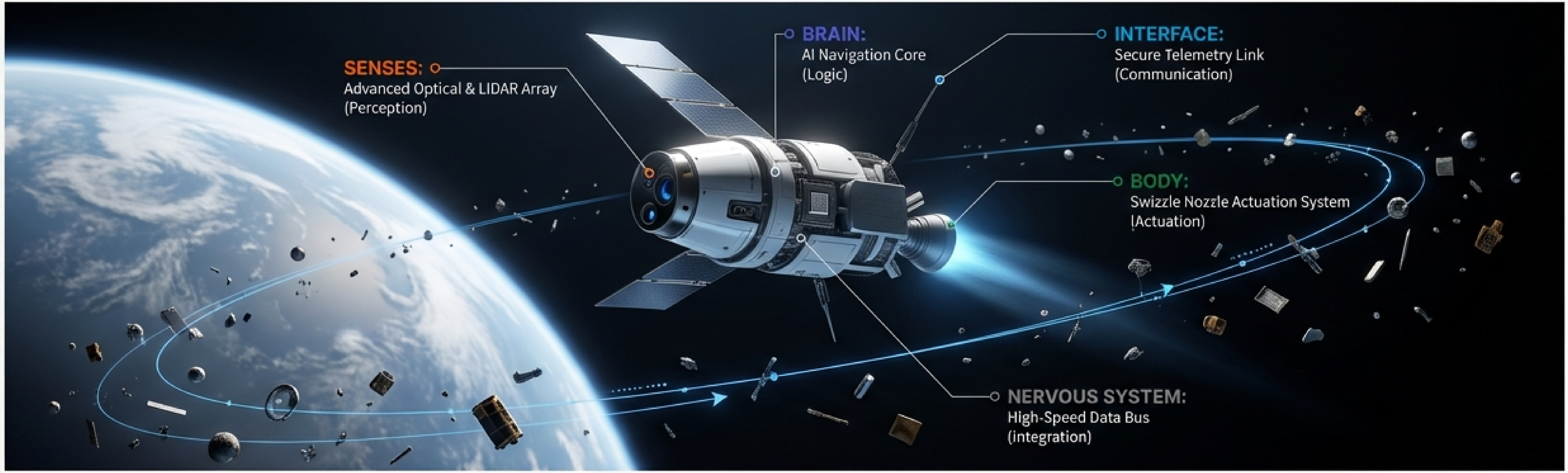


AI-POWEREDWERED SWIZZLE NOZZLE

The Anatomy of an Autonomous Debris Avoidance System

A prototype by Sanchit ('Dark'), simulating next-generation spacecraft navigation.

SENSES (Perception)

Real-time debris detection and tracking via multi-spectral sensors.

BRAIN (Logic)

Autonomous path planning and collision avoidance algorithms.

BODY (Actuation)

Precise thrust vectoring for rapid trajectory adjustments.

NERVOUS SYSTEM (Integration)

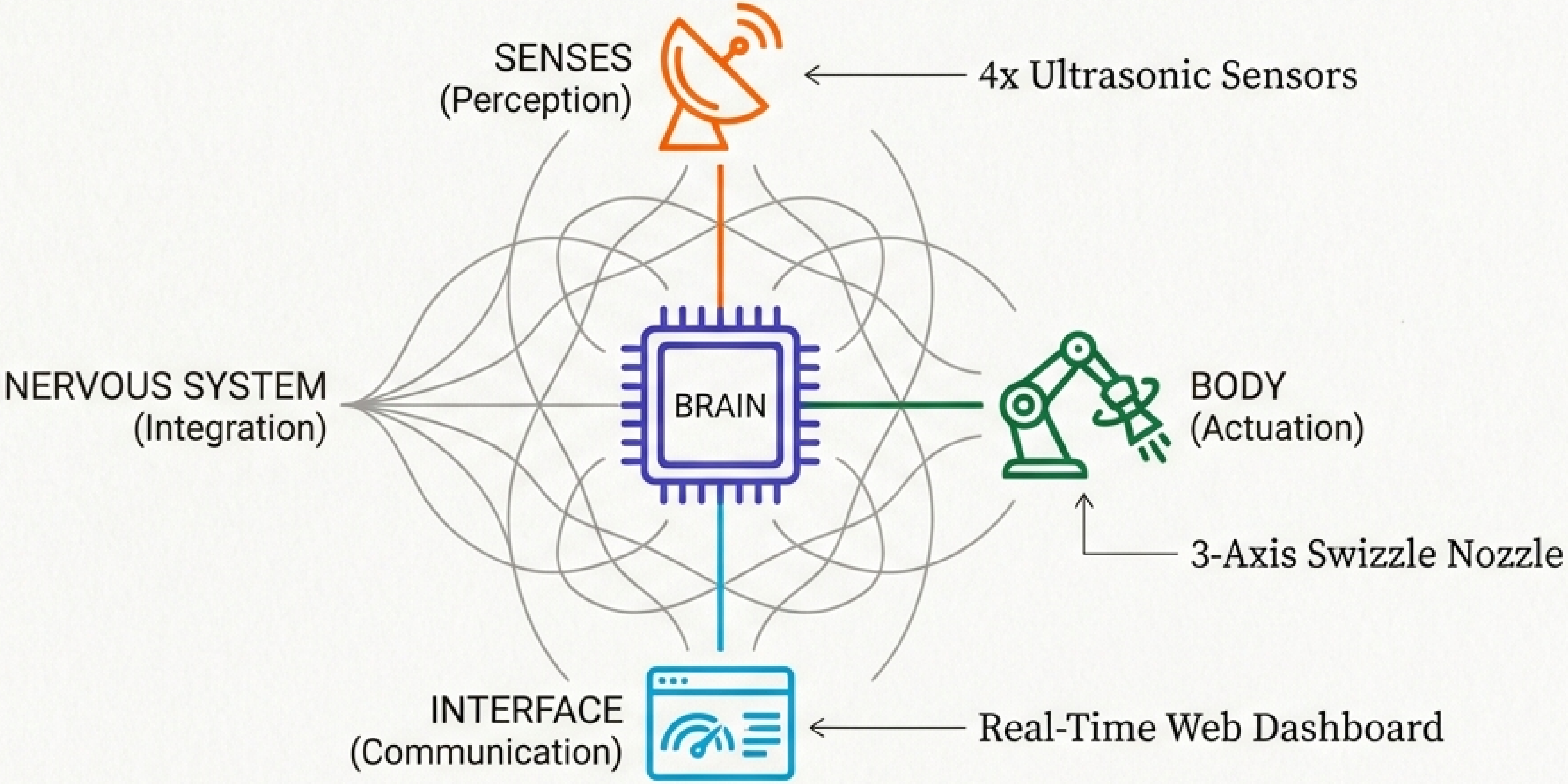
Low-latency signal processing and sensor fusion.

