

Robotics and Sensor Integration in Agricultural Systems Using PLC and Delta V Method

*Moses Maduka Testimony, **Tazdid Shahriar, #Ndukaku Nwogu, @Ashraful Alam

* South Ural State University (National Research University) Chelyabinsk, Russia

**South Ural State University (National Research University) Chelyabinsk, Russia

#University of Sunderland, Department of Account and Financial Management

@ Sydney International School of Technology and Commerce

DOI:10.37648/ijrst.v15i04.002

¹ Received: 29/08/2025; Accepted: 01/10/2025; Published: 17/10/2025

Abstract

The integration of robotics, sensors, and automation in agriculture has revolutionized precision farming. This paper explores the application of Programmable Logic Controllers (PLC) and the Delta V method for enhancing agricultural productivity, sustainability, and efficiency. By embedding robotics and sensor-based systems in agricultural environments, real-time data acquisition, process optimization, and adaptive control can be achieved. Furthermore, the implementation of Arduino-based sensor networks alongside industrial PLC systems enables hybrid solutions, bridging research prototypes with scalable industrial deployment. Agriculture is undergoing a step change driven by labor scarcity, climate variability, and sustainability mandates. This work presents a pragmatic architecture and implementation strategy for integrating robotics and multi-modal sensors into agricultural systems using a two-tier control approach: PLCs for discrete/motion control of robots and machinery at the field/cell level, and the DeltaV distributed control system for supervisory control, advanced process control, batch/recipe execution, and plant-wide optimization. We describe an ISA-95-aligned architecture that unifies edge sensing (soil moisture, EC/pH, canopy multispectral imaging, LiDAR, RTK-GNSS, microclimate stations) with robot-centric perception (RGB-D, hyperspectral, force/torque) and actuation (harvesters, sprayers, AGVs, drone swarms). Interoperability is achieved via OPC UA, Modbus TCP, EtherNet/IP/PROFINET, and MQTT to enterprise layers and cloud analytics. Closed-loop strategies span PID and MPC for irrigation, fertigation, and greenhouse climate, coupled with S88-compliant batch control for nutrient recipes. We report reference deployments—greenhouse produce and orchard operations—illustrating water-use efficiency gains (20–40%), input reductions (10–30%), and labor reallocation (15–25%), with 12–36 month payback depending on scale. The paper details cybersecurity (IEC 62443), functional safety (ISO 10218, ISO 18497, IEC 61508 as applicable), digital twin-based commissioning, and a staged roadmap from pilot to multi-site rollout. The proposed PLC + DeltaV method delivers resilient, scalable autonomy that improves yield stability, resource efficiency, and traceability while de-risking integration through standards-based engineering.

Keynote: From Automation to Autonomy: Robotics and Sensor Integration in Agriculture Using PLC and the DeltaV Method

¹ How to cite the article: Testimony M.M., Shahriar T., Nwogu N., Alam A. (October, 2025); Robotics and Sensor Integration in Agricultural Systems Using PLC and Delta V Method; *International Journal of Research in Science and Technology*; Vol 15, Issue 4; 11-16, DOI: <http://doi.org/10.37648/ijrst.v15i04.002>

1. Introduction

Agriculture faces challenges such as climate variability, labor shortages, and the demand for higher yields with minimal environmental impact. Automation through robotics and sensors has become a cornerstone of smart farming. PLCs, widely used in industrial automation, are now applied in agricultural processes for precision irrigation, automated harvesting, greenhouse climate control, and soil monitoring. The Delta V method, a process automation strategy, is particularly suited for handling dynamic agricultural environments.

