

Dynamics Control and Instrumentation

Monkey Deterrent System

GOEE4002

Kieran Prescott

2026

Table of Contents

1. Introduction.....	p3
2. Supporting Theory.....	p3
3. System Overview.....	p3
3.1. Motion Detection.....	p4
3.2. Light Detection.....	p4
3.3. Alert Activation.....	p4
4. Modelling for System Behaviour.....	p5
4.1. Sound Output Model.....	p5
4.2. Light Output Model.....	p5
5. Simulink Model and Simulation.....	p6
5.1. Inputs.....	p6
5.2. Logic.....	p7
5.3. Arming Timer.....	p9
5.4. Outputs.....	p10
5.5. Simulation Time.....	p11
5.6. Embedded Implementation Arduino Closed-Loop System.....	p11
6. Simulation Results.....	p12
6.1. Overview.....	p12
6.2. Day-Time Scenario.....	p12
6.3. Night-Time Scenario.....	p13
7. Design Justification.....	p14
8. Modelling Assumptions.....	p14
9. System Limitation.....	p15
10. Critical Evaluation of System Performance.....	p15
11. Components Required.....	p15
12. Solar Powered Design.....	p16
13. Cost Analysis.....	p17
14. Maintainability.....	p17
15. Conclusion.....	p17
16. References.....	p18
Appendix A – Arduino Source Code.....	p19
Appendix B – Arduino Simulation Video.....	p23

1. Introduction

The Chomzangari Women's Cooperative in Zimbabwe relies heavily on small scale agriculture, but crop losses caused by monkeys entering the fields have become a major issue. Traditional Scarecrows are no longer effective because monkeys quickly learn that they pose no real threat. Human guarding is also unsustainable due to labour demands.

This project presents a low cost, solar powered monkey deterrent system that uses motion and light sensing to trigger sound and light alerts. The design applies principles of dynamic systems, instrumentation, and control, and uses MATLAB/Simulink to simulate system behaviour. The aim is to create a practical, maintainable solution suitable for remote, off grid agricultural environments.

2. Supporting Theory

Effective wildlife deterrent systems rely on understanding animal behaviour, sensing technologies and control strategies. Research shows that multi-sensory, non-harmful deterrents can reduce primate crop damage, especially when stimuli are unpredictable (O'Brien & Hill, 2018). Electronic and remotely controlled deterrent systems have also been shown to improve reliability and reduce labour requirements (Ojara, 2023). IoT-based deterrent systems in plantations demonstrate the value of combining sensors with automated alerts to discourage monkeys (Rahim & Zainal, 2024)

From an engineering perspective, dynamic systems are often modelled using differential equations to predict systems response. Sensors such as PIR detectors and LDR's provide environmental feedback, while simple logic-based control is suitable for low power, low-cost applications. MATLAB and Simulink offer a flexible environment for modelling, simulation and analysis of such systems.

3. System Overview

The proposed monkey deterrent system uses a layered sensing and response structure to detect monkey activity and activate multiple deterrent outputs. The design is intentionally simple, low power, and suitable for solar operation in remote agricultural environments. The system combines mechanical movement, sound output, and light output to create a multisensory deterrent that is harder for monkeys to habituate to.

A physical prototype of the system was also constructed using an Arduino microcontroller (Arduino, 2024), PIR sensor, LDR, motor, buzzer, and LEDs. In this build, the Arduino acts as a closed-loop controller, continuously monitoring sensor inputs and updating the outputs according to the logic described in this section. This confirms that the proposed design can be realised as a low-cost, standalone deterrent suitable for remote agricultural environments.

The system consists of three main function layers:

1. Sensing layer – Detects Motion and Ambient Light
2. Logic Layer – Determines day/night behaviour
3. Actuation Layer – Drives the motor, buzzer and light outputs
 - Motor Spins a light weight arm fitted with recyclable tins or metal strips to create unpredictable movement and mechanical noise.

3.1. Motion Detection

Motion is detected using a passive infrared (PIR) sensor. PIR sensors detect changes in infrared radiation caused by movement of warm bodied animals. When motion is detected, the PIR outputs a digital high signal (Logic 1). In Simulink, this is represented using a step block that transitions from 0 to 1 at 5seconds.

Real world note:

PIR sensor can be mounted at specific height and downward angle, that is more sensitive to low height movement (monkeys) and less sensitive to tall humans walking upright. This reduces false activations when cooperative members move around the field.

3.2. Light Detection

Ambient light level is measured using a Light Dependent Resistor (LDR). The LDR output decreases as light level fall. A comparator check whether the LDR value is below 2.5, representing night time conditions.

$LDR < 2.5 = \text{Night} = 1$

$LDR \geq 2.5 = \text{Night} = 0$

3.3. Alert Activation

The system activates different deterrents depending on the time of day:

Day-Time (Night = 0)

1. Motor ON (Spinning arm creates movement/metallic noise)
2. Buzzer ON
3. Light OFF

Night-Time (Night = 1)

1. Motor ON (Spinning arm creates movement/metallic noise)
2. Buzzer ON
3. Light ON

A short arming delay is included so that when the system is first switched on, cooperative members have time to move away before the deterrent becomes active. This improves safety and prevents accidental triggering during setup.

A manual override (Kill switch) is also included on the enclosure to allow users to safely enter the field without activating the deterrent. When switched off, the override isolates the actuators from the control logic, ensuring the system remains inactive regardless of sensor input.

4. Modelling for System Behaviour

To simulate realistic actuator behaviour, each output is modelled as a first order dynamic system. Real motors, buzzers and lights do not switch instantly. They gradually ramp over time.

General first order system is described by:

