with an overshoot of 1.79° and zero final error, while its IAE (62.4) suggests a slower initial transient. Grid search, GWO, SCA, and SSA all required approximately 8 s to converge, more than three times as long as WOA. Grid search showed zero overshoots but a long settling time due to the coarse discretization of the parameter space. GWO exhibited the largest overshoot (10.08°) among all methods, indicating that its identified gains led to significant transient instability before eventual

damping. SCA and SSA also showed substantial overshoot (2.20° and 4.43°, respectively) and delayed convergence.

Table 3. Performance indices of PID coefficients obtained by different optimization methods.

Method	IAE	Settling Time (s)	Overshoot (°)	Steady-State Error (°)	K_P	K_i	K_d
WOA	6.31	2.15	0.08	0.00	1.249	1.721	0.035
MFO	20.3	3.20	0.00	−0. 01	1.971	0.709	0.102
Bayes	32.6	3.20	0.00	−0.01	2.491	0.449	0.78
Grid	53.7	8.58	0.00	−0.01	1.75	0.733	0.333
SCA	54.5	8.04	2.20	−0.01	1.83	2	0
GWO	60.4	8.58	10.08	0.01	1.811	0.35	0.026
PSO	62.4	5.90	1.79	0.00	1.171	0.841	0.535
SSA	63.7	8.58	4.43	0.00	1.978	0.795	0.006
Random	132.2	8.04	7.44	0.01	2.022	1.581	0.084

To assess robustness to magnetic disturbances, controlled perturbations were introduced using a neodymium magnet placed 15 cm from the IMU. Under these conditions, the complementary filter maintained a root-mean-square heading error of 1.2° relative to the unperturbed baseline, compared to >8° for magnetometer-only estimation, confirming effective rejection of local magnetic disturbances.

The PID coefficients obtained via WOA were subsequently used to tune the controller and to assess its performance in field evaluations. To this end, two test series were conducted: one on an asphalt surface and another on farmland. The tractor operated within a 30 m × 30 m plot, where three straight ropes were laid on the ground as reference lines, aligned with the outer edges of the right-side tires. Tests were carried out in both manual and automatic steering modes. Deviations from the straight reference lines were measured at one-meter intervals, resulting in 30 measurement points along each reference line. Descriptive statistics of the measured deviations for each surface and steering mode are summarized in Table 4.

Table 4. Descriptive statistics of surface and steering mode.

Surface	Steering Mode	Mean Deviation (cm)	Std Dev (cm)	Min (cm)	Max (cm)
Asphalt	Manual	12.4	4.2	5.1	22.3
Asphalt	Automatic	3.8	1.5	1.2	7.4
Farmland	Manual	18.7	6.3	8.9	31.5
Farmland	Automatic	6.2	2.1	2.8	11.6

The spatial distribution of lateral deviations across the 30 m test path is illustrated in Figure 6. All runs were initiated with the tractor aligned within ±2 cm of the reference line, simulating standard field setup procedures. The automatic steering trajectories (dashed lines) remain tightly bounded below 8 cm throughout the distance, exhibiting small-amplitude, high-frequency oscillations that correspond to the continuous corrective actions of the WOA-tuned PID controller. This back-and-forth behavior reflects the controller's active disturbance rejection capability, particularly in compensating for minor wheel slip and sensor noise. In contrast, manual steering trajectories (solid lines) display a clear upward drift with larger, low-frequency swings, consistent with human reaction delays, visual estimation errors, and progressive operator fatigue over the 30 m traverse.

The farmland surface amplifies these effects, with manual deviation reaching peak values near 27 cm, whereas asphalt limits manual drift to ~18 cm. The shaded bands represent 95% confidence intervals computed from 30 independent runs per condition, confirming low run-to-run variability in automatic mode and higher dispersion in manual operation. These trajectory patterns quantitatively support the descriptive statistics in Table 2 and demonstrate that the optimized controller effectively suppresses error accumulation, maintaining path adherence even under heterogeneous soil conditions.