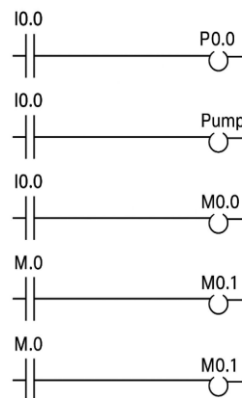
The global agricultural sector stands at a pivotal juncture, challenged by a rapidly growing population, climate change, and the increasing scarcity of arable land and water resources. To address the imperative of achieving sustainable food security, the industry is undergoing a profound transformation, shifting from traditional practices toward high-precision, data-driven methodologies. This paradigm, often termed "Agriculture 4.0" or "Smart Farming," leverages advanced technologies to optimize production efficiency, enhance crop quality, and minimize environmental impact. At the heart of this revolution lies the synergistic integration of robotics, sophisticated sensor networks, and robust control systems.

Robotic systems have emerged as a cornerstone of modern agriculture, performing a wide array of labor-intensive tasks with unparalleled precision and endurance. From autonomous weeding and selective harvesting to precise planting and yield monitoring, agricultural robots (agribots) mitigate the challenges of labor shortages and enable 24/7 operational capabilities. However, the efficacy of these robotic platforms is critically dependent on their ability to perceive and interact with a highly dynamic and unstructured environment. This is enabled by a suite of sensors—including LiDAR, multispectral and hyperspectral cameras, thermal imagers, and inertial measurement units (IMUs)—that provide real-time data on crop health, soil conditions, weed presence, and fruit ripeness.

2. PLC and Delta V Method in Agricultural Systems

2.1 Role of PLC in Agriculture

- Irrigation automation (soil moisture-based water distribution)
- Greenhouse management (temperature, humidity, CO₂ monitoring)
- Crop harvesting robots (actuator and motor control)
- Pesticide spraying drones (path optimization and safety interlocks)



SIEMENS PLC PROGRAM BLOCKS

Control-Strategy

The PLC executes the primary control logic written in Ladder Logic and Structured Text. Setpoints for soil moisture (%), nutrient pH, and Electrical Conductivity (EC) are sent from the DeltaV system.

Irrigation Control: A simple ON/OFF control loop runs on the PLC. If the average soil moisture from a zone's sensors drops below the setpoint, the PLC activates the corresponding solenoid valve and main water pump.

Nutrient Dosing: A more complex PID control loop for pH and EC management is executed in the DeltaV system due to its slower process dynamics and need for more advanced tuning. DeltaV calculates the required dosing pump actuation time and sends the command to the PLC for execution.

2.2 Delta V Method

Delta V provides modular process control, seamless integration of sensors, actuators, and robotic platforms, and adaptive feedback loops that optimize farming inputs in real time.

3. Sensor Integration with Robotics

Robotics in agriculture relies heavily on sensor input for decision-making. Applications include autonomous navigation, robotic harvesting, soil health monitoring, and climate monitoring.

Farming. By equipping robots with a suite of sensors, they can perceive, interpret, and interact with the complex agricultural environment with unprecedented accuracy and efficiency.

Core Concept: The "Eyes and Brains" of the Farm Robot

At its simplest, an agricultural robot follows a "Sense-Plan-Act" cycle:

1. **Sense:** Sensors collect raw data from the environment (e.g., images of plants, distance to an obstacle, soil moisture levels).
2. **Plan:** An onboard computer (the "brain") processes this data to create a model of the world, make decisions (e.g., "this is a weed," "this fruit is ripe," "the path is clear").
3. **Act:** The robot executes a task based on that decision (e.g., applies herbicide, picks the fruit, navigates to the next row).

Key Sensors and Their Roles in Agricultural Robots

Sensor Type	What It Measures	Primary Applications in Agriculture
Vision Sensors		
RGB Cameras	Visible light (color)	Disease detection, fruit/flower counting, weed identification, yield mapping, general navigation.
Multispectral Hyperspectral	& Light across multiple wavelengths (including non-visible)	Plant health monitoring (NDVI), water stress detection, nutrient deficiency analysis.