$$\tau \frac{dy(t)}{dt} + y(t) = u(t)$$

$U(t)$ = Input Signal

$Y(t)$ = Actuator Response

T = Time constant

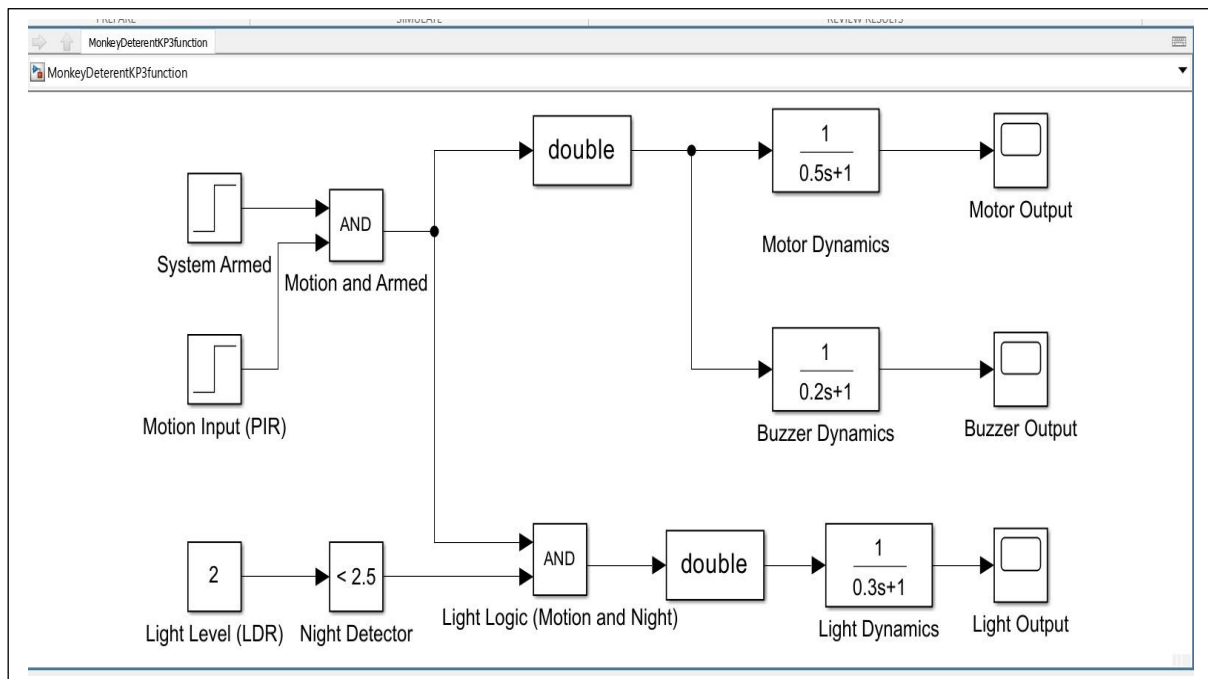
4.1. Sound Output Model Buzzer

$$Gb(s) = \frac{1}{0.2s + 1}$$

4.2. Light Output Model

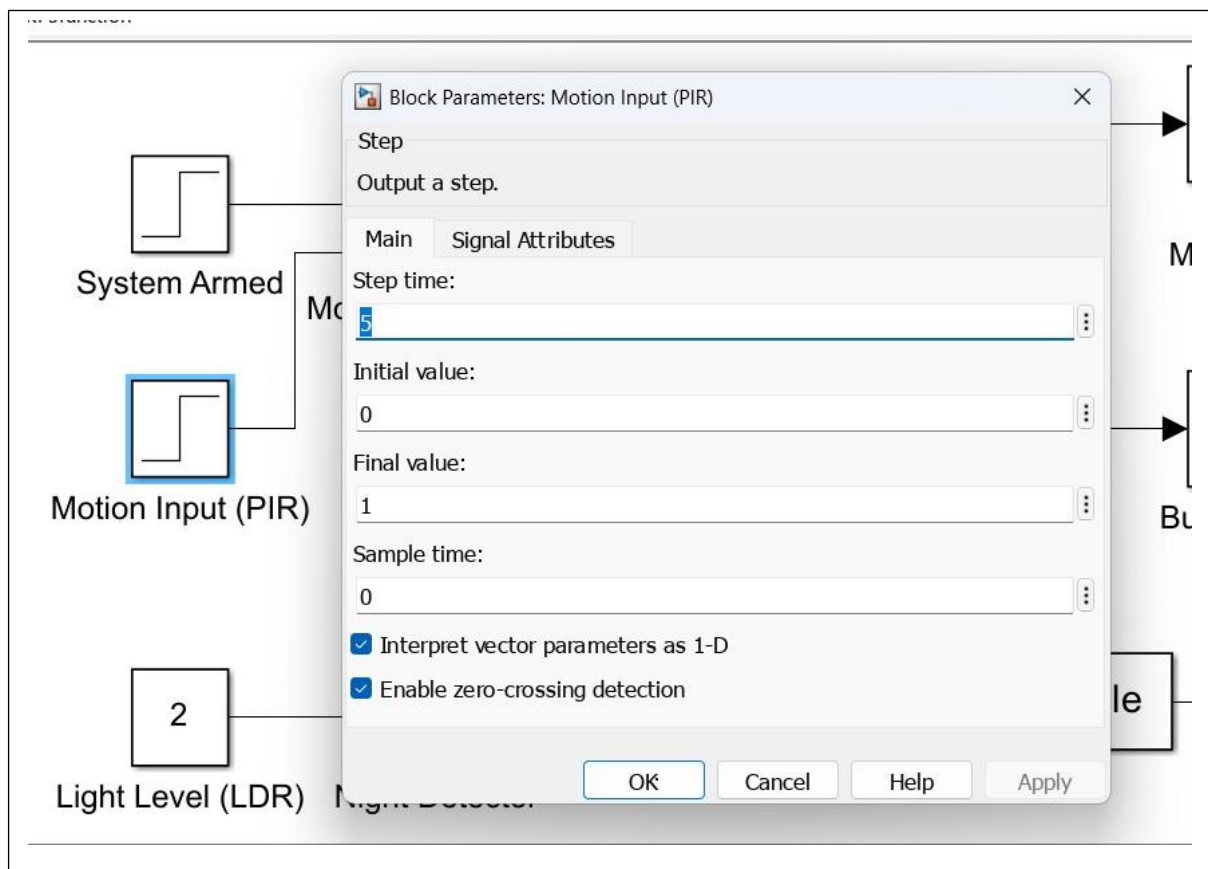
$$Gi(s) = \frac{1}{0.3s + 1}$$

