Plant-Wise Soil Nutrient Monitoring in Greenhouses Using a PID-Controlled Mobile Robot with Interpolated Robotic Arm

[1st] Karunarathna U.G.O.S
Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
it21108372@my.sliit.lk

[2nd] Yatawarage U.S
Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
it210835322@my.sliit.lk

[3rd] Basnayake B.M.I.R.B
Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
it21306372@my.sliit.lk

[4th] Kavindya J.A.G
Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
it21928192@my.sliit.lk

[5th] Dr. Kapila Dissanayaka
Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
Kapila.d@sliit.lk

[6th] Suranjini Silva
Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
Suranjini.s@sliit.lk

***Abstract*—Conventional greenhouse automation systems typically manage plants as a collective, applying uniform conditions for watering, nutrient delivery, and environmental control. This generalized approach fails to account for individual plant variability, leading to inefficiencies in resource usage and suboptimal plant health. In this work, we present an autonomous greenhouse management system that enables per-plant monitoring and control using a mobile robotic platform integrated with a multi-joint robotic arm and soil nutrient sensors. The system autonomously navigates through a structured greenhouse layout via PID-controlled line following, identifying individual plant stations through IR-based junction detection. At each plant, the robot halts and activates a calibrated robotic arm that collects soil data—including Nitrogen, Phosphorus, and Potassium (NPK) levels—via an industrial-grade sensor using the Modbus RTU protocol. These readings are wirelessly transmitted to a cloud platform for real-time monitoring and decision-making. The robot's movement and arm gestures are synchronized using interpolated servo control, ensuring smooth, precise operations. Experimental results demonstrate the system's reliability in station detection, nutrient sensing accuracy, and its potential to enable fine-grained plant care [1], which is critical for maximizing agricultural productivity and sustainability in modern greenhouse environments.**

Greenhouse automation, Robotic arm, IoT in agriculture, PID control, Individualized plant management

I. Introduction

Traditional greenhouse systems treat all plants uniformly, applying the same water, nutrients, and care regardless of individual needs. This one-size-fits-all approach leads to resource inefficiency and inconsistent plant health [2]. Our project addresses this limitation by introducing a system that enables individualized care of plants. By monitoring and responding to each plant separately, the system ensures optimized growth conditions, offering a more sustainable and precise solution for modern agricultural and greenhouse environments.

II. Literature Review

Automated systems are increasingly important in modern agriculture, particularly in controlled environments such as greenhouses, to improve efficiency and reduce labor. This involves the use of mobile robots and other automated systems. This review focuses on two key aspects: line-following mobile robots and robotic arm systems, and their application in automating greenhouse tasks.

Line-following robots are designed to navigate autonomously following a defined path. These robots are used in various automation and transportation applications. Various sensors can be used for this purpose, with infrared (IR) sensors being a common choice to detect the line and guide the robot's movement [3]. These robots offer a cost-effective solution for navigation in structured environments. Effective line following requires robust control strategies. PID control is a widely used technique to achieve accurate and stable line following, ensuring that the robot stays on the intended path. Fuzzy logic control has also been explored to handle uncertainties and variations in the line path or environmental conditions. Almasri et al. (2016) [4] developed collision avoidance and line-following techniques for mobile robot navigation, integrating fuzzy logic for enhanced performance. In practical scenarios, line-following robots may encounter obstacles, and research has addressed this by integrating obstacle avoidance capabilities, allowing the robot to navigate around obstructions and return to its original path. Valsalan (2019) [5] presented an autonomous emergency indicating line follower and obstacle avoidance robot.