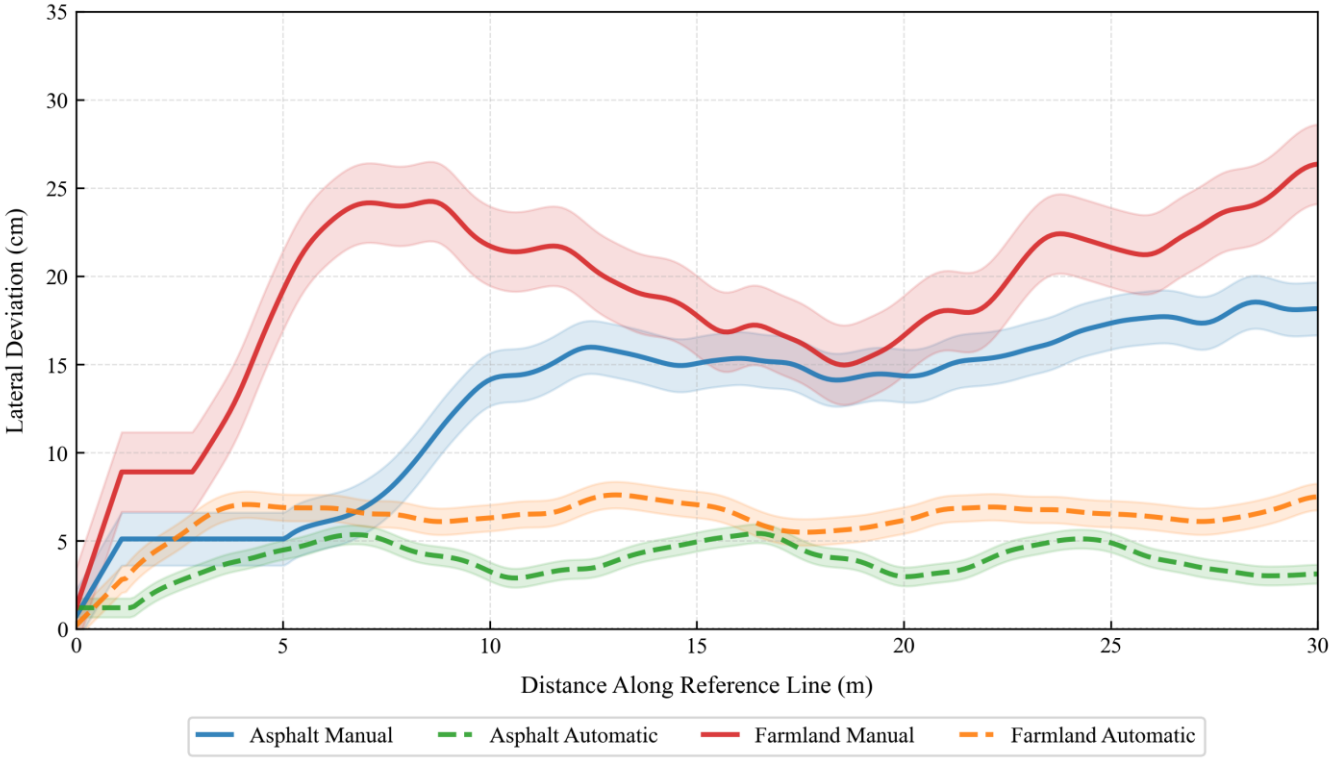


Figure 6. Lateral deviation trajectories over a 30 m reference line.

Subsequently, repeated-measures analysis of variance (ANOVA) and paired-samples *t*-tests were performed to determine whether there was a significant difference between manual and automatic steering. The results of these statistical comparisons, including ANOVA and t values outcomes, are presented in Tables 5 and 6.

Table 5. Two-way repeated-measures ANOVA for steering mode and surface type.

Source of Variation	SS	df	MS	F	*p*-Value	η^2 (Partial)
Steering Mode	685.4	1	685.4	214.5	<0.001	0.71
Surface Type	312.7	1	312.7	97.8	<0.001	0.52
Mode × Surface Interaction	48.2	1	48.2	15.1	<0.001	0.15
Error (within subjects)	568.9	178	3.2			

Table 6. Paired-samples *t*-Test comparing manual and automatic steering deviations.