5. Simulink Model and Simulation



5.1. Inputs

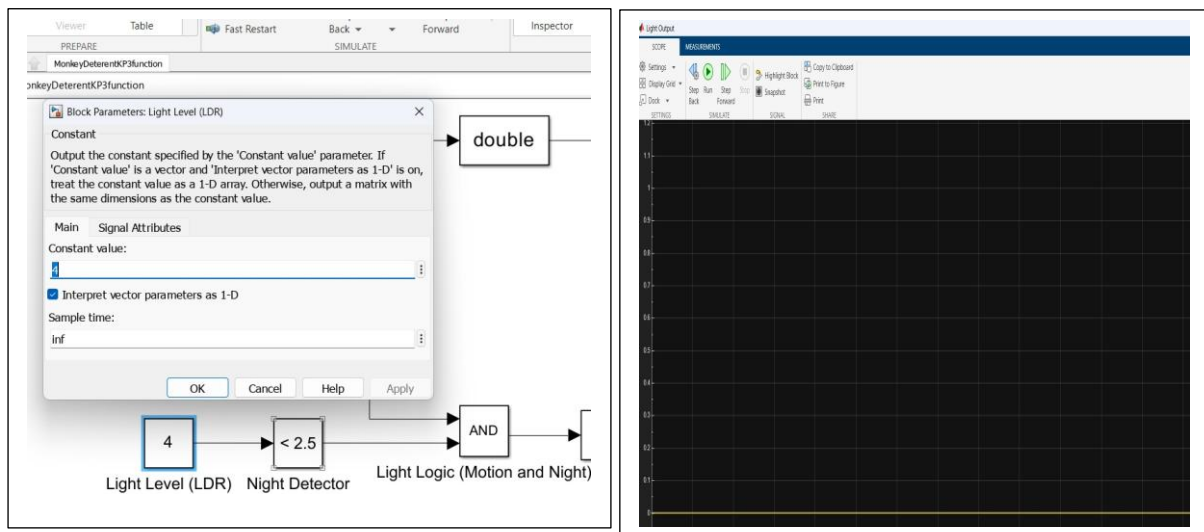
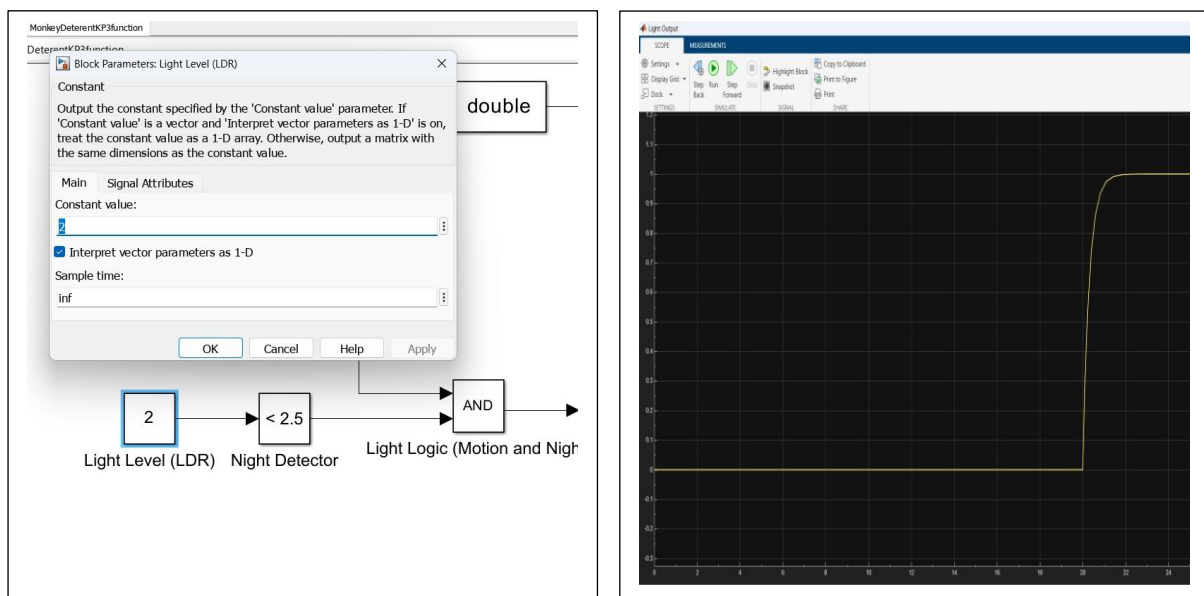
PIR Step Block Settings



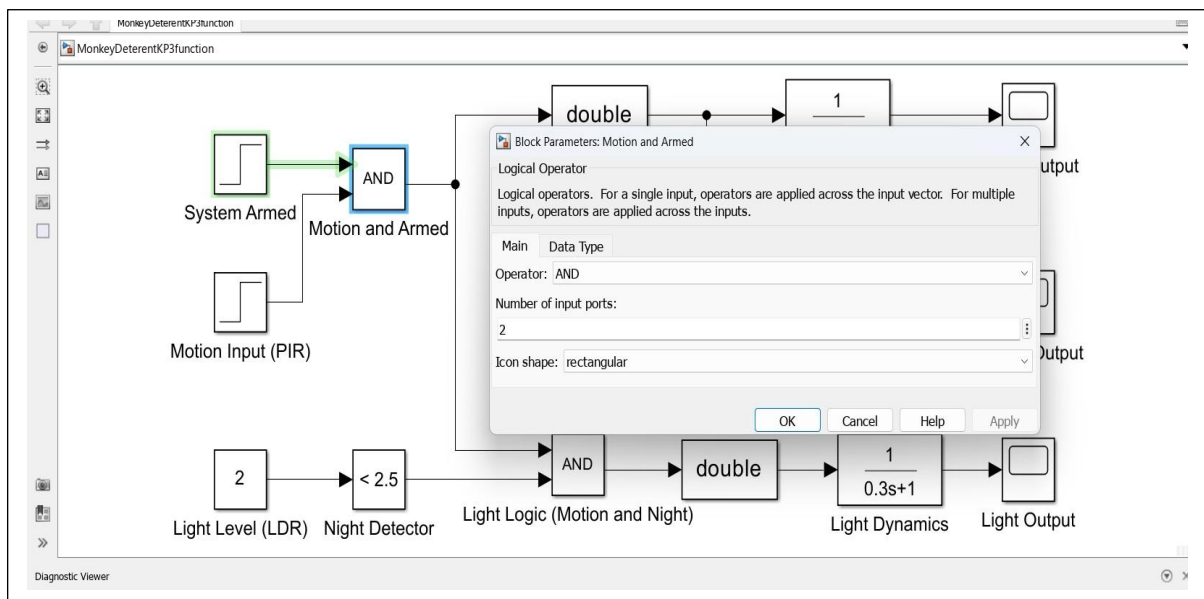
LDR Constant Block

LDR set at $4 \geq 2.5$ No Light Output Activation

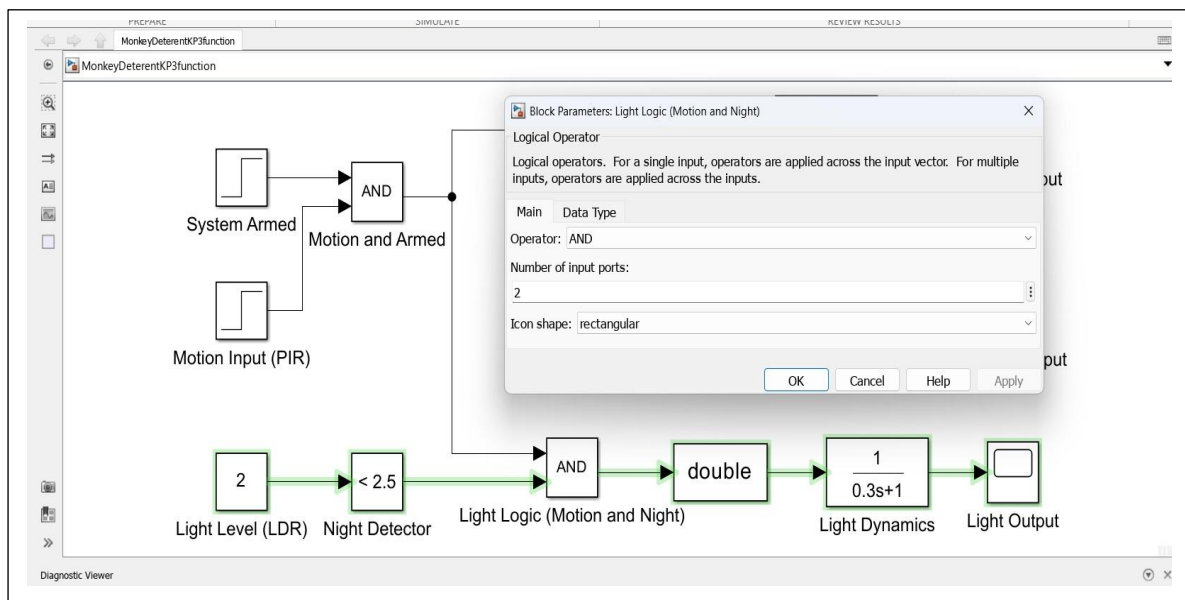
Day-Time Representation.