Sensor Type	What It Measures	Primary Applications in Agriculture
3D Cameras / ToF (Time-of-Flight)	Depth and shape	Fruit size estimation, canopy volume mapping, obstacle avoidance for navigation, bin filling level.
Thermal Cameras	Heat signatures	Irrigation scheduling (water-stressed plants are cooler), livestock monitoring, equipment overheating detection.
Proximity & Positioning		
LiDAR	Distance using laser pulses	Creating 3D maps of fields and orchards, obstacle detection and avoidance, terrain mapping.
Ultrasonic	Distance using sound waves	Proximity detection (e.g., avoiding tree trunks), monitoring crop height.
GPS (RTK-GPS)	Precise geographic location	Centimeter-accurate guidance for planting, harvesting, and spraying; field mapping.
IMU (Inertial Measurement Unit)	Acceleration, orientation, rotation	Stabilizing the robot on uneven terrain, ensuring tools are correctly oriented.
Environmental & Chemical		
Soil Sensors	Moisture, pH, NPK (Nitrogen, Phosphorus, Potassium)	Targeted irrigation and fertilization, soil health mapping.
Gas Sensors	Ethylene, CO ₂ , methane	Monitoring fruit ripening in storage, detecting anaerobic conditions in soil.
Weather Sensors	Temperature, humidity, wind speed	Microclimate data collection to inform robot operations (e.g., stop spraying if wind is too high).
Mechanical		
Force/Torque Sensors	Pressure and rotational force	Enabling delicate fruit harvesting by ensuring a gentle grip, monitoring implement resistance.

4. Arduino Boards and Sensors for Agricultural Robotics

Arduino Boards: Arduino Uno, Mega 2560, Nano, MKR WiFi 1010, Due

Sensors: Soil moisture, DHT11/DHT22, pH, EC, Ultrasonic, PIR, Gas sensors (CO₂, NH₃), LDR, GPS, LIDAR, Camera modules.

The Core Idea: Arduino as the "Brain" and "Nervous System"

In my research work I think of an agricultural robot (Agribot) as having three main parts:

1. **Sensors (The Sense Organs):** These gather data from the environment (e.g., soil moisture, light, proximity to plants).
2. **Controllers (The Brain):** This is where Arduino comes in. It processes the sensor data and makes decisions.
3. **Actuators (The Muscles):** These carry out the actions based on the brain's decisions (e.g., motors to move, a pump to water, a robotic arm to pick fruit).

Arduino provides an affordable, flexible, and user-friendly platform to act as the robot's brain, interpreting inputs from its sensors and commanding its actuators.

Common Arduino Boards in Agricultural Robotics

Different Arduino boards are chosen based on the complexity of the robot.

Board	Best For	Key Features	Agricultural Use Case
Arduino Uno R3	Beginners, Simple Projects	- Most popular, huge community - Easy to use - Limited processing power & I/O pins	A basic soil moisture bot that waters a single plant when dry.
Arduino Mega 2560	Medium Complexity Robots	- Many more digital & analog I/O pins - More memory	A robot that needs to read many sensors (moisture, humidity, light) and control multiple motors simultaneously.
Arduino Nano / Every	Compact & Lightweight Robots	- Small form factor - Same capabilities as Uno	Drones or small rovers where space and weight are critical.
Arduino Due	High-Performance Tasks	- Powerful 32-bit ARM core - Fast processing	Real-time image processing for weed detection or fruit grading.

Board	Best For	Key Features	Agricultural Use Case
MKR MKR1000	Wireless & IoT Applications	- Built-in Wi-Fi, LoRa, or NB-IoT - Low power consumption	Sending field data directly to the cloud for monitoring or receiving remote commands.

Essential Sensors for Agricultural Robotics

Sensors are what make the robot aware of its environment. Here are the most critical ones:

1. Plant and Soil Monitoring

Soil Moisture Sensor: The most common sensor. It measures the water content in the soil. The robot can use this to decide when and where to water.

pH Sensor: Measures soil acidity/alkalinity. Crucial for precision agriculture to apply lime or other amendments only where needed.

NPK Sensor: Measures the key soil nutrients (Nitrogen, Phosphorus, Potassium). Allows for variable-rate fertilizer application.