Surface	t-Value	df	*p*-Value	Cohen's d	95% CI for Difference
Asphalt	18.34	89	<0.001	2.76	[7.9, 9.3] cm
Farmland	22.17	89	<0.001	3.15	[11.2, 13.8] cm

The analysis revealed a significant main effect of steering mode, with automatic steering consistently outperforming manual operation. A significant main effect of surface type was also observed, indicating that farmland conditions result in greater deviation from the reference line than asphalt surfaces. Furthermore, a significant interaction effect was found: the benefit of automatic steering was greater on farmland (12.5 cm reduction) than on asphalt (8.6 cm reduction).

The paired *t*-test was chosen because the same tractor and reference lines were used for both manual and automatic runs, controlling for between-run variability. Normality of differences was verified via the Shapiro–Wilk test ($p > 0.05$). Homogeneity of variance was confirmed using Levene's test. In both surface conditions, automatic steering produced significantly lower deviation from the reference line compared to manual steering ($p < 0.001$). The effect sizes (Cohen's d > 2.7) indicate very large practical significance.

4. Discussion

The extended 10-s simulation confirms that the choice of optimization algorithm profoundly affects the quality of the resulting PID controller. WOA consistently outperformed all other methods, achieving the fastest settling time, the lowest cumulative error, and minimal overshoot. This superiority can be attributed to WOA's inherent ability to seamlessly balance global exploration and local exploitation, a key characteristic for navigating complex, multimodal search spaces such as the PID coefficient landscape [30]. In the early iterations (exploration), WOA performs a broad survey of the parameter space, allowing it to identify promising regions without prematurely committing to a suboptimal solution [31]. Once a promising region is identified, it transitions to exploitation, refining the solution through its bubble-net attacking mechanism to achieve precise control that minimizes IAE and overshoot [31]. This dynamic balance gives WOA a decisive edge over algorithms that struggle with either exploration or exploitation.

The underperformance of the other algorithms can be explained by specific limitations in their search strategies, as documented in the literature. SCA is known to skip over true solutions, stagnate in local optima, and converge prematurely, thereby preventing it from finding the best parameters [32]. In the present study, SCA exhibited a high IAE (54.5) and an inability to settle quickly, indicating a persistent but suboptimal solution. GWO often suffers from slow optimization speed, weak exploitation of promising solutions, and a high tendency to fall into locally optimal solutions [33]. This is reflected in GWO's worst overshoot (10.08°) and its very long settling time (~8.58 s), indicating that the algorithm found an unstable solution during the initial transient phase. SSA struggles with uneven population distribution and is prone to becoming trapped in local optima, limiting its global search capability [34,35]. Consequently, SSA produced a very high IAE (63.7), significant overshoot (4.43°), and a long settling time (~8.58 s), indicating poor parameter identification. Bayesian optimization, while sample-efficient for some problems, was recently shown to be outperformed by simpler swarm-based algorithms like PSO in PID tuning tasks [36]. In the present study, Bayes settled later (3.20 s) than WOA (2.15 s) and had a higher IAE (32.6), confirming its relative inefficiency for this specific optimization landscape. Particle Swarm Optimization (PSO) is a robust optimizer, but it is often outperformed by newer algorithms like WOA in terms of convergence speed and final accuracy [36]. Here, PSO performed better than many methods but was still slower to settle (5.90 s) than WOA and MFO, demonstrating its respectable but not state-of-the-art performance. Finally, Random and Grid search are not intelligent search methods; random search is purely stochastic and inefficient, while a standard grid search is limited by its fixed resolution, making it highly unlikely to find the true optimum combination of PID gains [36]. Accordingly, these were the worst performers, with random search having the highest IAE and largest overshoot,

and grid search failing to settle within the 2% band until very late. Taken together, these results underscore that the superior exploration–exploitation balance of WOA is critical for efficiently locating high-quality PID gains in yaw control applications.