5.2. LogicNight Detector (<2.5) Block

Motion and Armed Block



Light Logic (Motion and Night)



5.3. Arming Timer

To prevent accidental activation when the system is first switched on, a 20 second arming delay was added.

Step Block was added with:

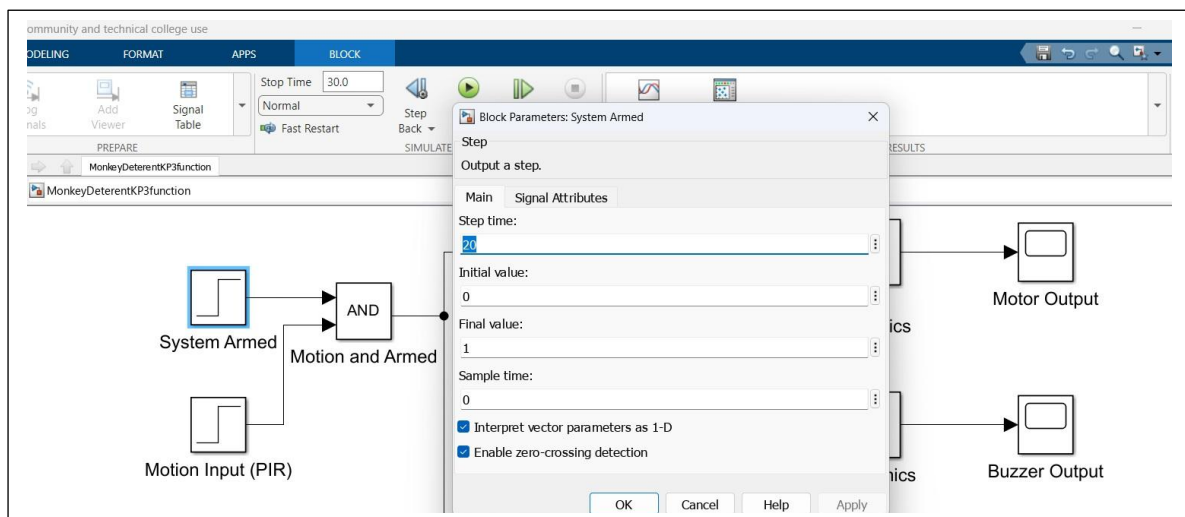
Step Time = 20s

Initial Value = 0

Final Value = 1

This signal was AND logic with the PIR motion signal to create Armed Motion.

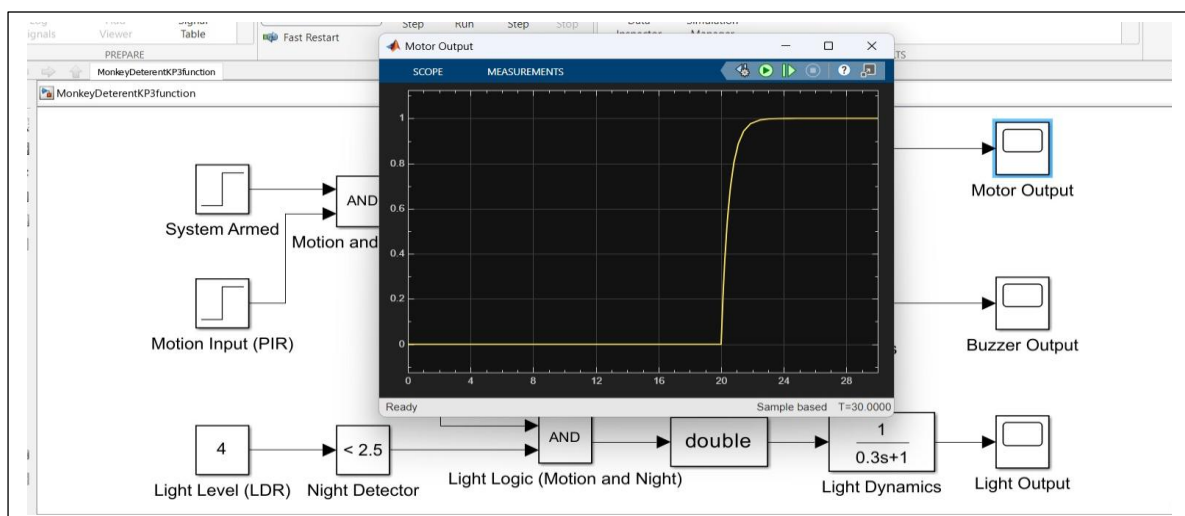
The Armed Motion signal drives all actuator logic.



Example:

Day set $LDR \geq 2.5$

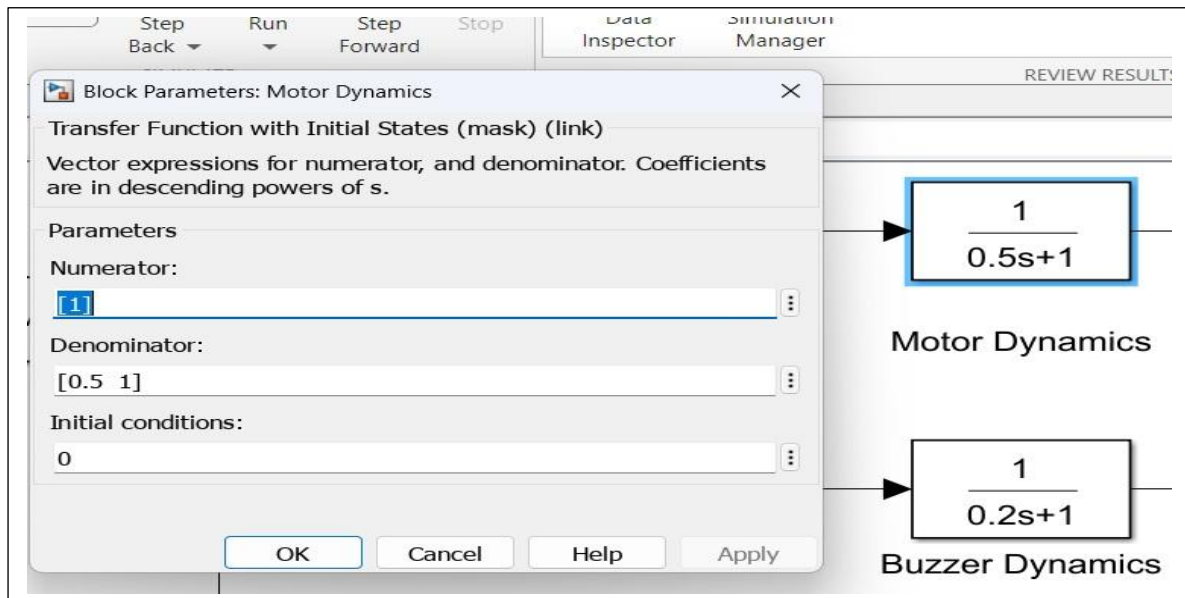
Arm count shows 20sec before motor activation.