Robotic arm systems offer precise manipulation capabilities that are valuable in various agricultural tasks. These systems can be used in greenhouses for specific operations. Yang et al. (2024) [6] propose using a mobile robot to automate the transport of plants within a greenhouse. The functionality of a robotic arm is determined by its end effector, which can be designed to hold various tools or sensors. For example, in the context of precision agriculture, an end effector can be designed to hold an NPK sensor for soil analysis. Musa et al. (2024) reviewed Wireless Sensor Networks (WSNs) for precision agriculture, focusing on NPK sensor implementations [7]. The use of IoT is also crucial for the development of smart greenhouse systems and data collection from the sensors. Farooq et al. (2022) discussed an IoT-based smart greenhouse framework and control strategies for sustainable agriculture [2]. Yeasdani et al. (2023) introduced a semi-autonomous IoT robot to modernize traditional farming practices and enhance agricultural productivity in developing countries [8]. Vision-based systems, such as those using AprilTags, can also be used for autonomous navigation and docking of robots, which is relevant to the broader context of mobile robot applications. Guangrui (2017) proposed a vision-based autonomous docking and recharging system for a mobile robot in a warehouse environment [9].

The integration of robotics and automation in agriculture aims to address the increasing global food demand and the need for sustainable farming practices. By automating tasks such as plant monitoring and care, these technologies can improve efficiency, reduce labor costs, and optimize resource utilization. The development of autonomous mobile robots, including line-following robots, is crucial for navigating within agricultural environments like greenhouses. Similarly, robotic arm systems provide the necessary precision and manipulation capabilities for tasks such as soil analysis, crop harvesting, and other plant-specific interventions. The use of IoT enables the collection and analysis of data from various sensors, facilitating informed decision-making and optimized control strategies in smart farming systems.

Research Gap

While line-following robots and robotic arms have been extensively studied and implemented in various domains, their integration for precision agriculture—particularly in the context of individualized plant monitoring and care within greenhouses—remains underexplored. Most existing automated agricultural systems are designed to treat the entire cultivation area homogeneously, lacking the capability to perform localized interventions on a per-plant basis. This generalized approach can lead to inefficiencies in resource usage and suboptimal plant health management. Although robotic arms have shown promise in tasks like harvesting or spraying in agriculture, the coordinated deployment of these systems to perform high-precision, plant-specific tasks such as soil nutrient sensing has not been comprehensively addressed. This project fills that gap by combining a PID-controlled line-following robot with an interpolated robotic arm capable of executing precise sensor deployment at each plant location. The system's ability to navigate autonomously and interact with each plant individually represents a significant advancement toward personalized, data-driven plant care in greenhouse environments.

III. Methodology

This system is designed to enable individualized monitoring and care of plants within a greenhouse environment. Unlike conventional setups that manage crops as a collective unit, this solution integrates a mobile robot capable of navigating to each plant location and performing targeted measurements. The system is structured around a predefined path, where the robot identifies plant stations using junctions and halts for data collection. At each station, a robotic arm performs soil analysis, collecting nutrient data specific to that plant. This modular and autonomous approach enables scalable, precise agricultural management, reducing waste and improving crop health outcomes.

A. Overall Architecture

*1) **Overview**:* This system is built to address a major gap in automated agriculture: the inability to manage and monitor plants at an individual level. Unlike conventional systems that treat plant beds or greenhouse zones uniformly—delivering water, nutrients, and care collectively—this system introduces a fully autonomous robotic solution that moves through the greenhouse, detects plant locations, and evaluates each plant's condition independently. The process begins with a mobile robot that follows a predefined network of black lines laid out on the greenhouse floor. These lines form a path with junctions and endpoints, each of which corresponds to the position of a plant. The robot navigates the path autonomously using an array of infrared (IR) sensors and a PID controller that continuously keeps it aligned to the center of the line, allowing it to detect intersections and station points accurately.

As the robot travels, it identifies when it has reached a plant by checking for the absence of the guiding line—an indication that it has reached a terminal point on the path. At this point, the robot halts its movement and engages a calibrated robotic arm mounted on its platform. The robotic arm performs a sequence of smooth, interpolated movements to lower an NPK sensor into the soil near the plant. This motion is carefully controlled using multiple servo motors that move in synchronization to avoid abrupt or damaging motions. The interpolation logic ensures that each servo gradually transitions from its current position to the target pose, mimicking the fluidity of human-like movements. Once in place, the arm holds the sensor steady for a defined duration to allow the sensor to collect accurate nutrient data from the soil.

The nutrient data is retrieved through the Modbus RTU protocol, commonly used in industrial sensor communication.