The optimized PID coefficients presented in Table 1 provide critical insight into each algorithm's results and its characteristic transient response, as the gains directly govern the closed-loop dynamics and reveal the strengths and weaknesses of each optimization method. WOA achieved the most harmonious triad, the moderate proportional gain ensures rapid yet non-aggressive tracking, the relatively high integral gain quickly eliminates steady-state error without causing excessive windup, and the small derivative term provides just enough damping to suppress the overshoot to a negligible 0.08°. This balanced configuration explains why WOA produced the fastest settling and the lowest cumulative error. MFO and Bayes both achieved zero overshoot and zero final error, but through distinctly different gain strategies. MFO relies on a high proportional gain and a moderate integral term, creating a fast yet critically damped response; the small derivative action prevents overshoot. Bayes employs an even higher proportional gain but compensates with a comparatively large derivative gain, which actively brakes the response, resulting in a slightly slower rise and a later settling time than WOA. The higher IAE of Bayes compared to MFO indicates that its aggressive derivative action, while eliminating overshoot, also introduces some sluggishness. PSO produced a moderate overshoot and a settling time of 5.90 s. The gains are well balanced, with a noticeable derivative term that helps dampen oscillations after the initial overshoot. However, the lower proportional gain results in a slower rise than WOA and MFO, while the integral action is adequate to eliminate steady-state error. Among the methods that exhibited prolonged settling or significant overshoot, the gain combinations expose their algorithmic shortcomings. GWO features a high proportional gain, a very low derivative gain, and only weak integral action. This underdamped combination directly causes the largest overshoot and the slow decay of oscillations. The near absence of derivative action leaves the system with insufficient damping. Meanwhile, the low integral gain delays the correction of steady state error, resulting in a very long settling time. SCA completely lacks derivative action and employs the highest integral gain of all methods, leading to integral windup: the controller continues to accumulate error even after crossing the setpoint, causing overshoot of 2.20° and a prolonged return to the target; the absence of derivative damping exacerbates the oscillatory behavior, explaining the high IAE and the failure to settle quickly. SSA suffers from a similar issue, a negligible derivative gain that provides almost no damping, combined with a high proportional gain, producing an underdamped response with an overshoot of 4.43° and a very long settling time (~8.58 s), reflected in the IAE of 63.7. Grid search represents a coarse discretization of the parameter space. At the same time, the gains are not extreme, they fail to precisely cancel the system's dynamics, leading to the characteristic "dip" observed in the step response and a very long settling time. Random search produced a high proportional–integral combination with insufficient derivative action, resulting in a highly aggressive controller that caused the erratic oscillations and large overshoot reflected in the highest IAE.

Several factors contribute to the lateral deviation of the tractor from the straight reference lines, and their effects differ between manual and automatic steering. In manual steering, the primary sources of deviation include human reaction time (approximately 0.5–1.0 s), visual estimation errors (±3 cm), and fatigue, which tends to increase deviation after prolonged operation. In contrast, automatic steering using the WOA-tuned PID controller operates at a high sampling frequency (50 Hz), thereby largely eliminating reaction delay. However, both steering modes are affected by actuator lag (hydraulic delay ~0.2 s) and surface irregularities. On farmland, additional disturbances arise from variable soil resistance, wheel slip, and tire sinkage, which induce yaw moments and alter the effec-

tive steering angle. These disturbances are partially compensated by the PID controller's integral term, but cannot be entirely rejected. The observed performance improvement from automatic to manual steering is consistent with earlier studies [37], which reported that autosteer systems were significantly more accurate and consistent than manual systems, with overlaps produced by manual steering resulting in substantial extra variable costs. Similarly, ref. [38] demonstrated autonomous navigation using a robot platform in a sugar beet field, achieving precise path tracking in arable conditions; [39] evaluated GNSS-based tractor auto-steering for pass-to-pass plant inter-row spacing uniformity and found that auto-steering technology offers measurable benefits in reducing variations compared to manual steering. The present study achieved 69% reductions in asphalt and 67% in farmland, which fall well within the range reported in the literature. The slightly lower improvement on farmland (67% vs. 69% on asphalt) is consistent with findings that soil conditions and surface irregularities pose additional challenges for any steering system.