INTERFACE (Communication)

Telemetry and command downlink for mission control monitoring.

NotebookLM

A Purpose-Built Agent on a Critical Mission

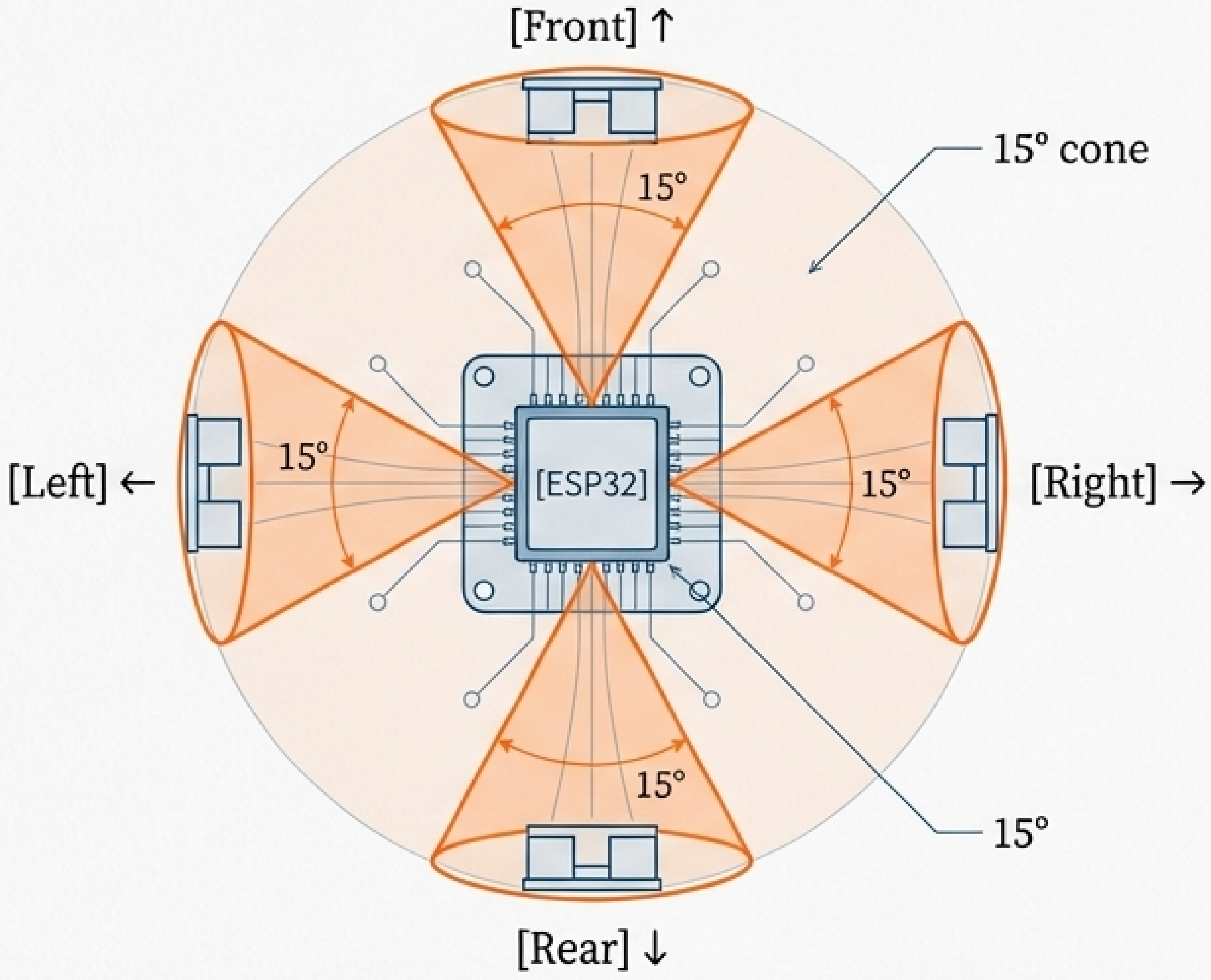


This presentation dissects the Swizzle Nozzle prototype as an autonomous agent. We will explore how it senses its environment, how its electronic nervous system connects brain to body, how it thinks and decides, how it acts, and how it communicates its status to a human operator.