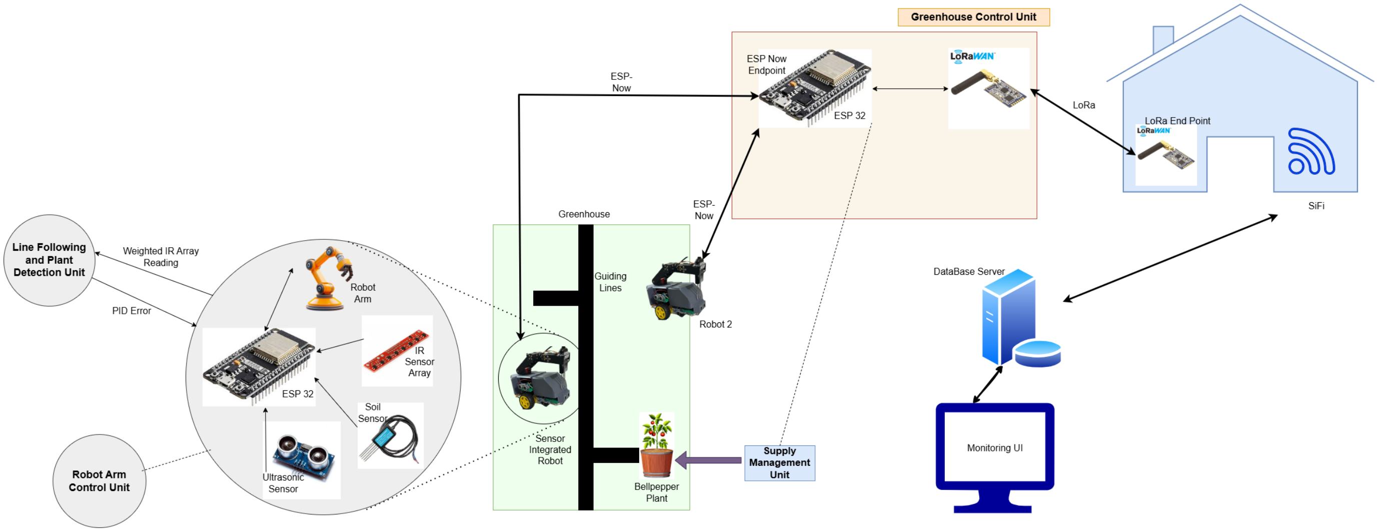

Fig. 1. Enter Caption

The arm transmits the readings to an onboard ESP32 microcontroller, which wirelessly uploads the data to a cloud-based database. This enables remote monitoring and logging of each plant's condition, providing a comprehensive, plant-by-plant status overview to greenhouse operators or automated decision-making systems.

Once the sensing is complete, the robot arm returns to its default position, and the robot reorients itself on the line to continue its journey. If a station is detected to be a terminal point—such as at the end of a branch in the path—the robot performs a 180-degree turn to resume line following in the opposite direction.

The system operates in a loop, moving between stations, performing nutrient sensing, and reporting back to the cloud. This architecture allows it to scale with minimal changes to the infrastructure—more plants simply require extending the physical track. Furthermore, because each plant is managed independently, the system supports adaptive control, where water or nutrients can be supplied based on real-time data specific to each plant's condition.

By integrating movement, sensing, and communication in a synchronized manner, the system achieves a level of granularity and intelligence that traditional greenhouse automation methods cannot offer.

*2) **PID Controlled Line Follower**:* The robot employs a Proportional-Integral-Derivative (PID) control algorithm to autonomously follow a predefined black line path laid across a white surface. An array of eight infrared (IR) sensors mounted at the front of the robot continuously monitors the contrast between the line and the background. Each sensor outputs binary data indicating the presence of the line, and a weighted average algorithm is used to calculate the line's relative position. The deviation of this position from the center generates an error value, which the PID controller processes in real-time. The proportional, integral, and derivative components adjust motor speeds on the left and right sides independently, enabling the robot to steer accurately and smoothly. This approach ensures minimal oscillation and quick correction in curves or misalignments. The system also includes logic to detect junctions and dead ends, allowing the robot to stop at designated plant locations for further action by the robotic arm.