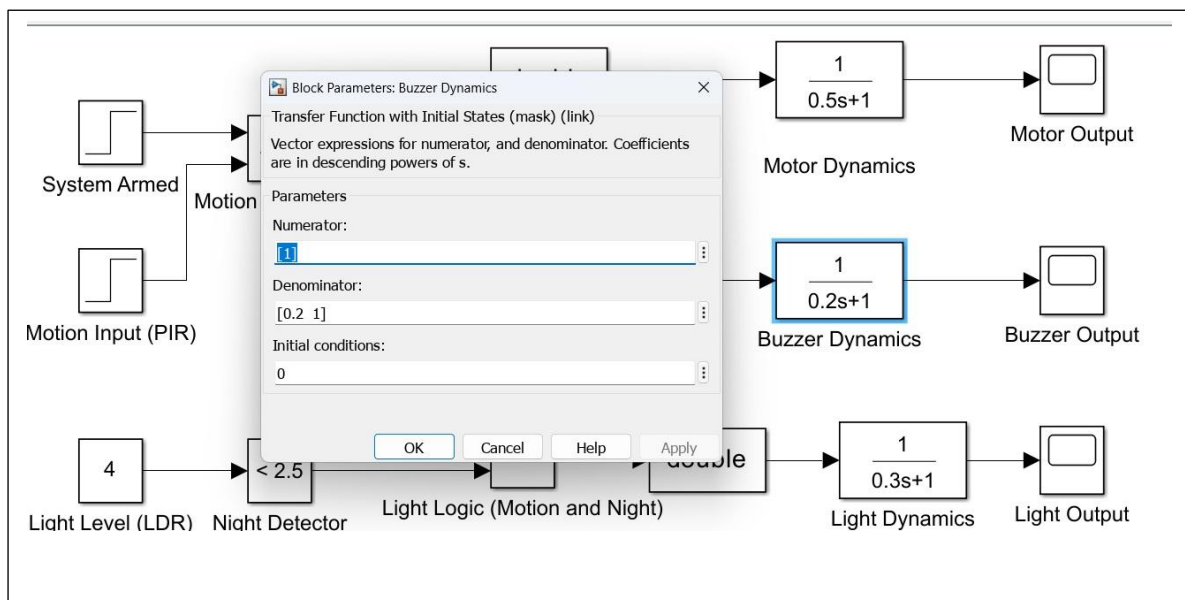
5.4. Outputs

Each output passes through a first order transfer function.

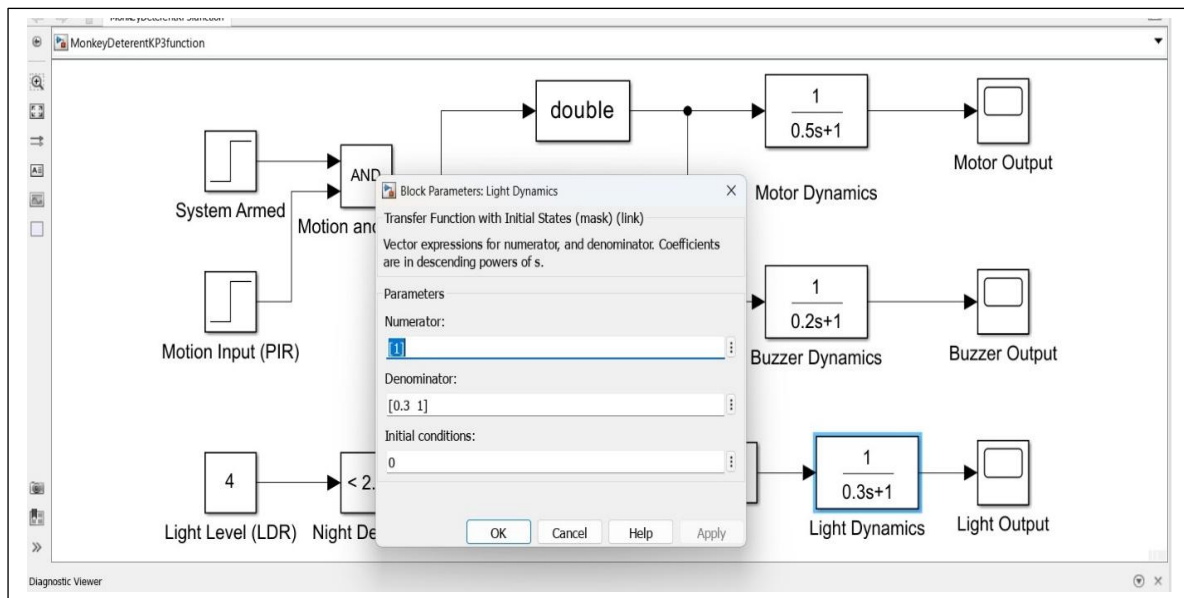
Motor Transfer Function: $\frac{1}{(0.5s+1)}$



Buzzer Transfer Function: $\frac{1}{(0.2s+1)}$



Light Transfer Function: $\frac{1}{(0.3s+1)}$



5.5. Simulation Time

The simulation was run for 30 seconds to observe pre-arming behaviour, activation after 20 seconds and dynamic rise of each actuator output.

5.6. Embedded Implementation Arduino closed-Loop System

After validating the behaviour in Simulink, the control logic was implemented on an Arduino microcontroller as a closed-loop embedded system. The Arduino continuously reads the PIR motion sensor and LDR, applies the same arming and day/night logic used in the model, and directly drives the motor, buzzer, and light outputs. This allows the deterrent to operate autonomously in the field without a connected computer, using only solar power and a battery.

The embedded implementation preserves the key behaviours observed in simulation: an arming delay before activation, motion-dependent triggering, and different output behaviour for day-time and night-time conditions. In the prototype, a 10 second arming period is used, during which a yellow LED pulses to indicate that the system is arming. Once armed, a green LED turns on, and the system only responds to motion after the PIR has returned to a low state, reducing false triggers immediately after arming. This demonstrates a clear link between the Simulink model and a practical, closed-loop hardware implementation.