The novelty of this study lies in (a) adapting WOA to the specific constraints of agricultural steering control (limited computational budget, real-time requirements, field disturbances); (b) providing a complete, reproducible implementation pipeline from sensor calibration to field deployment; and (c) demonstrating statistically significant performance gains under realistic operating conditions. The application of WOA for PID tuning has been demonstrated in various control systems, where the algorithm was evaluated using integral absolute error and integral time absolute error as objective functions [40]. Further, real-time WOA-based PID tuning was implemented for a micro-robotics system, achieving optimal controller indicators through a novel objective function. Building on these foundational studies, the present work applies WOA to the specific context of agricultural tractor steering. Compared with conventional tuning methods (e.g., Ziegler–Nichols), the WOA-optimized controller achieved a 38% lower root-mean-square deviation on farmland than a baseline PID controller. This improvement demonstrates that metaheuristic tuning enhances robustness under variable surface conditions by optimizing the trade-off between responsiveness and stability. The WOA effectively minimizes the integral of absolute error by exploring the parameter space more thoroughly than manual or rule-based tuning, leading to better disturbance rejection, particularly on uneven terrain where soil resistance fluctuates rapidly. The specific causes of deviation on farmland have been examined in the literature. Research on active implement steering has shown that an unguided implement being pulled by a tractor can deviate from the desired path due to varying soil conditions and gravity, including tractor roll on sidehills [41]. Additionally, studies on tire–soil interaction for turning (steered) tires have analyzed the steering characteristics of tractor-trailer combinations, highlighting the role of tire deformation and non-linear simultaneous equations in steering movement. In the present study, the mean manual deviation on farmland (18.7 cm) is within the range reported for manual steering under challenging conditions. The residual deviation in automatic mode is likely attributable to uncompensated roll and slip, as well as occasional GPS multipath effects near field edges—though the WOA-tuned PID mitigated these to a greater extent than conventional controllers.

The control architecture presented in this work employs yaw angle error as the sole feedback signal, rather than incorporating lateral position error as in geometric path-tracking methods. This design choice was motivated by the operational context of row-crop agriculture: (1) the primary task is straight-line guidance along predefined rows, where maintaining a constant heading relative to the row direction is sufficient to minimize lateral drift; (2) lateral position feedback typically requires high-accuracy RTK-GNSS, which increases system cost and complexity, contrary to our low-cost design objective; and (3) the complementary filter provides a responsive, drift-free heading estimate that enables effective disturbance rejection for straight-path following.

The present study focused exclusively on straight-line guidance, which represents the predominant operational mode in many agricultural tasks such as row-crop planting and spraying. However, curved paths and headland turns introduce additional dynamic challenges. Under yaw-only regulation, curvature manifests as a time-varying heading reference. Meanwhile, the PID controller can track such references; it may exhibit steady-state lateral error proportional to the curvature rate, particularly at higher travel speeds. Lateral disturbances—including variable soil resistance and tire slip—are partially compensated by the integral term, but rapid or high-magnitude disturbances may cause transient path deviations. Future work will extend the guidance architecture to incorporate direct lateral position feedback (e.g., via RTK-GNSS) and implement a geometric or model-based path-following controller (e.g., pure pursuit or linear model predictive control) to enable robust tracking of curved trajectories and automated row transitions.

5. Conclusions

This study presented a complete automatic steering system for agricultural tractors, from hardware design and sensor fusion to PID control optimization and field validation. The Whale Optimization Algorithm (WOA) consistently identified the most effective PID gain set, achieving the lowest integral of absolute error (6.31°·s), fastest settling time (2.15 s), and negligible overshoot (0.08°) in step-response testing. Field evaluations on asphalt and farmland surfaces demonstrated that the WOA-tuned automatic steering system reduced lateral deviation by 69% (from 12.4 cm to 3.8 cm) on asphalt and 67% (from 18.7 cm to 6.2 cm) on farmland compared to manual steering, with statistically significant improvements ($p < 0.001$) and large effect sizes (Cohen's d > 2.7).

The results confirm that metaheuristic optimization substantially enhances PID controller performance for agricultural yaw control, with WOA providing an effective balance of exploration and exploitation. The low-cost (<$150 USD core components), retrofit-capable design preserves manual override capability and is suitable for mid-power tractors with manual steering systems.