B. Precise Robot Arm for The Insertion of Sensor

The robotic arm operates as a precision-controlled manipulator designed to individually assess each plant's soil conditions. Once the mobile robot arrives at a plant station, the arm initiates a predefined movement routine composed of interpolated servo positions. Each of the four servo motors governing the joints is controlled using a smooth linear interpolation technique, which gradually transitions the servos from one position to another over a fixed duration.

This ensures synchronized, fluid motion across all axes and prevents mechanical stress or positional inaccuracies. Upon reaching the target coordinates, the arm carefully lowers a soil sensor into the plant's root zone. The sensor is held stable for the duration of data acquisition to avoid noise due to motion. Communication between the arm and the soil sensor is handled via Modbus RTU protocol, allowing reliable, real-time transmission of NPK (Nitrogen, Phosphorus, Potassium) values to the onboard ESP32. After reading, the arm retracts gracefully and prepares for the next operation.

C. Handelling the NPK Sensor

The communication between the robotic arm's controller and the NPK soil sensor is established using the Modbus RTU protocol over a serial interface. Upon positioning the sensor in the soil, the ESP32 microcontroller sends a structured Modbus request frame to the sensor, specifying the register addresses corresponding to nitrogen, phosphorus, and potassium levels.

The sensor responds with a fixed-format data frame containing the measured values. This protocol ensures robust and reliable data exchange in noisy environments. The retrieved nutrient values are then parsed by the controller and transmitted to the cloud database for storage, analysis, and potential remote monitoring or control actions.

D. PID Line-Following Control Methodology

The mobile robot's navigation relies on a Proportional-Integral-Derivative (PID) controller that continuously adjusts the left and right motor speeds to maintain alignment with a black line path[10], [11]. The path-following is enabled by a linear infrared (IR) sensor array comprising eight equally spaced sensors mounted at the front of the robot[11], [12]. These sensors detect the contrast between the black line (logic 1) and the white background (logic 0).

At each iteration, the system reads the binary state $S = [S_0, S_1, \ldots, S_7]$, where each $S_i \in \{0, 1\}$ represents the presence of the line under the i^{th} sensor. A corresponding error weight vector $W = [-3, -2, -1, 0, 0, 1, 2, 3]$ is defined to reflect each sensor's deviation from the central axis of the robot[11].

The **error** at time t, denoted $e(t)$, is calculated as the **dot product** of the sensor array and weight vector:

$$e(t) = \frac{\sum_{i=0}^{7} S_i \cdot W_i}{\sum_{i=0}^{7} S_i}$$

This approach ensures that when the robot is centered on the line, the error is approximately zero. A positive error indicates deviation to the right, and a negative error indicates deviation to the left. The normalization by the number of active sensors prevents fluctuations due to varying numbers of detections[11], [10].

The PID correction term $C(t)$ is then computed using the standard PID formulation[13], [14]:

$$C(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau) d\tau + K_d \cdot \frac{de(t)}{dt}$$

Where:

- K_p, K_i, and K_d are the proportional, integral, and derivative gains respectively.
- $\int e(\tau) d\tau$ is the accumulated past error (integral).
- $\frac{de(t)}{dt}$ is the rate of change of the error (derivative), approximated using finite differences.

The correction is used to adjust the pulse-width modulation (PWM) signal applied to the motors:

$$LeftSpeed = V + C(t)$$

$$RightSpeed = V - C(t)$$

Where V is the base forward velocity. This dynamic correction ensures smooth and stable alignment even at turns or curves[12].

Fig. 2. Implemented Robot System

1) Turn and Junction Detection: To navigate through T-junctions or L-bends, the system includes logic for **junction detection** based on pattern recognition within the sensor readings[11], [14]. A **T-junction** is identified when both the leftmost (S_0 to S_3) and rightmost (S_4 to S_7) sets of sensors detect the line simultaneously. This is computed as:

$$L = S_0 \cdot S_1 \cdot S_2 \cdot S_3, \quad R = S_4 \cdot S_5 \cdot S_6 \cdot S_7$$

The conditions are:

- **T-junction**: $L = 1$ and $R = 1$
- **Left junction**: $L = 1$, $R = 0$
- **Right junction**: $R = 1$, $L = 0$

Upon detection, the robot halts momentarily, processes the junction type, and executes a predefined maneuver such as turning left, right, or proceeding straight. The turns are achieved through timed differential motor control using fixed-duration PWM pulses in opposite directions[10], [14].