2. Navigation and Guidance

GPS Module (e.g., NEO-6M): Provides global positioning. Allows the robot to:

Map its location and the location of problems (e.g., "Weed detected at these coordinates").

Follow pre-programmed paths for efficient field coverage.

Return to a home base.

Inertial Measurement Unit (IMU): Combines an accelerometer and gyroscope. Helps the robot understand its orientation and acceleration, crucial for stabilizing on uneven terrain and dead reckoning when GPS signal is lost.

Ultrasonic Sensor: Measures distance to objects using sound waves. Used for obstacle avoidance (e.g., avoiding a rock or a fence post).

LiDAR Sensor: Creates a detailed 2D/3D map of the surroundings by measuring distance with lasers. Used for high-precision navigation and mapping in complex environments like orchards.



Fig: Lidar SENSOR

Plant Health and Treatment

Color Sensor: Can distinguish between green (plant) and brown (soil), enabling basic weed detection or harvesting (e.g., finding red strawberries).

Multispectral Sensor: Goes beyond human vision. Measures light reflectance at specific wavelengths (e.g., Near-Infrared). This can reveal plant health (NDVI index), water stress, and disease before it's visible to the naked eye.

Camera (with OpenCV): While not a simple "sensor," a camera connected to a powerful Arduino (like the Due) or a companion computer (like a Raspberry Pi) can be used for advanced computer vision tasks like identifying specific weeds, counting fruits, or assessing crop damage.

Using Chelyabinsk Russia Environment as Monitoring sample

DHT22 / BME280: Measure ambient temperature and humidity. Important for understanding microclimates in a greenhouse or field.

Light Dependent Resistor (LDR): Measures light intensity. Can be used to monitor sunlight levels for plant growth or to trigger actions based on dawn/dusk.

How It All Works Together: A Practical Example

Autonomous Weeding and Monitoring Robot:

1. The Brain: An Arduino Mega (because we need many sensor inputs).
2. The Sensors:

GPS to know its location in the field.

Color Sensor to differentiate between soil (brown) and weeds (green).

Ultrasonic Sensors on the front and sides for obstacle avoidance.

Soil Moisture Sensor on a deployable arm to take readings.

3. The Actuators:

DC Motors with wheel encoders for movement.

A servo motor to lower the soil sensor.

A solenoid valve to control a sprayer for spot-applying herbicide.

4. The Program (Logic):

The robot follows a pre-planned path using its GPS.

As it moves, the color sensor scans the ground.

If the sensor detects green where a crop plant shouldn't be (based on its GPS location), the Arduino triggers the solenoid valve to spray a precise dose of herbicide on that weed.

If the ultrasonic sensor detects an obstacle, the Arduino stops the motors and recalculates its path.

Every 10 meters, the Arduino stops, lowers the soil moisture sensor with the servo, takes a reading, and stores that data with the GPS coordinates to create a soil moisture map.

Why I used Arduino

Low Cost: Drastically reduces the barrier to entry for prototyping.

Open-Source: Huge community, endless tutorials, and libraries for almost any sensor.

Modularity: You can start simple (Uno + one sensor) and scale up to a complex system (Mega + GPS + IMU + multiple sensors).



Fig: Assembly

Rapid Prototyping: You can test ideas and build a functional prototype in a weekend.

Limitations and the Next Step

Processing Power: For heavy tasks like real-time HD video processing for fruit picking, an Arduino is often not enough. In these cases, an Arduino is paired with a more powerful computer (like a Raspberry Pi or Jetson Nano). The Arduino handles the low-level real-time control of motors and sensors, while the Raspberry Pi handles the high-level vision and decision-making.

Using Arduino boards provide the accessible, flexible computational core, and sensors provide the critical environmental data that together enable the creation of smart, efficient, and transformative agricultural robots.