NotebookLM

SENSES: 360° Perception of the Environment

Four ultrasonic sensors (HC-SR04) provide a complete, real-time picture of potential threats.



Key Specifications

Coverage: 360° via Front, Rear, Left, and Right sensors.

Range: 2cm to 400cm with ±2cm accuracy.

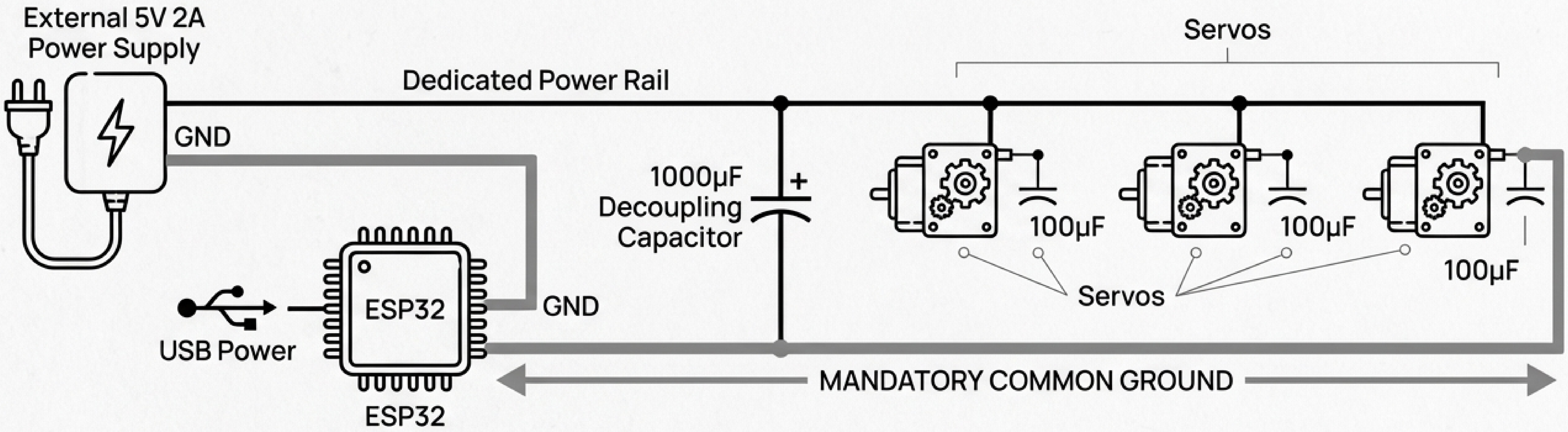
Refresh Rate: Minimum 10Hz (data updated every 100ms).

Data Filtering: A 5-sample moving average filter smooths noisy data. Invalid readings (<2cm or >400cm) are automatically rejected to ensure data integrity.

NotebookLM

THE NERVOUS SYSTEM: A Robust Electronics Core

A disciplined power and wiring design ensures reliability and prevents catastrophic failure.



Separate Power

Servos are powered by a dedicated 5V 2A supply to handle peak currents (up to 600mA per servo), preventing ESP32 brownouts.

Common Ground

The ESP32's ground and the servo power supply's ground are tied together. This is non-negotiable for stable servo control.

Decoupling

A 1000µF capacitor across the servo power rail absorbs current surges from servo movements, protecting the entire system from voltage drops.

NotebookLM

THE BRAIN: Real-Time Vector Computation

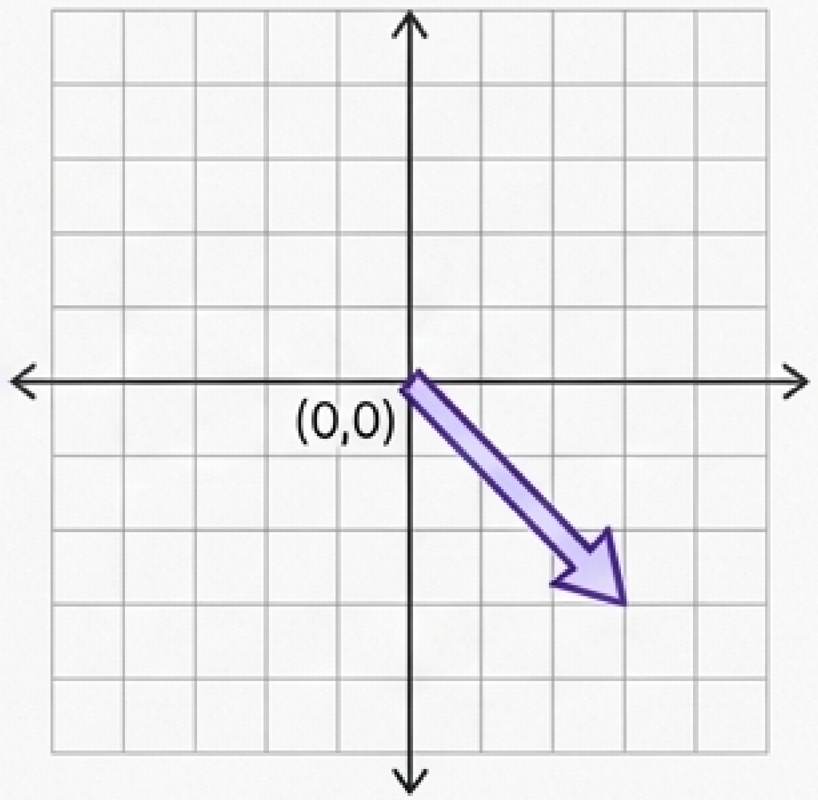
The system computes an avoidance vector using an inverse distance weighting algorithm, intelligently prioritizing the closest threats.