2) Station (Plant) Detection and Stopping: A **plant station** is identified when **no sensor** in the array detects the line, i.e., $S_i = 0$ for all i. This is interpreted as a terminal point of a path segment. Mathematically:

$$\sum_{i=0}^{7} S_i = 0$$

Then it senses the distance to the plant in front of it using an Ultra Sound Distance sensor and positions itself in the correct distance to the plant to insert the NPK with the arm[14].

When this condition is met:

- The robot halts immediately.
- The `currentStation` counter is incremented.
- A signal is sent to activate the robotic arm module.
- After the robotic operation is complete, the robot performs a 180 turn to realign with the path and resumes movement.

This methodology allows the robot to not only follow lines autonomously but also to dynamically detect and interact with environmental cues for station-based processing[12].

subsectionRobotic Arm Control via Interpolated Servo Actuation

When the robot has reached the plant and is positioned in the proper distance, it sends a message through serial communication to the arm to start. The robotic arm used in this system is actuated using multiple servo motors, each responsible for one degree of freedom in the arm. These servos respond to PWM (Pulse Width Modulation) signals, where pulse durations typically range from 1000 µs (minimum angle) to 2400 µs (maximum angle). To achieve smooth and coordinated joint movement, the system employs linear interpolation between predefined poses rather than abrupt transitions [15], [16].

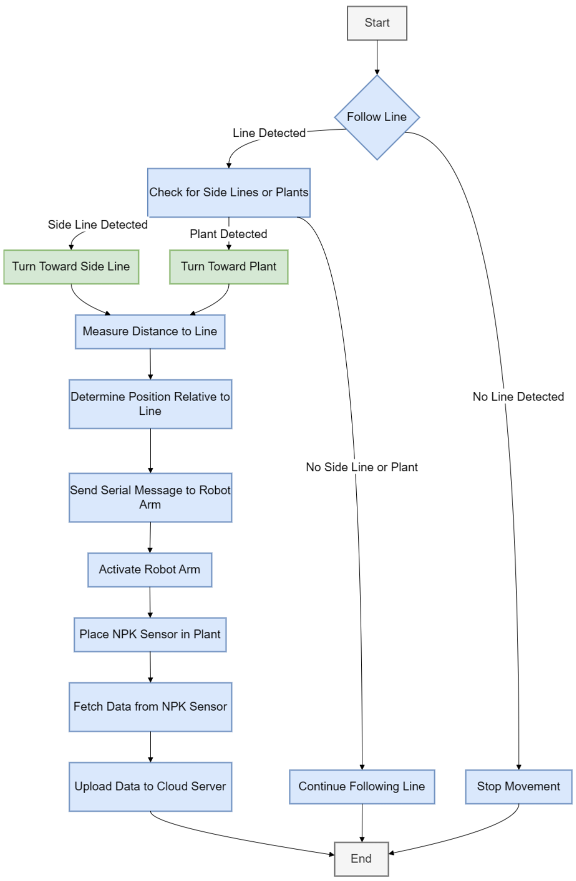

Fig. 3. Robot Arm Control with System Logic

Each movement of the arm is broken into discrete time steps for real-time control. Suppose a servo must move from an initial pulse width P_0 to a final pulse width P_f over a total duration T. If the system updates the servo position every Δt, the total number of steps N is given by:

$$N = \frac{T}{\Delta t}$$

The incremental change in position per step is:

$$\Delta P = \frac{P_f - P_0}{N}$$

At each time step i, the pulse width P_i sent to the servo is:

$$P_i = P_0 + i \cdot \Delta P, \quad for i = 0, 1, 2, \ldots, N$$

This formula ensures that the motion between two positions is distributed evenly over time, creating a smooth and gradual movement. Since each servo undergoes this interpolation independently, but under the same time constraints, the motion remains synchronized across all joints [16], [17]. This guarantees that the arm maintains its intended trajectory and posture throughout the movement.