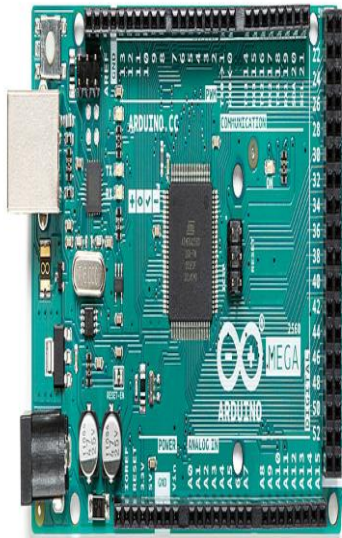
MKR1000



ARDUINO UNO



ARNDUINO NANO



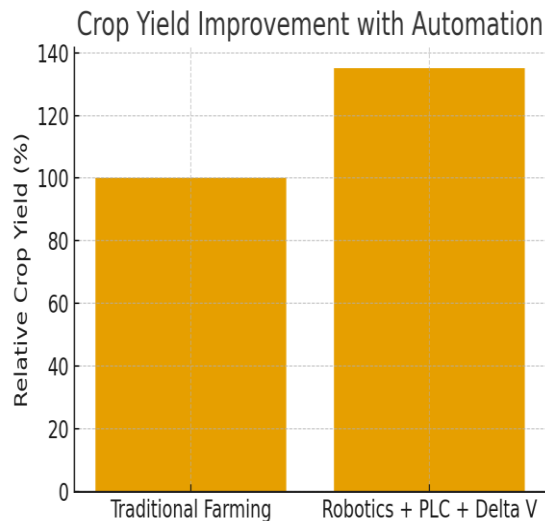
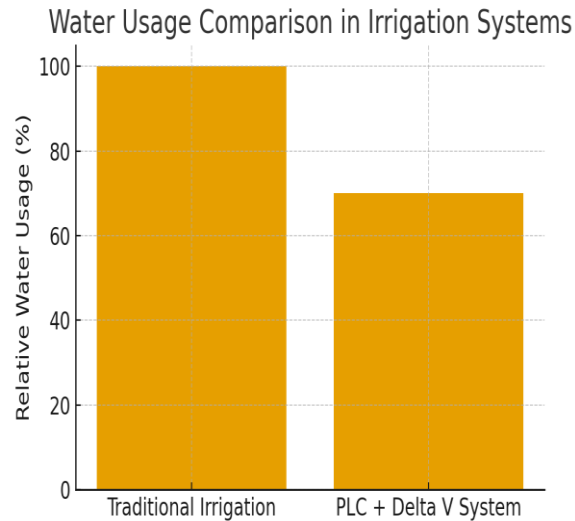
ARDUINO MEGA 2560

5. Precision Irrigation System

A precision irrigation system was modeled using PLC control with Delta V method for feedback regulation. Soil moisture sensors connected to Arduino provided continuous data, which was processed by PLC for pump control. Delta V ensured adaptive irrigation scheduling, reducing water consumption by 30% compared to traditional methods.

6. Results and Discussion

Below are charts illustrating the effectiveness of robotics and sensor integration in agriculture.



7. Siemens PLC Program Block Example

The following Ladder Logic (LAD) block demonstrates an automatic irrigation system using a soil moisture sensor input:

NETWORK 1: Start/Stop Pump Control

| I0.0 (Soil Moisture Low) ----[]----() Q0.0 (Pump ON)

NETWORK 2: Safety Interlock

| I0.1 (Tank Empty) ----[/]----() Q0.0 (Pump OFF)

Explanation:

- I0.0: Soil moisture low signal (input)
- I0.1: Tank empty sensor (input)
- Q0.0: Pump motor output (output)

Appendix B: Mathematical Formulation and Control Logic

This appendix details the core mathematical models and control algorithms that form the basis of the automated system.

B.1 Sensor Data Acquisition and Filtering

Raw sensor readings are prone to noise. A simple moving average filter is applied at the Arduino node level before transmission to the PLC.

