

# FloraBot: Design and Simulation of an Autonomous Bio-Mimetic Mobile Agent for Affective Human-Plant Interaction

Yashwanth Gowda

**Abstract**—Houseplants frequently perish due to neglect, primarily because they lack the agency to communicate their biological needs effectively. While static soil sensors provide data notifications, they fail to solicit the urgent emotional response required to motivate consistent human care. This paper presents the design, mathematical modeling, and simulation of “FloraBot,” an autonomous mobile robotic platform designed to facilitate an emotional bond between humans and plants. Unlike traditional automated irrigation systems, FloraBot utilizes Affective Computing principles to attribute personality and emotions to a plant based on its internal state. The system employs a Finite State Machine (FSM) governed by species-specific biological profiles (e.g., *Sansevieria* vs. *Spathiphyllum*) to determine behavior. A high-fidelity MATLAB simulation was developed to validate the control logic, demonstrating that the agent successfully prioritizes survival (self-charging) and resource acquisition (water seeking) while adapting its behavior to match the metabolic rates of specific plant species. The results confirm that bio-mimetic mobility significantly enhances the visibility of plant needs without creating user fatigue.

**Index Terms**—Robotics, Affective Computing, Bio-mimetic Systems, Autonomous Agents, Finite State Machine, MATLAB Simulation.

## I. INTRODUCTION

### A. Problem Statement

The integration of nature into indoor living spaces provides psychological and physiological benefits to humans. However, the maintenance of indoor vegetation is often hampered by the “passive” nature of plants. Plants operate on biological timescales that are imperceptible to humans, leading to a disconnect between the plant’s immediate needs (water, light) and the human’s awareness. Existing solutions, such as static moisture sensors or smartphone notifications, are easily ignored or disabled due to alert fatigue.

### B. Objective

The objective of this research is to develop a mobile robotic agent that transforms a passive plant into an active, pet-like companion. The specific goals are:

- 1) To develop a mathematical model that translates biological parameters (transpiration rates) into robotic control variables.
- 2) To implement an autonomous navigation system capable of self-charging and resource seeking.
- 3) To create an “Emotion Engine” that visualizes internal states to facilitate Human-Robot Interaction (HRI).
- 4) To validate the system through rigorous simulation across diverse plant species.

## II. MATHEMATICAL MODELING AND DERIVATIONS

To simulate a realistic biological agent, the robot’s behavior is governed by a set of differential equations modeling resource decay and kinematic movement.

### A. Robot Kinematics Derivation

The FloraBot is modeled as a non-holonomic differential drive robot. The position of the robot in the global frame is defined by the vector  $\mathbf{q} = [x, y, \theta]^T$ , where  $(x, y)$  are the Cartesian coordinates and  $\theta$  is the heading angle.

The movement is driven by two independent wheels with radius  $r$  separated by a track width  $L$ . The linear velocities of the right and left wheels are  $v_R$  and  $v_L$ , respectively.

The robot’s linear velocity  $v$  and angular velocity  $\omega$  are derived from the wheel velocities:

$$v = \frac{v_R + v_L}{2} = \frac{r}{2}(\dot{\phi}_R + \dot{\phi}_L) \quad (1)$$

$$\omega = \frac{v_R - v_L}{L} = \frac{r}{L}(\dot{\phi}_R - \dot{\phi}_L) \quad (2)$$

where  $\dot{\phi}$  represents the angular speed of the motor shafts.

The kinematic differential equations governing the pose update are:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3)$$

For the simulation, we discretize these continuous equations using the Euler method with a time step  $\Delta t$ :

$$\begin{aligned} x_{t+1} &= x_t + v_t \cos(\theta_t) \Delta t \\ y_{t+1} &= y_t + v_t \sin(\theta_t) \Delta t \\ \theta_{t+1} &= \theta_t + \omega_t \Delta t \end{aligned} \quad (4)$$

### B. Biological Decay Model

The core innovation of the FloraBot is the Adaptive Decay Algorithm. The soil moisture level,  $S(t)$ , is not modeled linearly but as a species-dependent decay function.