- [Front] Sensor Distances: 20cm
- [Left] Sensor Distances: 300cm
- [Right] Sensor Distances: 300cm
- [Rear] Sensor Distances: 300cm

Inverse Distance Weighting Logic

```
// For each sensor, weight is higher for closer objects
weight = (threshold - distance) / threshold;

// Combine weights to form a final vector
vector.x = weight[LEFT] - weight[RIGHT];
vector.y = weight[REAR] - weight[FRONT];
```



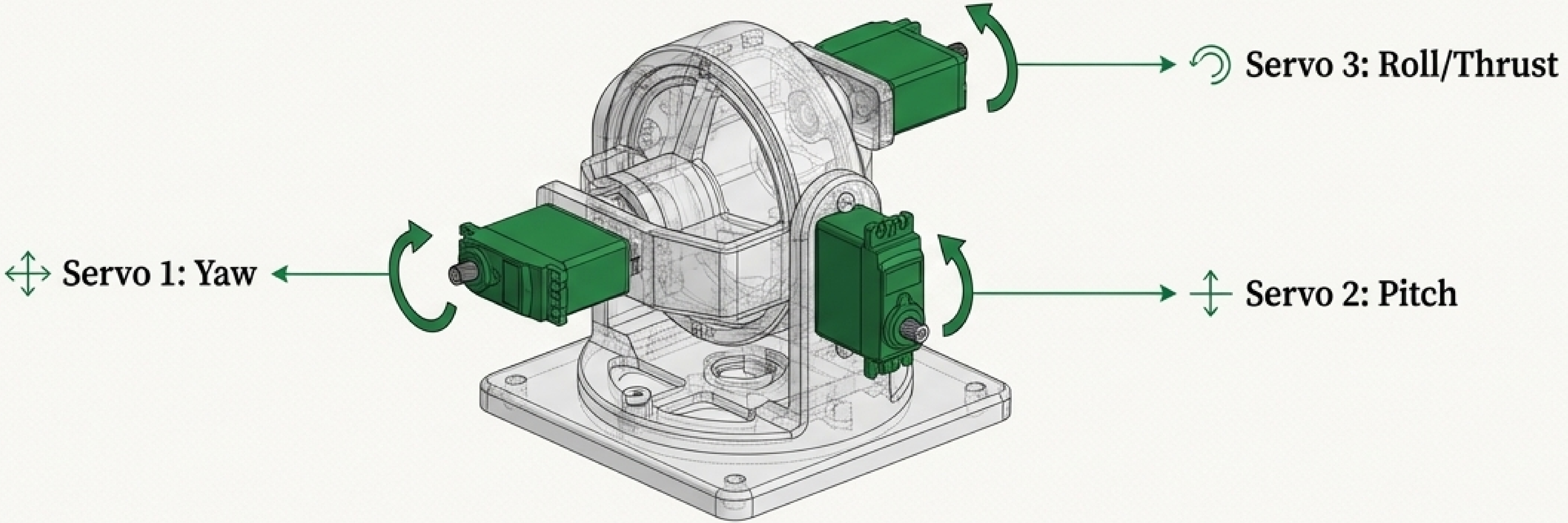
Scenario Detected	Distances (cm)	Computed Vector	Resulting Servo Action
Front Obstacle	F=20, L/R/B=300	(0, -1)	Pitch down, thrust backwards
Left Obstacle	L=15, F/R/B=300	(1, 0)	Yaw right, thrust away
Front-Left Corner	F=25, L=25, R/B=300	(0.707, -0.707)	Diagonal back-right thrust

Response time from detection to servo command is under 150ms.

NotebookLM

THE BODY: The 3-Axis Swizzle Nozzle

Three servo motors provide precise, continuous thrust vectoring, mimicking modern rocket gimbal systems.



Servo 1: YAW
Controls left-right nozzle tilt. Directs thrust laterally.

Servo 2: PITCH
Controls up-down nozzle tilt. Directs thrust vertically.

Servo 3: ROLL / THRUST
Controls thrust intensity (e.g., via a shutter) or provides roll stabilization. Fine-tunes maneuver magnitude.