Complex arm motions are implemented as sequences of such transitions between multiple poses. Each pose represents a distinct joint configuration defined by a set of PWM pulse widths. The robotic controller processes a list of these poses, interpolating between each pair to form a fluid motion path [17].

This interpolation-based methodology avoids mechanical jerk, reduces wear on servo motors, and ensures higher repeatability and reliability. It also enhances the robot's ability to interact delicately with individual plants, positioning the arm accurately before initiating any further sensing or actuation task. All computations and timing controls are managed by an onboard microcontroller that executes the interpolation routine and transmits the corresponding PWM signals in real time [15], [16].

E. Soil Nutrient Monitoring with NPK Sensor via Modbus RTU

The robot arm is equipped with an NPK sensor to measure the concentration of Nitrogen (N), Phosphorus (P), and Potassium (K) in the soil of individual plants. This sensor communicates with the system's microcontroller (ESP32) via the Modbus RTU protocol, a robust and widely used industrial serial communication standard.

Modbus RTU operates over a UART (Universal Asynchronous Receiver-Transmitter) interface. The ESP32 acts as the Modbus master and initiates communication by sending a structured query to the sensor, specifying the register address of the nutrient data it wants to read. Each query packet follows the Modbus format:

Modbus Packet:$[SlaveAddress] + [FunctionCode]$
$+[RegisterAddress] + [CRCCheck]$

The sensor (Modbus slave) responds with a data packet containing the requested value(s) along with a checksum to ensure data integrity. For instance, to read the nitrogen level, the ESP32 may issue a function code `0x03` (read holding registers) targeting the register assigned to nitrogen. The sensor's response is parsed to extract the numeric nutrient concentration, usually encoded in a two-byte or four-byte format.

After retrieving the NPK values, the ESP32 temporarily stores them in its internal memory. These values are then transmitted to a cloud-based database or IoT platform over WiFi. The data upload is performed using HTTP or MQTT protocols depending on the backend architecture. Each plant's data is tagged with metadata such as:

- Timestamp,
- Plant ID (inferred from position on the path),
- Robot ID.

IV. Results

The proposed system underwent extensive testing to evaluate the performance of each key component. Key areas of testing included PID tuning for the line-following module, calibration of the robot arm for precise soil measurements, reliability and accuracy of communication with the NPK sensor, and data transmission to the cloud.

A. PID Tuning

It takes an error signal derived from the position of the black line relative to the sensor array and calculates a corrective response. Tuning this controller required a methodical approach where each gain—Proportional (K_p), Derivative (K_d), and Integral (K_i)—was adjusted individually and then collectively to optimize performance.

1) Proportional Tuning (K_p): Tuning began with setting the **integral and derivative gains to zero**, isolating the effect of the proportional gain. K_p directly amplifies the error value and adjusts the wheel speeds accordingly. Initial tests with low values of K_p (around 10) resulted in slow and sluggish responses—the robot followed the line but reacted too late to curves and sharp turns.

As K_p was increased incrementally, the robot began to correct itself more aggressively. At $K_p = 30$, it was able to follow straight paths and gentle curves with high accuracy. However, if increased too much (e.g., $K_p = 50$), the robot began oscillating around the line, overcorrecting due to the large response to minor deviations.

2) Derivative Tuning (K_d): Once a stable K_p was established, derivative tuning was introduced. K_d accounts for the rate of change of the error, helping the robot anticipate upcoming deviations. Without derivative control, the robot often overshot turns and wobbled when correcting for curves. Starting with a low value of $K_d = 5$, the robot's steering smoothed out slightly. Increasing to $K_d = 10$ allowed the robot to take turns more gracefully, significantly reducing oscillations caused by the proportional gain. It effectively acted like a damping factor, countering the aggressive corrections from K_p.

3) Integral Tuning (K_i): Finally, the integral term was tuned. K_i compensates for small, persistent errors—especially useful when the robot drifts slightly over long distances. Early values of $K_i = 0.05$ provided minimal improvement, but with $K_i = 0.1$, the robot corrected slight misalignments that would otherwise have accumulated.