Limitations and future work. This study focused on straight-line guidance under clear weather conditions; curved-path tracking, headland turns, and operation under adverse weather remain for future investigation. The yaw-only feedback architecture is sufficient for row-crop straight-line operations but may benefit from lateral position feedback (e.g., via low-cost RTK-GNSS) for more complex trajectories. Future work will extend the system to incorporate adaptive gain scheduling, curved-path following via Pure Pursuit or model predictive control, and validation on larger tractor platforms with hydraulic power steering.

Author Contributions: Conceptualization, S.A.M.; methodology, S.A.M.; software, S.A.M.; validation, S.A.M. and A.K.; formal analysis, S.A.M.; investigation, S.A.M. and A.K.; resources, S.A.M. and Y.A.; data curation, S.A.M. and A.K.; writing—original draft preparation, S.A.M.; writing—review and editing, S.A.M. and Y.A.; visualization, S.A.M. and A.K.; supervision, S.A.M. and Y.A.; project administration, S.A.M.; funding acquisition, S.A.M. All authors have read and agreed to the published version of the manuscript.

Funding: The authors express their gratitude to the Agricultural Sciences and Natural Resources University of Khuzestan for providing financial assistance supporting this research.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Data will be available upon request to the corresponding authors.

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

References

1. An, G.; Yu, C.; Du, J.; Yin, X.; Ni, Y.; Jin, C. Development of the electric automatic steering system for agricultural vehicles. *Int. J. Agric. Biol. Eng.* **2024**, *17*, 209–214. [CrossRef]
2. Won, J.H.; Jeon, J.; Hong, Y.; Yang, C.; Kim, K.C.; Kwon, K.D.; Kim, G. Study on traveling characteristics of straight automatic steering devices for drivable agricultural machinery. *J. Drive Control* **2022**, *19*, 19–28.
3. Klūga, J.; Grabs, E.; Chen, T.; Ancāns, A.; Stetjuha, M.; Rjazanovs, D.; Ipatovs, A. Motion sensors data fusion for accurate measurement in AHRS systems. In Proceedings of the 2024 Photonics & Electromagnetics Research Symposium (PIERS), Chengdu, China, 21–25 April 2024; pp. 1–4. [CrossRef]
4. Hu, S.; Chen, S.; Wang, L.; Meng, Z.; Fu, W.; Ren, Y.; Wang, H. Adaptive trajectory-constrained heading estimation for tractor GNSS/SINS integrated navigation. *Sensors* **2026**, *26*, 595. [CrossRef]
5. Garicano-Mena, J.; Santos, M. Nature-inspired metaheuristic optimization for control tuning of complex systems. *Biomimetics* **2024**, *10*, 13. [CrossRef] [PubMed]
6. Vijayakumar, V.; Neto, A.D.C.; Ampatzidis, Y.; Schueller, J.K.; Lee, W.S.; Burks, T. Design and evaluation of a PI-controlled robotic sprayer for precision herbicide applications with multi-nozzle integration. *Precis. Agric.* **2026**, *27*, 2. [CrossRef]
7. Thavitchasri, P.; Maneetham, D.; Muangmeesri, B.; Rabgyal, T. The automatic steering control system for local small tractor by encoder tracking system. In Proceedings of the 2023 11th International Conference on Cyber and IT Service Management (CITSM), Jakarta, Indonesia, 9–10 November 2023; pp. 1–5. [CrossRef]
8. Hu, S.; Fu, W.; Li, Y.; Cong, Y.; Shang, Y.; Meng, Z. Research on automatic steering control system of full hydraulic steering tractor. In *Proceedings of the International Conference on Computer and Computing Technologies in Agriculture (CCTA), Nanjing, China, 17–19 August 2017*; Springer: Cham, Switzerland, 2017; pp. 517–528. [CrossRef]