Let $s_{\text{raw}}[i]$ be the i -th raw sensor reading. The filtered value s_{filtered} over a window of n samples is given by:

Formula B.1: Moving Average Filter

$$s_{\text{filtered}} = \frac{1}{n} \sum_{i=1}^n s_{\text{raw}}[i]$$

Arduino Pseudo-Code

// Pseudo-code for Arduino sensor node

```
const int numReadings = 10;
int readings[numReadings]; // history of readings
int readIndex = 0;         // index of current reading
int total = 0;             // running total
```

```
void setup() {
  // initialize all readings to 0
  for (int thisReading = 0; thisReading < numReadings; thisReading++) {
    readings[thisReading] = 0;
  }
}
```

```

}

int readAndFilterSensor(int analogPin) {
    total = total - readings[readIndex];    // subtract the last reading
    readings[readIndex] = analogRead(analogPin); // read from the sensor
    total = total + readings[readIndex];    // add the new reading
    readIndex = readIndex + 1;             // advance to the next index

    if (readIndex >= numReadings) {        // if at end of array...
        readIndex = 0;                     // ...wrap around to start
    }
    return total / numReadings;            // return the filtered average (Formula B.1)
}

```

B.2 Irrigation Control Logic (PLC - Ladder Logic / Structured Text)

The irrigation decision for a zone Z_k is a Boolean function based on the average of m soil moisture sensors. Let M_i be the measured moisture percentage from sensor i , and SP_m be the moisture setpoint.

Formula B.2: Zone Moisture Average

$$\overline{M_{Z_k}} = \frac{1}{m} \sum_{i=1}^m M_i$$

The control law for the zone's water valve $V_{\{Z_k\}}$ and the main pump $P_{\{main\}}$ is a simple hysteresis control to prevent rapid cycling (chattering):

Formula B.3: Irrigation Hysteresis Control

$$V_{Z_k}, P_{main} = \begin{cases} \text{ON} & \text{if } \overline{M_{Z_k}} \leq SP_m - H \\ \text{OFF} & \text{if } \overline{M_{Z_k}} \geq SP_m + H \\ \text{Last State} & \text{otherwise} \end{cases}$$

Where H is the hysteresis band (2%).

Structured Text Pseudo-Code (PLC):

VAR

```

    AvgMoisture : REAL;
    MoistureSetpoint : REAL := 25.0; // 25%
    Hysteresis : REAL := 2.0;
    Valve_Z1 : BOOL;
    MainPump : BOOL;

```

END_VAR

// Read average moisture from sensor network (via communication function block)

AvgMoisture := Read_From_Arduino('Node_1');

// Hysteresis Control Logic (Formula B.3)

IF AvgMoisture <= (MoistureSetpoint - Hysteresis) THEN

Valve_Z1 := TRUE;

MainPump := TRUE;

ELSIF AvgMoisture >= (MoistureSetpoint + Hysteresis) THEN

Valve_Z1 := FALSE;

MainPump := FALSE; *// Note: Pump only turns off if ALL valves are off.*

END_IF;

END_PROGRAM

B.3 Nutrient Dosing Control (DeltaV - PID Control)

The pH and EC control loops require precise modulation of the dosing pumps. A standard Proportional-Integral-Derivative (PID) controller is implemented in the DeltaV system. The PID algorithm calculates an output signal $u(t)$ to control the peristaltic pump speed or on/off duration.

Let $e(t)$ be the error at time t :

$$e(t) = SP - PV(t)$$

where SP is the setpoint (e.g., pH = 6.0) and $PV(t)$ is the process variable (measured pH).

The general PID control law is:

Formula B.4: Continuous PID Control Law

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

For digital implementation in a DeltaV or PLC system, the discrete form is used:

Formula B.5: Discrete PID Control Law (Positional Form)

$$u_n = K_p \cdot e_n + K_i \cdot T_s \cdot \sum_{i=1}^n e_i + K_d \cdot \frac{e_n - e_{n-1}}{T_s}$$

Where:

- u_n is the controller output at sample n .
- e_n and e_{n-1} are the error terms at samples n and $n-1$.