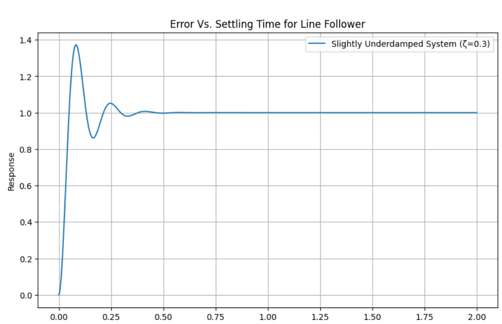

Fig. 4. Settling of PID Error over time

However, further increasing K_i (e.g., to 0.2) resulted in overcompensation and slow oscillations as the robot tried to "catch up" with integrated error.

Gain	Low Value Effects	Optimal Value Effects	High Value Effects
K_p	Slow, delayed correction	Accurate, responsive tracking	Oscillations and overcorrection
K_d	No anticipation, overshoots	Smooth turning, reduced oscillations	Jittery motion, may suppress K_p effect
K_i	Persistent drift in long runs	Fine correction over time, improved centering	Slow oscillations, cumulative overshoot

TABLE I
EFFECTS OF PID GAIN VALUES ON ROBOT PERFORMANCE

4) Summary of Effects: **Final Tuned Values:**

- $K_p = 30$
- $K_d = 10$
- $K_i = 0.1$

These provided the best balance between responsiveness and stability, allowing the robot to accurately track the line, smoothly handle curves, and quickly detect junctions or endpoints.

B. Robot Arm Calibration

The robot arm's ability to insert the NPK sensor into the soil with precision required careful calibration of the servos and motion sequences. Interpolation between servo positions ensured smooth transitions. The robot arm was calibrated using a multi-step process:

1) **Joint Calibration:** Each servo was manually adjusted to ensure it moved precisely through its full range of motion, avoiding mechanical binding or unnecessary force.
2) **Pose Calibration:** The robot arm's movements were synchronized to ensure that all joints moved together smoothly. The interpolation between servo positions was fine-tuned to ensure that each motion between key waypoints (e.g., reaching, positioning, and retracting) was fluid and precise.
3) **End Effector Calibration:** The sensor's insertion depth into the soil was carefully calibrated to achieve consistent positioning without damaging the plant or disturbing surrounding soil.

Calibration Results:
The robot arm was able to achieve a repeatable and consistent insertion depth, with variations of less than 2 mm between trials. This level of precision is sufficient for accurate soil nutrient measurements.

	Pot 1	Pot 2	Pot 3	Pot 4
Nitrogen (mg/Kg)	113	138	127	112
Phosphorus (mg/Kg)	97	116	110	92
Potassium (mg/Kg)	148	159	150	137

Fig. 5. NPK sensor readings

C. Communication with NPK Sensor

The system uses Modbus RTU [18] for communication between the ESP32 microcontroller and the NPK sensor. The Modbus protocol allows for reliable data transmission and is widely used in industrial settings for real-time sensor communication.

Communication Results:
The NPK sensor successfully provided real-time readings of Nitrogen (N), Phosphorus (P), and Potassium (K) levels, with a response time of around 200 ms per query. The accuracy of the sensor was verified by comparing it against manual soil testing. The deviation between the sensor's reading and manual tests was found to be consistently below 5%. This level of accuracy is acceptable for most agricultural applications, especially when used for monitoring trends over time rather than precise one-time measurements.

CONCLUSION

The system successfully addressed the problem of monitoring individual plants in an automated manner, offering a significant advantage over traditional zone-based agricultural systems. Future improvements could focus on refining PID tuning further, expanding the robot's ability to handle more complex paths, and enhancing cloud data processing with advanced machine learning algorithms for predictive analysis.

REFERENCES

[1] Haifa Group, "Crop guide: Nutrients for pepper," https://www.haifa-group.com/articles/crop-guide-nutrients-pepper, 2025, accessed: April 25, 2025.

[2] M. S. Farooq, R. Javid, S. Riaz, and Z. Atal, "IoT Based Smart Greenhouse Framework and Control Strategies for Sustainable Agriculture," *IEEE Access*, vol. 10, pp. 99 394–99 420, 2022.