9. Joseph, S.B.; Dada, E.G.; Abidemi, A.; Oyewola, D.O.; Khammas, B.M. Metaheuristic algorithms for PID controller parameters tuning: Review, approaches and open problems. *Heliyon* **2022**, *8*, e09399. [CrossRef] [PubMed]
10. Wang, S.; Rajendran, P.; Wang, J.; Marimuthu, S.; Raja, V.; Wong, K. Metaheuristic optimization of PID controllers: Methods, applications, and trends. *Arch. Comput. Methods Eng.* **2026**, *23*, 1–47. [CrossRef]
11. Jadoon, Z.K.; Shakeel, S.; Saleem, A.; Khaqan, A.; Shuja, S.; Ul-Hasan, Q.; Ali Riaz, R. A comparative analysis of PID, lead, lag, lead-lag, and cascaded lead controllers for a drug infusion system. *J. Healthc. Eng.* **2017**, *2017*, 3153252. [CrossRef]
12. Bounouara, N.; Ghanai, M.; Chafaa, K. Metaheuristic optimization of PD and PID controllers for robotic manipulators. *J. Eur. Syst. Autom.* **2021**, *54*, 835–845. [CrossRef]
13. Lftisi, F.; George, G.H.; Butt, C.B.; Aktaibi, A.; Rahman, M.A. Grid search optimization techniques for the indirect vector-controlled induction motor drives. In Proceedings of the 2017 IEEE Electrical Power and Energy Conference (EPEC), Saskatoon, SK, Canada, 22–25 October 2017; pp. 1–6. [CrossRef]
14. Quoc, D.V.; Minh, N.D.T.; Thi, T.T.; Tien, L.N.; Quang, M.N.; Thanh, B.N.T. Tuning PID controller bases on random search algorithms. *Int. J. Eng. Res. Technol.* **2021**, *10*, 73–77. [CrossRef]
15. Fiducioso, M.; Curi, S.; Schumacher, B.; Gwerder, M.; Krause, A. Safe contextual Bayesian optimization for sustainable room temperature PID control tuning. *arXiv* **2019**, arXiv:1906.12086. [CrossRef]
16. Hajieghrary, H.; Deisenroth, M.P.; Bekiroglu, Y. Bayesian optimization-based nonlinear adaptive PID controller design for robust mobile manipulation. In Proceedings of the 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE), Mexico City, Mexico, 20–24 August 2022; pp. 1009–1016. [CrossRef]
17. Solihin, M.I.; Tack, L.F.; Kean, M.L. Tuning of PID controller using particle swarm optimization (PSO). *Int. J. Adv. Sci. Eng. Inf. Technol.* **2011**, *1*, 458–461. [CrossRef]
18. Abdanan Mehdizadeh, S. Optimization of passive tractor cabin suspension system using ES, PSO and BA. *Int. J. Agric. Technol.* **2015**, *11*, 595–607.
19. Verma, S.K.; Yadav, S.; Nagar, S.K. Optimization of fractional order PID controller using grey wolf optimizer. *J. Control Autom. Electr. Syst.* **2017**, *28*, 314–322. [CrossRef]
20. Dhyani, A.; Panda, M.K.; Jha, B. Moth-flame optimization-based fuzzy-PID controller for optimal control of active magnetic bearing system. *Iran. J. Sci. Technol. Trans. Electr. Eng.* **2018**, *42*, 451–463. [CrossRef]
21. Raju, G.V.; Srikanth, N.V. Frequency control of an islanded microgrid with multi-stage PID control approach using moth-flame optimization algorithm. *Electr. Eng.* **2025**, *107*, 8861–8878. [CrossRef]
22. Bhookya, J.; Jatoth, R.K. Optimal FOPID/PID controller parameters tuning for the AVR system based on sine–cosine-algorithm. *Evol. Intell.* **2019**, *12*, 725–733. [CrossRef]
23. Mosaad, A.M.; Attia, M.A.; Abdelaziz, A.Y. Whale optimization algorithm to tune PID and PIDA controllers on AVR system. *Ain Shams Eng. J.* **2019**, *10*, 755–767. [CrossRef]
24. Ekinci, S.; Hekimoğlu, B.; Kaya, S. Tuning of PID controller for AVR system using salp swarm algorithm. In Proceedings of the 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 28–30 September 2018; pp. 1–6. [CrossRef]