6. Simulation Results

6.1. Overview

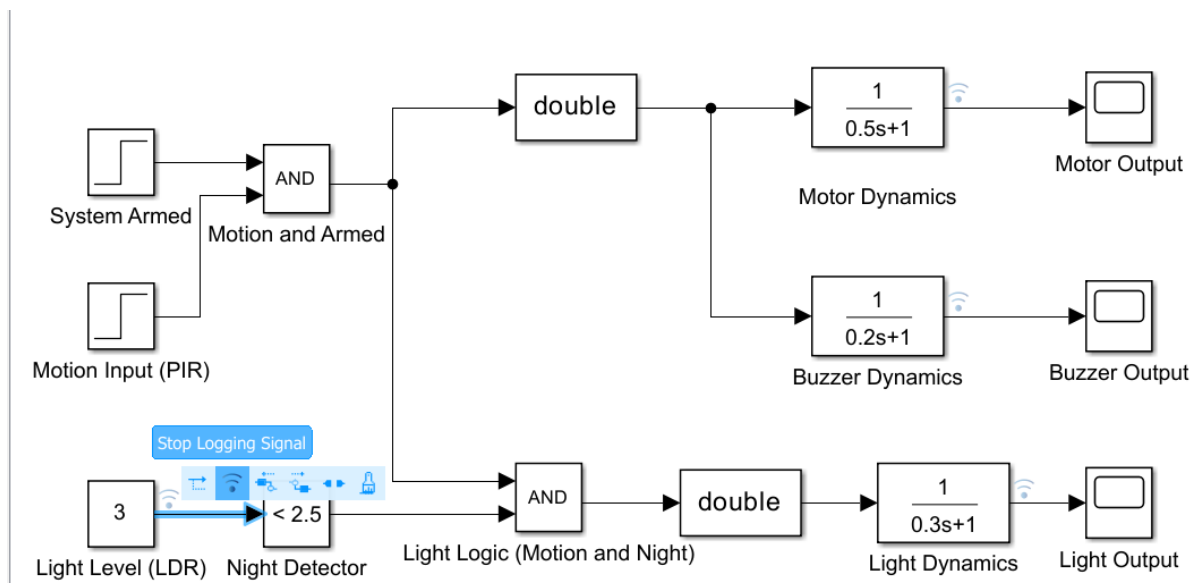
The simulation confirms that the system behaves exactly as intended (Prescott, 2026).

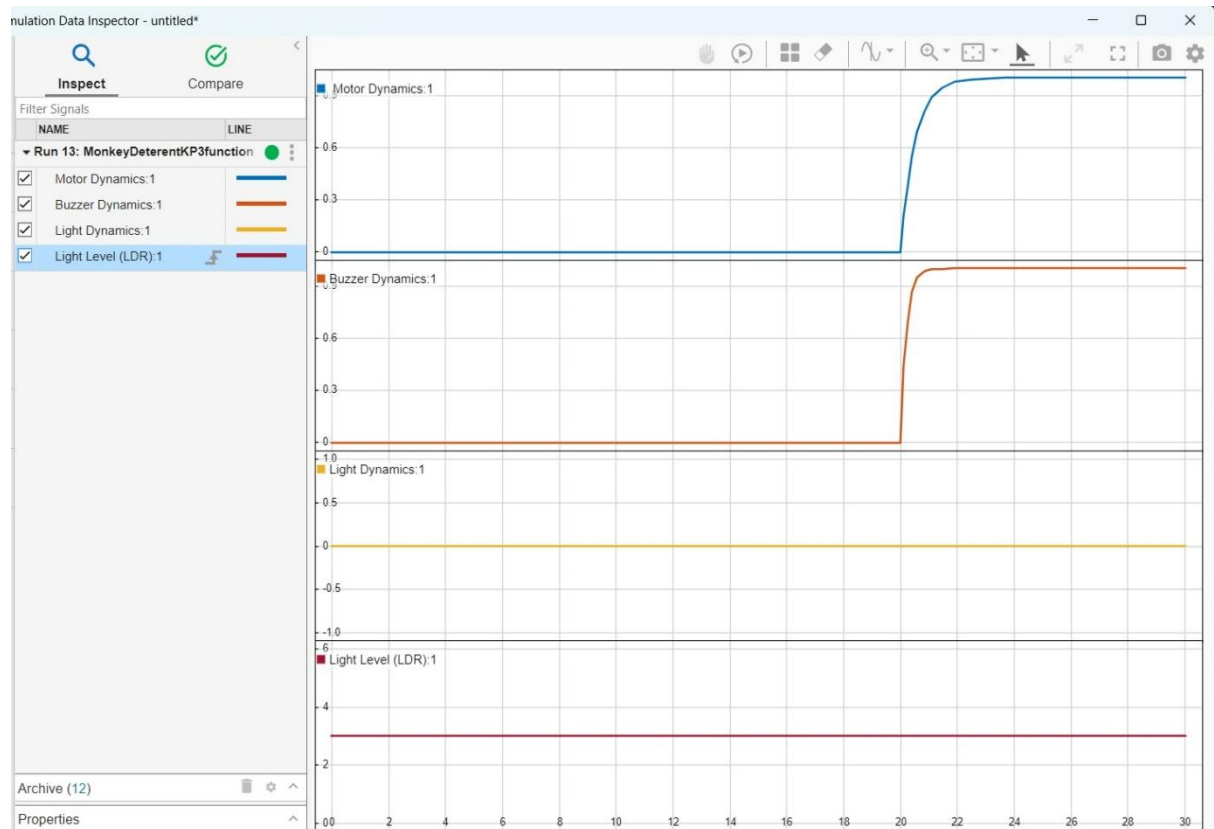
A short simulation video demonstrating the full system response, including the arming delay and actuator activation, is provided in Appendix B. All outputs remain at zero until:

1. The system is armed for 20 seconds.
2. Motion is detected
3. Night conditions are met for light to activate.

6.2. Day-Time Scenario (LDR = 3)

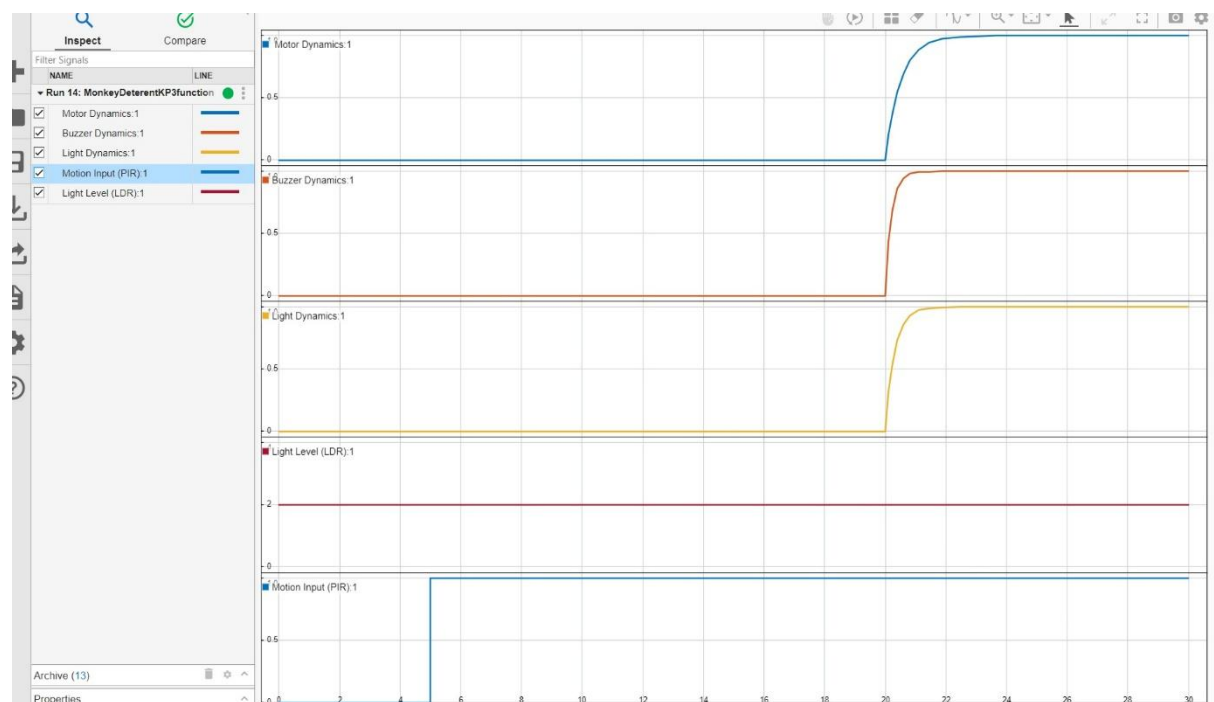
This test was carried out to simulate daytime conditions and observe how each output behaved using signal logging. All three outputs, along with LDR signal, were monitored throughout the 30 second test. Using the simulation Data Inspector made it significantly easier to review and compare all Day-Time system conditions in one place.