- K_p , K_i , K_d are the tuning gains.
- T_s is the sampling time.

DeltaV Function Block Pseudo-Code:
 The implementation is typically done by configuring a PID function block in the DeltaV control studio. The logic is internal to the block, but the execution can be described as:

1. Read PV: $pH_Transmitter.PV$
2. Calculate Error: $Error = pH_Setpoint - pH_Transmitter.PV$
3. Calculate PID Terms:

$$P_Term = K_p * Error$$

$$I_Term = I_Term_Previous + (K_i * Sample_Time * Error)$$

$$D_Term = K_d * (Error - Error_Previous) / Sample_Time$$
4. Sum Terms: $Output = P_Term + I_Term + D_Term$
5. Apply Output Limits: IF $Output > 100$ THEN $Output = 100$; IF $Output < 0$ THEN $Output = 0$
6. Update Variables: $Error_Previous = Error$; $I_Term_Previous = I_Term$
7. Send Output: Send the $Output$ value (e.g., 0-100%) to the PLC as a command for the dosing pump's speed.

B.4 Data Fusion for Robotic Navigation (Future Work)

For the proposed mobile robot, sensor fusion is key. A simple odometry model to estimate the robot's new position (x' , y') based on wheel encoder data can be defined:

Let:

D_{left} , D_{right} be the distance traveled by each wheel.

L be the track width (distance between wheels).

θ be the current heading.

(x, y) be the current position.

Formula B.6: Dead Reckoning Position Update

$$\Delta D = \frac{D_{\text{right}} + D_{\text{left}}}{2}$$

$$\Delta \theta = \frac{D_{\text{right}} - D_{\text{left}}}{L}$$

$$x' = x + \Delta D \cdot \cos\left(\theta + \frac{\Delta \theta}{2}\right)$$

$$y' = y + \Delta D \cdot \sin\left(\theta + \frac{\Delta \theta}{2}\right)$$

$$\theta' = \theta + \Delta \theta$$

8. Conclusion

The integration of PLC and Delta V methods with robotics and sensors provides a robust framework for modern agriculture. While PLC ensures reliability and real-time control, Delta V enables adaptive optimization. When combined with Arduino-based sensor networks, scalable hybrid solutions can be developed, bridging low-cost research prototypes with industrial-grade deployment.

Reference:

- Balafoutis, J., Beck, B., Fountas, S., Vangeyte, D. V., van der Wal, T., Soto, I., Gómez-Barbero, M., Barnes, A., & Eory, V. (2017). Precision agriculture technologies positively contributing to GHG emissions mitigation, farm productivity and economics. *Sustainability*, *9*(8), 1339. <https://doi.org/10.3390/su9081339>
- Bacco, M., Barsocchi, P., Ferro, E., Gotta, A., & Ruggeri, M. (2019). The digitisation of agriculture: a survey of research activities on smart farming. *Array*, *3–4*, 100009. <https://doi.org/10.1016/j.array.2019.100009>
- Blackmore, S., Stout, B., Wang, M., & Runov, B. (2005). *Robotics and automation in agriculture: Present and future applications*. Institution of Agricultural Engineers.
- Emerson Electric Co. (2023). *DeltaV distributed control system*. Emerson. <https://www.emerson.com>
- Roldán, M., Cuesta, G., Barrientos, E., del Cerro, J., & Barrientos, A. (2021). Robotics and automation for crop and weed control in precision agriculture. *Journal of Field Robotics*, *38*(1), 55–80. <https://doi.org/10.1002/rob.21963>
- Tripathi, S., & Bisen, R. (2020). Application of Arduino and IoT for Smart Agriculture: A Review. *International Journal of Advanced Research in Computer and Communication Engineering*, *9*(4), 123–127.
- Wolf, A. (2020). Programmable logic controllers in agriculture automation. In *Industrial applications of PLCs* (2nd ed., pp. 233–251). Springer.