[3] M. A. Rahman, P. Chakroborty, and M. F. Ali, "Design and implementation of a low-cost arduino based line follower gardener robot," in *2022 International Conference on Recent Progresses in Science, Engineering and Technology (ICRPSET)*. Rajshahi, Bangladesh: IEEE, 2022, pp. 1–4.

[4] M. M. Almasri, K. M. Elleithy, and A. M. Alajlan, "Development of efficient obstacle avoidance and line following mobile robot with the integration of fuzzy logic system in static and dynamic environments," in *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*. Farmingdale, NY, USA: IEEE, 2016, pp. 1–6.

[5] P. Valsalan and P. Surendran, "Implementation of an emergency indicating line follower and obstacle avoiding robot," in *2019 16th International Multi-Conference on Systems, Signals & Devices (SSD)*. Istanbul, Turkey: IEEE, 2019, pp. 479–482.

[6] J. Y. Qi, L. C. Hung, H. S. Jo, J. S. W. Kiat, E. L. W. Ting, and M. Dunn, "Robotic automation system for indoor greenhouse applications," in *2024 IEEE 12th Region 10 Humanitarian Technology Conference (R10-HTC)*. Kuala Lumpur, Malaysia: IEEE, 2024, pp. 1–5.

[7] P. Musa, H. Sugeru, and E. Prasetyo, "Wireless sensor networks for precision agriculture: A review of npk sensor implementations," *Sensors*, vol. 24, no. 1, p. 51, 2023.

[8] N. Yeasdani, M. Bhuiyan, S. Ifte, Adnan, L. Mannan, A. Harun, H. Pranto, M. Hasan, and N. Chisty, "Enhancing agricultural productivity through a semi-autonomous iot robot in smart farming systems," *Bangladesh Journal of Agriculture*, vol. 48, no. 2, pp. 94–105, Dec. 2023.

[9] F. Guangrui and W. Geng, "Vision-based autonomous docking and recharging system for mobile robot in warehouse environment," in *2017 2nd International Conference on Robotics and Automation Engineering (ICRAE)*, Shanghai, China, 2017, pp. 79–83.

[10] K. K. S. Kumar *et al.*, "Pid controller-based line following and obstacle avoidance robot," in *Proc. Int. Conf. Robotics Autom.*, 2011, pp. 1966–1971.

[11] R. Godse, A. Murkute, and P. R. Kharat, "Pid controllers for line tracing robots," in *Proc. IEEE Int. Conf. Control Syst.*, 2020, pp. 1–6.

[12] S. F. Jibrail and R. Maharana, "Pid control of line followers," *IEEE Trans. Educ.*, vol. 56, no. 3, pp. 332–338, 2013.

[13] U. o. M. Systems Engineering Department, "Design and implementation of line follower arduino mobile robot," University of Mosul, Tech. Rep., 2021.

[14] A. S. Nath, A. K. R, and T. Malik, "Implementation of pid control to reduce wobbling in a line following robot," in *Proc. IEEE Int. Conf. Mechatronics*, 2013, pp. 531–535.

[15] J. H. Park and J. H. Kim, "Trajectory planning for a robotic manipulator using cubic spline interpolation," in *2008 IEEE International Conference on Robotics and Biomimetics*, 2008, pp. 1862–1867.

[16] J. H. Kim, S. K. Kim, and J.-B. Song, "Smooth trajectory generation for robot manipulators using minimum jerk and cubic spline interpolation," in *2012 12th International Conference on Control, Automation and Systems (ICCAS)*, 2012, pp. 1957–1961.

[17] Z. Li, H. Liu, S. Liu, and Y. Hou, "Smooth trajectory generation for multi-joint robot arm using quintic polynomial interpolation," in *2016 IEEE International Conference on Information and Automation (ICIA)*, 2016, pp. 1811–1816.

[18] W. You and H. Ge, "Design and implementation of modbus protocol for intelligent building security," in *2019 IEEE 19th International Conference on Communication Technology (ICCT)*, Xi'an, China, 2019, pp. 420–423.