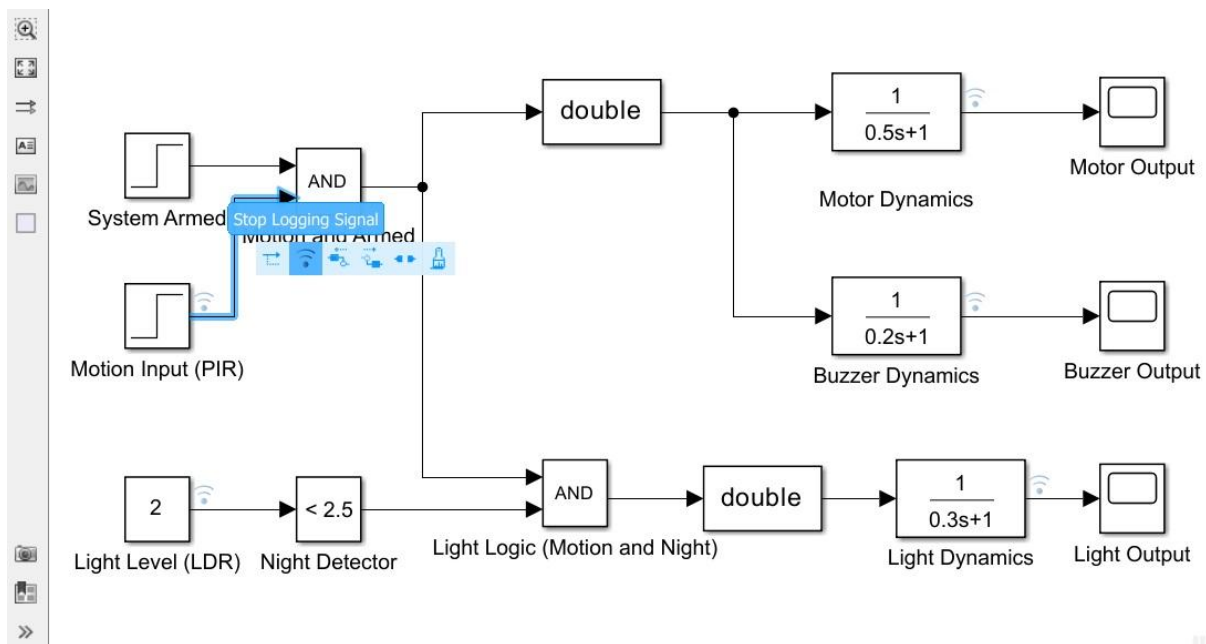




6.3. Night Time Scenario (LDR = 2)

The same test was carried out for night time conditions by adjusting the LDR to 2, ensuring that all night outputs activated as intended. Motion was added to show that step block was activating for motion after the 20 second arm count.





7. Design Justification

The design prioritises simplicity, reliability, and suitability for remote agricultural environments. Alternatives such as ultrasonic devices, water-spray systems, and analogue-only circuits were considered but rejected due to limited effectiveness, high resource requirements, or lack of flexibility.

A microcontroller-based system was chosen because it provides reliable sensor processing, supports timed logic, and enables multi-output control. The combination of mechanical movement, sound, and light creates a multi-sensory deterrent that is harder for monkeys to adapt to. The system is intentionally low power to ensure compatibility with a small solar panel and battery.

8. Modelling Assumptions

Several assumptions were made to simplify the simulation while maintaining realistic behaviour:

1. All actuators were modelled as first-order systems.
2. Motor time constant assumed as 0.5 s to represent mechanical inertia.
3. Buzzer and LED assigned shorter time constants (0.2 s and 0.3 s).
4. PIR sensor modelled as an instantaneous step input.
5. LDR modelled as a constant value for day/night scenarios.
6. Environmental factors such as temperature, sensor noise, and battery variation were excluded.

These assumptions allow the model to demonstrate system behaviour without unnecessary complexity.

9. System Limitations

The system has several limitations:

1. PIR sensors may trigger falsely due to heat sources or vegetation.
2. LDR thresholds may require seasonal adjustment.
3. Components may wear over time.
4. Solar performance depends on weather conditions.
5. Microcontroller requires basic programming knowledge.
6. Monkeys may eventually adapt to predictable deterrents.

10. Critical Evaluation of System Performance

Simulation results confirm that the system behaves as intended. The arming delay works correctly, day/night logic is reliable, and actuator outputs follow expected dynamic responses.

However, the simulation does not account for environmental noise, sensor variability, or mechanical wear. Future improvements could include:

1. Battery discharge modelling
2. Randomised output patterns
3. Additional sensors (e.g., ultrasonic or camera-based detection)

Overall, the system meets the project requirements and provides a strong foundation for real-world deployment.

The Arduino prototype was used to validate the simulated design under real operating conditions. The arming delay, motion-triggered activation, and day/night behaviour matched the Simulink model closely, although minor adjustments to the LDR threshold and arming time were required. This confirms that the closed-loop embedded implementation is consistent with the model while highlighting the importance of hardware testing for refining practical settings.

11. Components Required

To construct the monkey deterrent system, a set of electronic, mechanical, and power related components is required. These components allow the system to replicate the behaviour modelled in Simulink, with a microcontroller acting as the central controller. The following list outlines the essential hardware needed for a complete working build:

Control Processing

Microcontroller-Arduino Nano (Arduino, 2024).