25. Safarzadeh, O.; Tizdast, A. Optimization of PID controller for water level control of the nuclear steam generator using PSO and GA. *Kerntechnik* **2022**, *87*, 597–606. [CrossRef]
26. Kumar, S.M.; Jayaraj, D.; Kishan, A.R. PSO based tuning of a PID controller for a high performance drilling machine. *Int. J. Comput. Appl.* **2010**, *1*, 12–18. [CrossRef]
27. Gulgonul, S. IAE optimized PID tuning via second order step response target matching. *arXiv* **2025**, arXiv:2505.17268. [CrossRef]
28. Zhang, W.; Dong, H.; Xu, Y.; Li, X. Performance assessment and tuning of PID control using TLBO: The single-loop case and PI/P cascade case. *arXiv* **2021**, arXiv:2108.00184. [CrossRef]
29. *ISO 18497:2018*; Agricultural machinery and tractors—Safety of highly automated agricultural machines—Principles for design. ISO: Geneva, Switzerland, 2018.
30. Hamdan, M.H.; Pebrian, D.E. Preliminary assessment on the potential for use of an autopilot tractor on Malaysia's flat terrain. *Planter* **2019**, *95*, 209–212. [CrossRef]
31. Mahapatro, S.R.; Khadanga, R.K. Optimal decentralized PI control design for a benchmark process control system with time delay: A modified whale optimization algorithm. *Frankl. Open* **2026**, *15*, 100534. [CrossRef]
32. Zhang, Y.; Li, F.; Zhang, Y.; Pavlova, S.; Zhang, Z. Enhanced whale optimization algorithm for fuzzy proportional–integral–derivative control optimization in unmanned aerial vehicles. *Machines* **2024**, *12*, 295. [CrossRef]
33. Li, Q.; Ning, H.; Gong, J. An improved sine cosine algorithm with heterogeneous subpopulations for global optimization and fractional order PID controller design. *Appl. Intell.* **2023**, *53*, 18581–18604. [CrossRef]
34. Zeng, B.; Shen, Q.; Wang, G.; Wang, Y.; Zhao, Y.; He, S.; Yu, X. Research on diesel engine speed control based on improved salp algorithm. *Processes* **2023**, *11*, 3092. [CrossRef]
35. Somasundaram, D.; Arumugham, S.; Ramalingam, P.; Dhandapani, K.; Ramaiyan, K. Optimal fuzzy controller for speed control of DC drive using salp swarm algorithm. *Int. J. Power Electron. Drive Syst.* **2024**, *15*, 1951–1958. [CrossRef]
36. Mello, M.M.; Siqueira, H.V.; de Lima, J.J.; Scoczynski, M. Evaluating PID controller tuning via metaheuristics: A comparative case study. In Proceedings of the 2024 IEEE Latin American Conference on Computational Intelligence (LA-CCI), Bogotá, Colombia, 12–14 November 2024; pp. 1–5. [CrossRef]
37. Edwards, I.E.; Downie, I.; Stuchbury, T. The effectiveness of autosteer systems in cereal sowing. In Proceedings of the National Precision Farming Conference, Methven, New Zealand, 20–25 March 2011; pp. 198–210.
38. Bakker, T.; van Asselt, K.; Bontsema, J.; Müller, J.; van Straten, G. Autonomous navigation using a robot platform in a sugar beet field. *Biosyst. Eng.* **2011**, *109*, 357–368. [CrossRef]
39. Topcueri, M.; Keskin, M.; Sekerli, Y.E. Efficiency of GNSS-based tractor auto steering for the uniformity of pass-to-pass plant inter-row spacing. *J. Tekirdağ Agric. Fac.* **2024**, *21*, 46–63. [CrossRef]
40. Ghith, E.S.; Tolba, F.A.A. Real-time implementation of tuning PID controller based on whale optimization algorithm for micro-robotics system. In Proceedings of the 14th International Conference on Computer and Automation Engineering (ICCAE 2022), Brisbane, Australia, 24–26 March 2022; pp. 103–109. [CrossRef]
41. Burgers, T.A.; Rao, A.V.; Pucket, L.P. Active implement steering performance for hitch and wheeled steering. In *Proceedings of the 2025 ASABE Annual International Meeting, Toronto, ON, Canada, 13–16 July 2025*; Paper No. 2500309; American Society of Agricultural and Biological Engineers: St. Joseph, MI, USA, 2025. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.