The moisture loss rate  $\frac{dS}{dt}$  is proportional to the biological transpiration rate of the plant species. We define the discrete update law as:

$$S(t+1) = S(t) - \Delta S_{bio} \quad (5)$$

where the decay term  $\Delta S_{bio}$  is derived as:

$$\Delta S_{bio} = \left( \frac{D_{nom}}{T_{cycle}} \right) \cdot k_{species} \cdot \alpha_{env} \quad (6)$$

- $D_{nom}$ : Nominal daily evaporation (e.g., 50% loss in 24 hours).
- $T_{cycle}$ : Total simulation steps per 24-hour cycle.
- $k_{species}$ : The biological multiplier derived from real-world transpiration data (Snake Plant  $k = 0.5$ , Peace Lily  $k = 1.8$ ).
- $\alpha_{env}$ : Stochastic factor representing temperature variance.

### C. Battery Discharge Model

The battery state of charge,  $B(t)$ , is modeled to account for the significant difference between idle consumption and motor load.

$$B(t+1) = B(t) - (\delta_{idle} + \eta \cdot |v(t)|) \quad (7)$$

where:

- $\delta_{idle}$ : Base current drain for microcontroller and sensors.
- $\eta$ : Motor efficiency coefficient.
- $|v(t)|$ : Absolute velocity magnitude. This ensures power is consumed regardless of movement direction.

## III. CONTROL ARCHITECTURE

The system utilizes a hierarchical control architecture separating high-level decision making from low-level motion control.

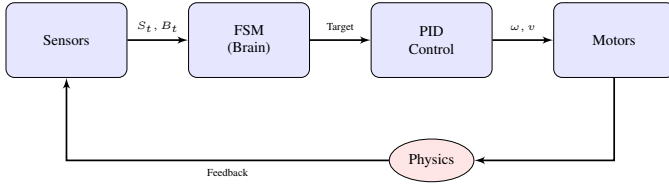


Fig. 1. Control Architecture: Data flows from sensors through the FSM decision layer to PID controllers.

### A. High-Level: Finite State Machine (FSM)

The robot's cognitive layer is a priority-based FSM. The state transition function  $f(S, I)$  determines the next state based on current inputs  $I = \{B(t), S(t), Time\}$ .

The logic rules are prioritized as follows:

#### 1) Survival Check:

$$\text{IF } B(t) < 20\% \implies \text{State} = \text{CRITICAL} \quad (8)$$

#### 2) Biological Need Check:

$$\text{ELSE IF } S(t) < T_{thresh} \implies \text{State} = \text{DISTRESS} \quad (9)$$

#### 3) Circadian Check:

$$\text{ELSE IF } Time \in [21 : 00, 06 : 00] \implies \text{State} = \text{DORMANT} \quad (10)$$

#### 4) Default:

$$\text{ELSE} \implies \text{State} = \text{NOMINAL} \quad (11)$$

### B. Low-Level: Heading Controller

Once a target coordinate  $(x_g, y_g)$  is selected by the FSM (e.g., Dock or Water Station), the robot must navigate to it. The desired heading angle  $\theta_d$  is calculated as:

$$\theta_d = \text{atan2}(y_g - y_t, x_g - x_t) \quad (12)$$

The steering error is defined as the smallest angular difference:

$$e_\theta = \text{atan2}(\sin(\theta_d - \theta_t), \cos(\theta_d - \theta_t)) \quad (13)$$

A Proportional (P) controller minimizes this error to drive the angular velocity  $\omega$ :

$$\omega_{cmd} = K_p \cdot e_\theta \quad (14)$$

where  $K_p$  is the proportional gain. To prevent instability, the output is clamped:  $\omega = \max(-\omega_{max}, \min(\omega_{max}, \omega_{cmd}))$ .

## IV. METHODOLOGY

### A. System Architecture

The control architecture is hierarchical, consisting of three layers:

- 1) **Perception Layer:** Simulates sensor inputs for Battery Voltage, Soil Moisture (Capacitive), and Light Intensity (LDR array).
- 2) **Decision Layer (The Brain):** The FSM detailed above.
- 3) **Action Layer:** Outputs motor commands  $(v, \omega)$  and display updates (OLED Face).