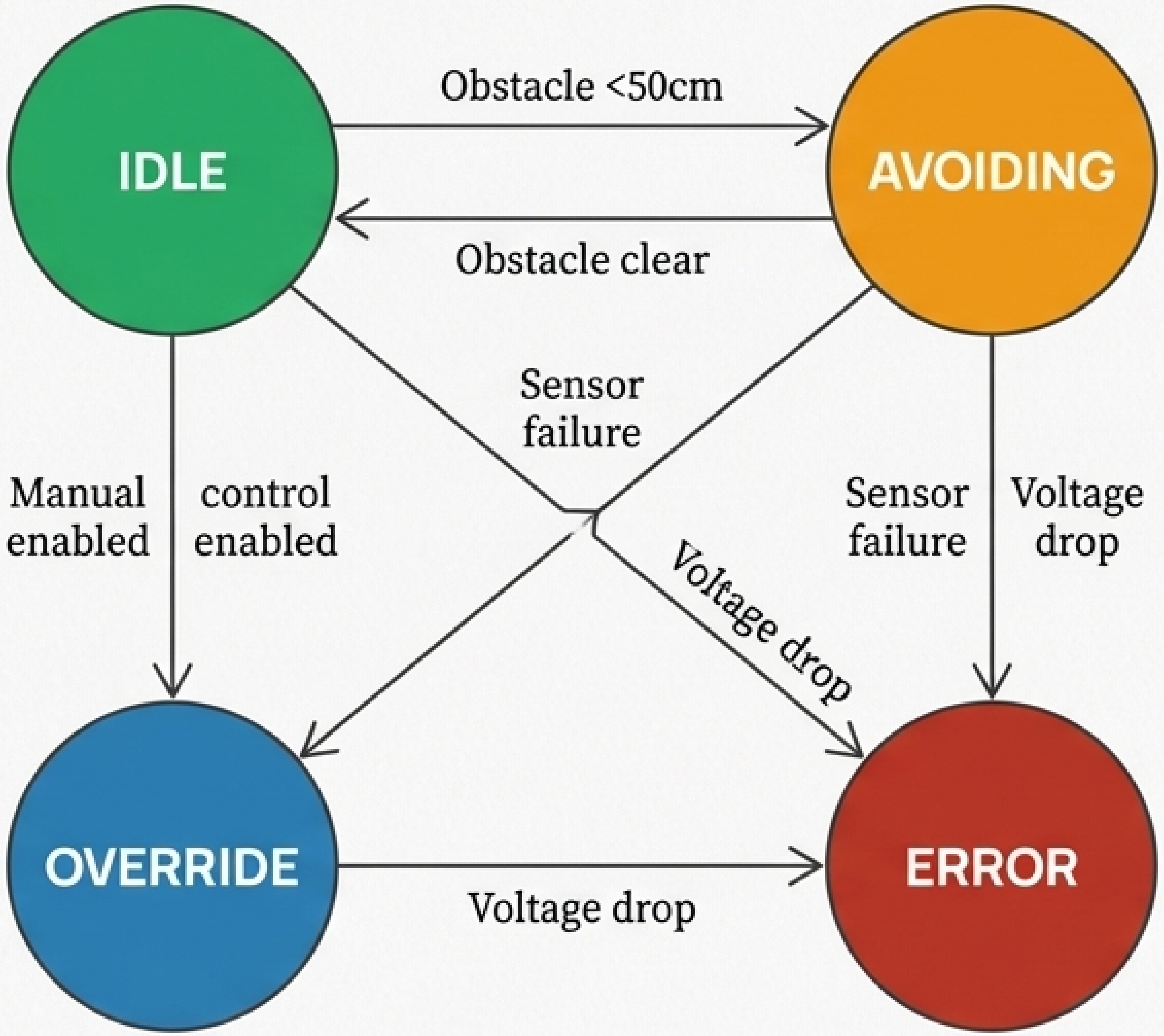
Key Mechanical Specifications

- **Range of Motion**: Software-constrained to 60°–120° (±30° from neutral) to prevent binding.
- **Smooth Ramping:** Servos move gradually (1° per 20ms) to eliminate mechanical shock and jitter.
- **Construction**: Lightweight 3D-printed PLA/ABS gimbal on a stable 150mm x 150mm base.

NotebookLM

Governing Behavior: The System State Machine

The firmware's state machine ensures predictable and safe operation under all conditions.



Non-Blocking by Design

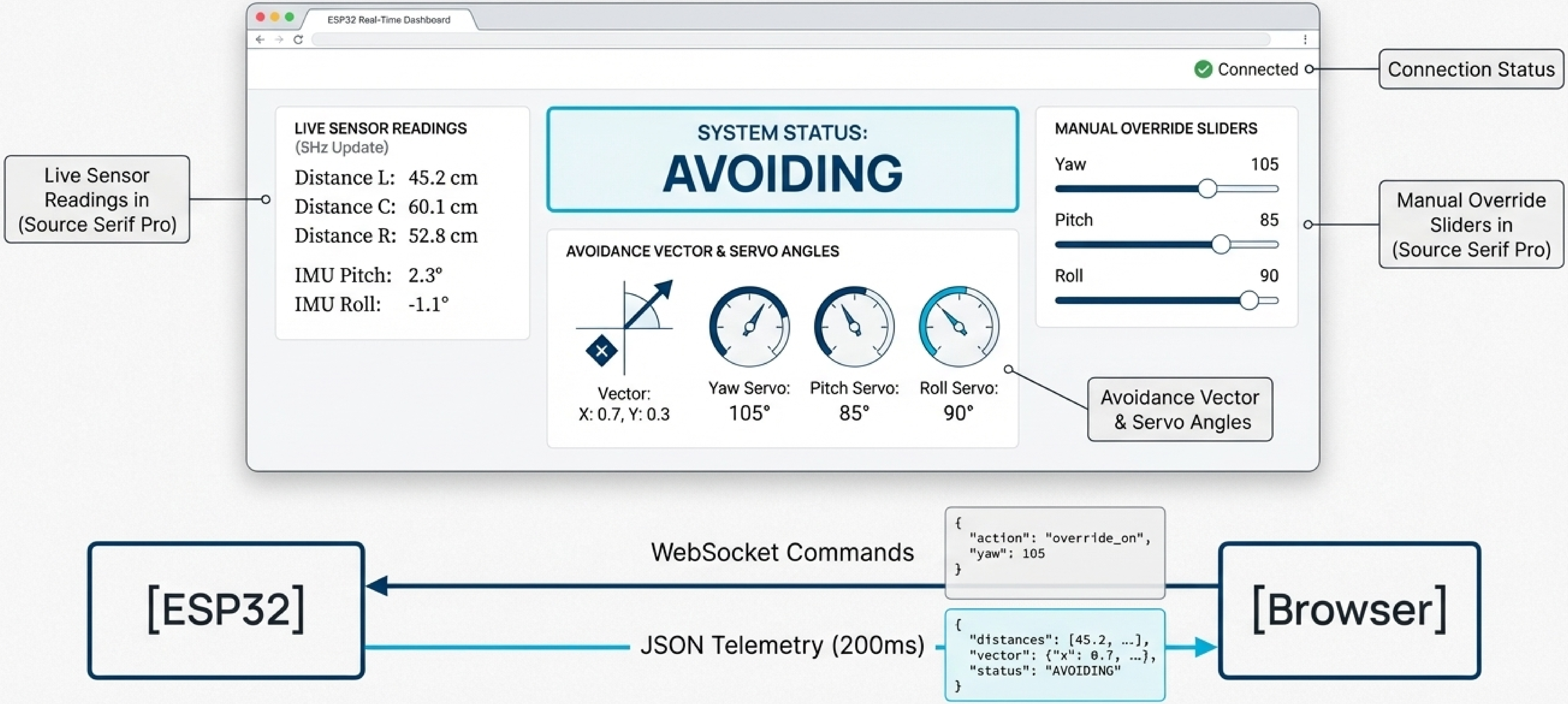
The system uses millis()`-based timing instead of `delay()` for all operations. This allows the sensors, servos, and web server to run in parallel without ever freezing the processor, ensuring consistent timing and responsiveness.

```
void loop() {
  // Read sensors every 100ms
  if (currentTime - lastSensorRead >= SENSOR_INTERVAL) {
    readAllSensors();
    // ...run control logic...
  }
  // Other tasks run continuously
  updateServos();
  handleWebRequests();
}
```