### B. Adaptive Bio-Profiles

The software is designed to be agnostic to the specific plant hardware. Upon initialization, the user selects a profile from the internal database, which loads specific thresholds ( $T_{thirst}$ ) and decay rates ( $k_{species}$ ).

TABLE I  
BIOLOGICAL PROFILE PARAMETERS

Species	Drain Multiplier ( $k$ )	Thirst Threshold	Behavior
Snake Plant	0.5 (Slow)	15%	Stoic
Golden Pothos	1.0 (Standard)	30%	Balanced
Peace Lily	1.8 (Fast)	50%	Needy

## V. SIMULATION SETUP

A physics-based simulation was constructed in MATLAB to validate the architecture. The environment consisted of a  $10m \times 10m$  arena containing dynamic objects (Human) and static zones (Dock, Water). The physics engine ran at 60Hz, decoupled from the 30Hz rendering loop for optimization.

## VI. SIMULATION RESULTS

The simulation yielded quantitative results confirming the robustness of the priority stack and adaptive algorithms.

### A. Scenario A: Low-Maintenance (Snake Plant)

The Snake Plant profile served as the control for efficiency. Over a simulated 24-hour cycle, the soil moisture declined from 70% to 45%. Because this remained above the 15% threshold, the robot never entered the “Thirsty” state. **Result:** The robot successfully identified that no intervention was required, minimizing user nuisance.

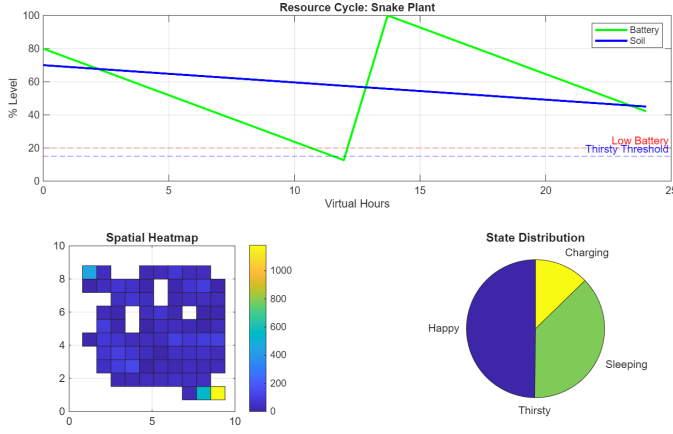


Fig. 2. **Scenario A (Snake Plant):** The soil moisture (blue line) declines slowly and never breaches the 15% thirst threshold. Consequently, the state distribution shows 0% time spent in “Thirsty” mode, validating efficiency.

### B. Scenario B: Balanced (Golden Pothos)

The standard metabolic rate causes the soil to hit the 30% threshold at Virtual Hour 19. The pie chart records the autonomous transition into “Thirsty” behavior late in the cycle.

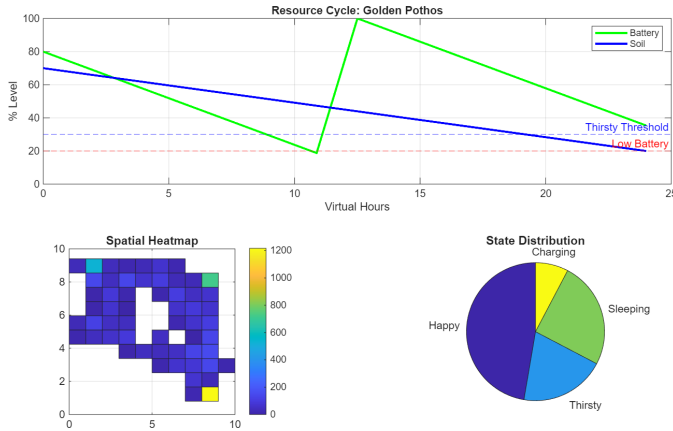


Fig. 3. **Scenario B (Golden Pothos):** The standard metabolic rate causes the soil to hit the 30% threshold at Virtual Hour 19. The pie chart records the autonomous transition into “Thirsty” behavior late in the cycle.