NotebookLM

THE INTERFACE: A Real-Time Web Dashboard

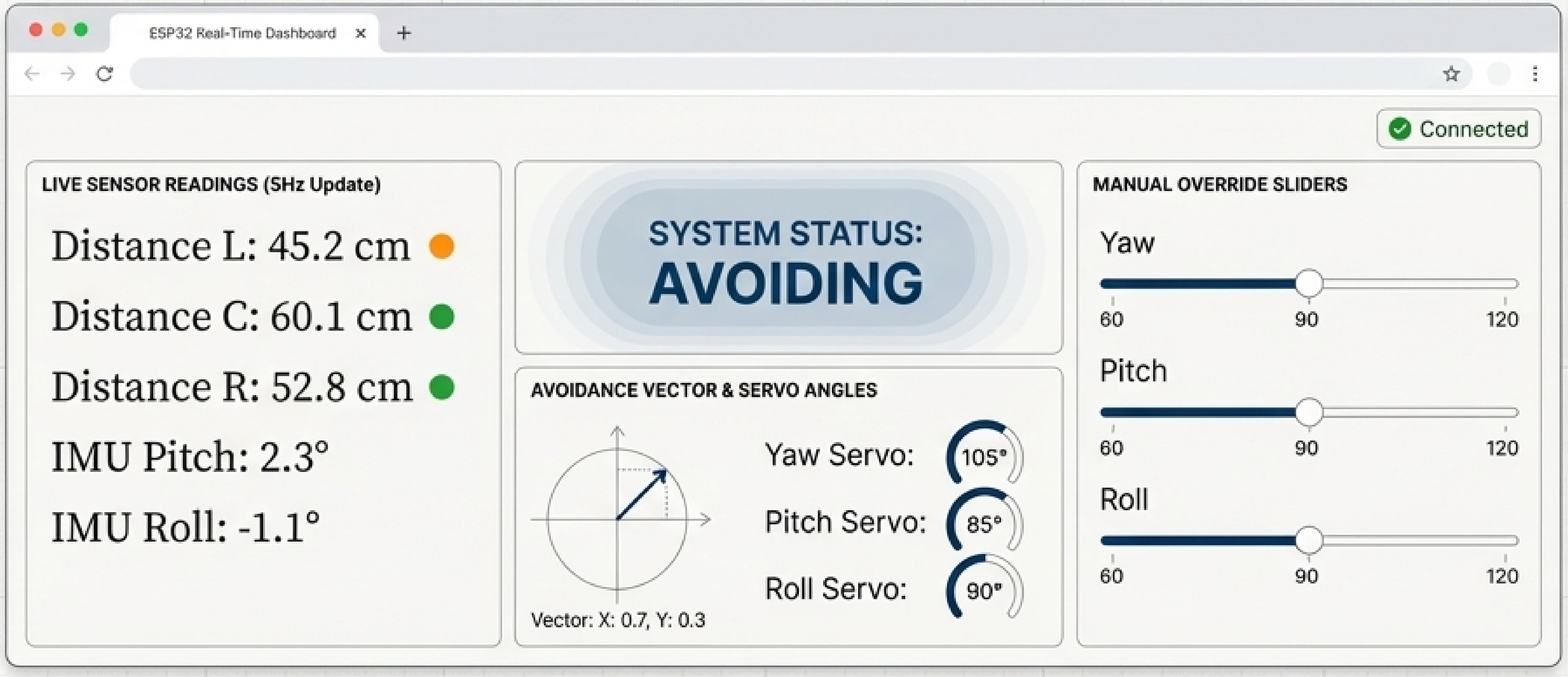
A single-page web application, hosted directly on the ESP32, provides live telemetry and manual control over WiFi.



NotebookLM

Designed for Control and Clarity

The interface is built to be intuitive and responsive, providing a seamless experience from desktop to mobile.



Visual Alerts: Sensor values are color-coded: green (safe), orange (caution), and red (<50cm, danger).

Intuitive Sliders: Manual controls use standard, touch-friendly sliders for direct manipulation of servo angles from 60° to 120°.

Live Status Badge: The status indicator (e.g., "AVOIDING") pulses when in an active state to draw user attention.

Resilient Connection: The interface includes a clear "Connected/Disconnected" status banner and logic for automatic reconnection.

NotebookLM

From Theory to Reality: Rigorous System Validation

The prototype passed a comprehensive suite of tests confirming its performance, reliability, and safety.

Hardware & Power Integrity

- ✅ Stable 5.0V ±0.2V supply under full servo load.
- ✅ No resets or brownouts during a 1-hour continuous stress test.

Control Logic & Accuracy

- ✅ Correct avoidance vector computed for all cardinal and diagonal obstacle scenarios.
- ✅ Sensor accuracy maintained at ±2cm; servo positioning confirmed at ±2°.

System Performance

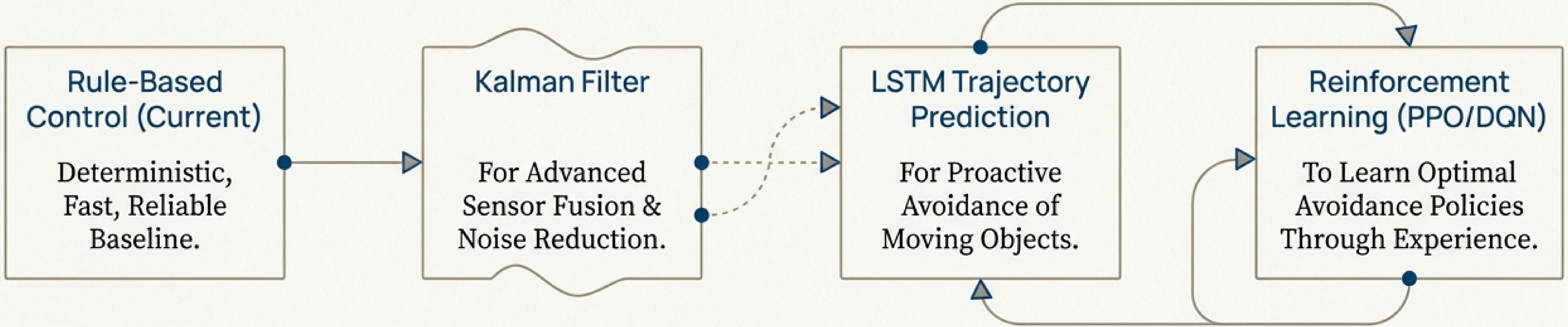
- ✅ Sensor-to-servo control latency is under 150ms.
- ✅ Web dashboard update latency is under 300ms, even with multiple clients connected.

The system is validated as fully operational, meeting all specified functional and performance requirements.

NotebookLM

THE EVOLVING MIND: An Architecture Built for AI

The current rule-based system is just the beginning. The modular architecture is designed to integrate advanced AI decision-making models.



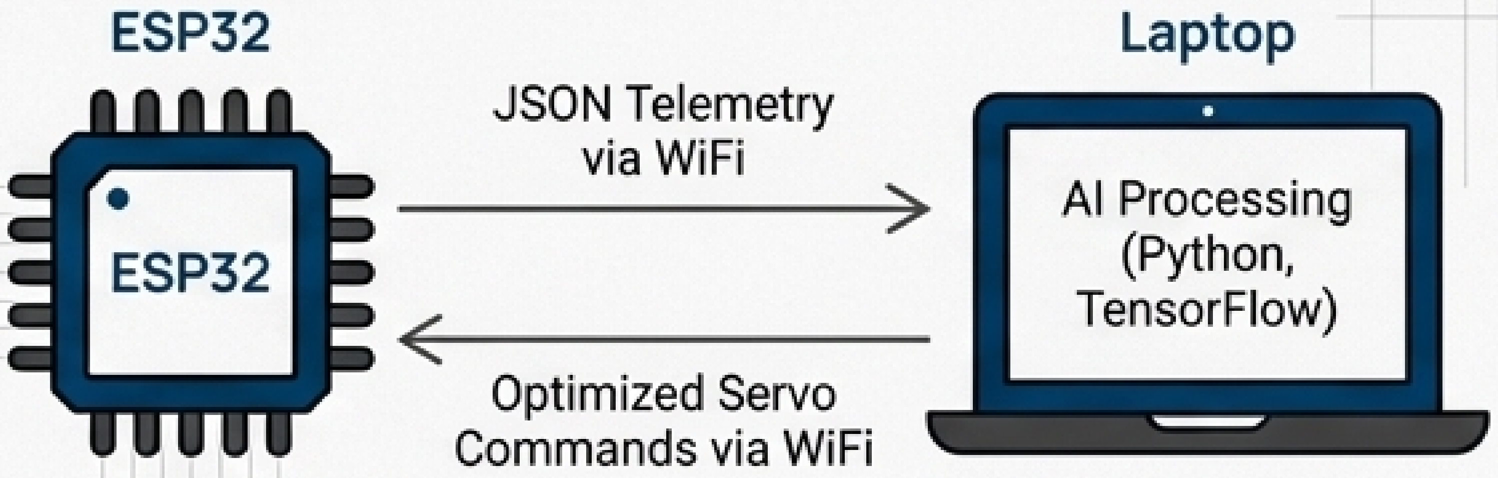
The system's true potential lies in its ability to learn. The data pipeline and control interfaces are ready for a series of upgrades that will transform the system from a reactive machine into a predictive, intelligent agent.