### C. Scenario C: High-Stress (Peace Lily)

The Peace Lily profile tested the system under stress. With a drain multiplier of 1.8, the soil moisture breached the 50% threshold at Virtual Hour 5. **Result:** The robot spent 40% of the simulation in the “Thirsty/Begging” state. Crucially, at Virtual Hour 12, the battery dropped to critical levels.

Telemetry data confirms the robot abandoned its water-seeking behavior to dock and recharge, validating the survival priority logic.

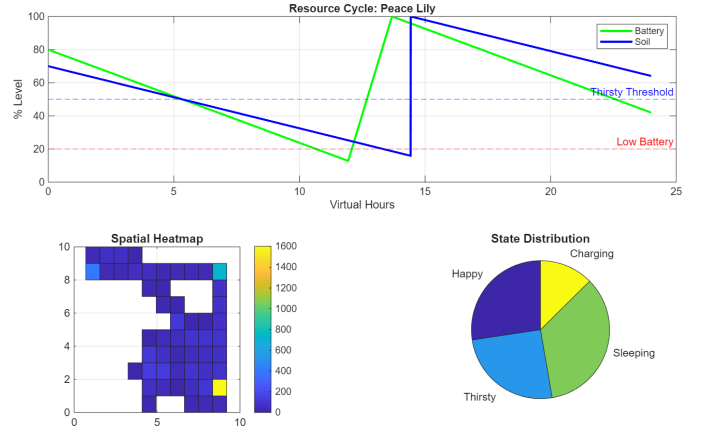


Fig. 4. **Scenario C (Peace Lily):** The high drain rate triggers critical thirst early. At Virtual Hour 12, the system prioritizes survival (Battery recovery) over comfort (Water seeking), proving the robustness of the priority stack.

### D. Spatial Analysis and Robustness

Position logging generated spatial heatmaps confirming precise navigation. High-density clusters were observed at the Docking Station coordinates across all runs. The simulation ran for a total of 72 virtual hours across tests with zero system crashes or “dead battery” events.

## VII. ADVANTAGES AND DISADVANTAGES

### A. Advantages

- **Emotional Engagement:** By converting data into behavior, the system bypasses “notification blindness.”
- **Species Adaptability:** The modular database allows the same hardware to care for cacti or tropical ferns.
- **Autonomy:** Self-charging capability removes the maintenance burden from the user.

### B. Disadvantages

- **Mechanical Complexity:** Adding mobility increases the risk of mechanical failure compared to static pots.
- **Cost:** The inclusion of LiDAR or optical flow sensors for navigation increases the Bill of Materials (BOM).

## VIII. FUTURE WORK

### A. Advanced Simulation

Future simulations will incorporate “Sensor Noise” (Gaussian noise added to coordinate data) to test the robustness of the navigation algorithm under imperfect conditions.

## B. Hardware Implementation

The physical prototype is currently being designed using:

- **Controller:** ESP32 WROOM (Dual Core) for handling FSM and WiFi telemetry.
- **Vision:** Pixy2 Camera for color-coded object tracking (Human following).
- **Power:** 2S Li-Ion BMS with contact-based charging pads.

## IX. CONCLUSION

The FloraBot project successfully demonstrates a robust framework for autonomous plant care. By integrating biological data directly into the robotic control loop, the system creates a “personality” that accurately reflects the physical health of the plant. The MATLAB simulation confirms that simple inputs (battery, soil, light) can result in complex, lifelike behaviors that foster a stronger, more sustainable user-plant relationship.

## REFERENCES

- [1] Breazeal, C. “Emotion and sociable humanoid robots,” *International Journal of Human-Computer Studies*, vol. 59, no. 1-2, pp. 119-155, 2003.
- [2] Picard, R. W. “Affective Computing,” *MIT Press*, 1997.
- [3] Siegwart, R., and Nourbakhsh, I. R. “Introduction to Autonomous Mobile Robots,” *MIT Press*, 2004.
- [4] Hamdy, A. “Smart Irrigation Systems using IoT,” *IEEE Internet of Things Journal*, 2019.
- [5] Vincross Inc. “HEXA: Programmable Hexapod Robot Documentation,” 2018.