NotebookLM

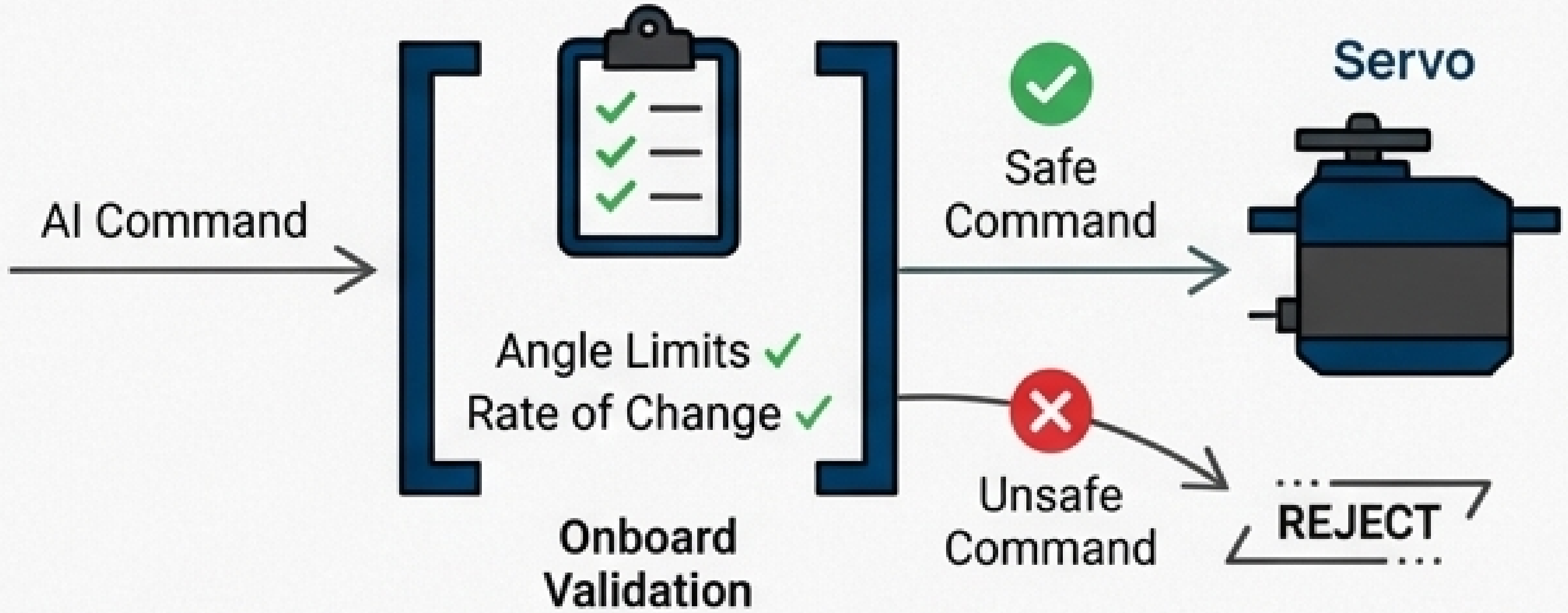
The Path to Intelligence

Integrating models for prediction and adaptive control via a flexible data pipeline.

The AI Data Pipeline



Safety First - The AI Command Validator



Onboard AI Command Validation

All AI-generated commands are validated on the ESP32 before execution to ensure they are within safe operating limits (angle constraints, max rate of change). If a command is invalid or the connection is lost, the system instantly reverts to its reliable onboard rule-based control.

```
bool validateAICommand(ServoAngles cmd) {
  // Check angle limits (60° to 120°)
  if (cmd.yaw < SERVO_MIN || cmd.yaw > SERVO_MAX) return false;
  // Check rate of change against max delta
  if (abs(cmd.yaw - currentAngles.yaw) > MAX_DELTA_PER_STEP) return false;
  return true; // Command is safe to execute
}
```

NotebookLM

Build Your Own: Specifications & Bill of Materials

An open-source project designed with accessible, off-the-shelf components.

Key Specifications

Platform: ESP32 DevKit (WiFi Capable)

Sensors: 4x HC-SR04 Ultrasonic

Actuators: 3x MG90S Micro Servos

Control Latency: <150ms

Dashboard Latency: <300ms

Power: External 5V 2A supply (peak draw ~1.1A)

Software Stack: Arduino IDE, ESPAsyncWebServer, WebSocket, ArduinoJson

Bill of Materials

Component	Qty	Estimated Cost (USD)
ESP32 DevKit	1	$8-12
HC-SR04 Sensors	4	~$6
MG90S Micro Servos	3	~$12
5V 2A Power Supply	1	~$7
Capacitors & Wires	Pack	~$5
Total Estimated Cost		**~$40-50**

Full wiring diagrams, 3D models, and source code are available in the project documentation.

NotebookLM

The Vision of the Architect

ExoNova
AI project

This system was designed, built, and programmed by **Sanchit ('Dark')**, a 16-year-old 11th-grade student and space technology enthusiast. As part of his ExoNova AI project, he combines hands-on experience in Arduino, 3D modeling, and machine learning to explore the complex challenges of autonomous spacecraft navigation.

"Building the future of space exploration, one prototype at a time."

The AI-Powered Swizzle Nozzle is a testament to the power of focused passion and a glimpse into the future of accessible, intelligent systems.

NotebookLM