Sensors

PIR motion sensor

10kΩ resistor-voltage divider

Actuator Outputs

DC motor

Motor Driver

Buzzer

High Brightness LED lamp

Power System

5w Solar Panel

Solar charge controller

1500-2000mAh Li-ion Battery

Resettable Fuse

Enclosure and Wiring

Weather-resistant IP enclosure

Hook-up wire and connectors

Cable glands

Mounting hardware

Mechanical arm + tins/metal recyclable strips

These components collectively enable the system to operate autonomously, reliably, and safely in rural Zimbabwean conditions.

12. Solar Powered Design

Although the system is being constructed in the UK, the solar design is based entirely on Zimbabwe's environmental conditions, where sunlight levels are significantly higher and more consistent throughout the year. Zimbabwe typically receives 5-7 peak sun hours per day, providing strong and reliable solar irradiance suitable for small of grid systems (Solar Electricity Handbook, 2024). This allows the deterrent system to operate effectively using a compact solar panel and a modest battery capacity.

Daily energy use is approximately 1 Wh/day, based on the combined consumption of the motor, buzzer, and light. Using Zimbabwe's high irradiance levels, a 3–5 W solar panel provides more than enough energy to recharge the system each day. A 1500–2000 mAh Li-ion battery offers 3–5 days of autonomy, ensuring the system remains operational during cloudy periods or when sunlight is reduced. The combination of strong sunlight and low power consumption makes the system highly reliable for rural Zimbabwean deployment.

13. Cost Analysis

All components required for the monkey deterrent system can be sourced easily within the UK from common suppliers such as RS Components, CPC Farnell, Amazon, and local hardware stores. Sensors, wiring, enclosures, and electronic components are inexpensive and widely available, while small solar panels, charge controllers, and hobby motors can be purchased from consumer suppliers at low cost. The overall system remains affordable, with a total build cost typically falling between £28 and £33, depending on supplier choice and availability.

14. Maintainability

The system is designed for long-term use and easy maintenance. A small microcontroller such as an Arduino would act as the control unit, running the logic developed in Simulink. These devices are low power, inexpensive, and widely supported, making them easy to replace or reprogram if required. The remaining components PIR sensor, LDR, motor, buzzer, and light are simple, modular parts that can be swapped individually without specialist tools. The mechanical noise arm can be repaired using locally available materials, and the solar system requires minimal upkeep.

15. Conclusion

This project successfully develops a low-cost, solar-powered monkey deterrent system tailored to the needs of the Chomzangari Women's Cooperative. By combining motion detection, light sensing, and multi-sensory deterrent outputs, the system provides an effective and sustainable method of reducing crop losses.

The use of first-order dynamic modelling and Simulink simulation demonstrates a clear understanding of system behaviour, actuator response, and control logic. The design is practical, maintainable, and affordable, making it suitable for deployment in remote agricultural environments.

16. References

MATLAB Documentation (2024) *Simulink User Guide*.

O'Brien, E.P. and Hill, C.M. (2018) Assessing the potential of non-harmful, natural feeding deterrents tested on captive primates. *International Journal of Pest Management*, 64(1), pp. 59–65.

Ojara, S. (2023) *Remotely controlled monkey repellent system*. [Institutional Report].

Rahim, A.M.A. and Zainal, M.S. (2024) An IoT based monkey deterrent for plantations with surveillance system: A method for damage control done by monkeys in plantations. *Progress in Engineering Application and Technology*, 5(1), pp. 114–121.

Solar Electricity Handbook (2024) *Solar irradiance data – Zimbabwe*. Available at: <https://solarelectricityhandbook.com/solar-irradiance.html> (solarelectricityhandbook.com in Bing) (Accessed: 20 February 2026).

Arduino (2024) Arduino Uno Rev3. Available at: <https://www.arduino.cc> (Accessed: 17 March 2026).

Prescott, K. (2026) Monkey Deterrent System – Simulink Simulation Video. Unpublished video recording.

Appendix A – Arduino Source Code

The following Arduino sketch implements the monkey deterrent system as a closed-loop embedded controller, using the PIR sensor, LDR, arming button, motor, buzzer, and LEDs to realise the logic described in the main report.

```
int lightValue = 0;

int threshold = 600;

bool systemArmed = false;
bool pirHasSettled = false;

void setup() {
  pinMode(2, INPUT);    // PIR
  pinMode(3, INPUT_PULLUP); // Arm/Disarm button

  pinMode(4, OUTPUT);   // Arming LED (yellow)
  pinMode(5, OUTPUT);   // Armed LED (green)

  pinMode(6, OUTPUT);   // Red LED (night indicator)
  pinMode(7, OUTPUT);   // White LED (day indicator)

  pinMode(8, OUTPUT);   // Buzzer OUTPUT NIGHT and DAY
  pinMode(9, OUTPUT);   // Motor relay OUTPUT NIGHT and DAY
  pinMode(11, OUTPUT);  // Blue LED (night deterrent) OUTPUT NIGHT ONLY

  // Startup state
  digitalWrite(4, HIGH); // Yellow ON (system idle)
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  digitalWrite(8, LOW);
  digitalWrite(9, LOW);
  digitalWrite(11, LOW);
```

```
Serial.begin(9600);  
}  
  
void loop() {  
  
    int motion = digitalRead(2);  
    int button = digitalRead(3);  
    lightValue = analogRead(A0);  
  
    bool isDark = (lightValue < threshold);  
  
    // DAY/NIGHT INDICATORS  
    if (isDark) {  
        digitalWrite(6, HIGH); // Red ON (night)  
        digitalWrite(7, LOW); // White OFF  
    } else {  
        digitalWrite(6, LOW); // Red OFF  
        digitalWrite(7, HIGH); // White ON (day)  
    }  
  
    // DISARM SYSTEM IF BUTTON PRESSED WHILE ARMED  
    if (button == LOW && systemArmed) {  
        Serial.println("System disarmed.");  
  
        systemArmed = false;  
        pirHasSettled = false;  
  
        digitalWrite(5, LOW); // Green OFF  
        digitalWrite(4, HIGH); // Yellow ON (idle/reset)
```

```
// Turn off all deterrents
digitalWrite(9, LOW);
digitalWrite(8, LOW);
digitalWrite(11, LOW);

delay(500); // Debounce
return;
}

// ARMING SEQUENCE (10 seconds with pulsing yellow LED)
if (button == LOW && !systemArmed) {
    Serial.println("Arming system...");

    unsigned long startTime = millis();
    while (millis() - startTime < 10000) { // 10 seconds
        digitalWrite(4, HIGH);
        delay(200);
        digitalWrite(4, LOW);
        delay(200);
    }

    digitalWrite(4, LOW); // Yellow OFF
    digitalWrite(5, HIGH); // Green ON (armed)

    systemArmed = true;
    pirHasSettled = false;

    Serial.println("System armed. Waiting for PIR to settle...");
}

// If system is not armed, ignore PIR
```

```
if (!systemArmed) {  
    return;  
}  
  
// WAIT FOR PIR TO GO LOW ONCE BEFORE ALLOWING TRIGGERS  
if (!pirHasSettled) {  
    if (motion == LOW) {  
        pirHasSettled = true;  
        Serial.println("PIR settled. System fully active.");  
    } else {  
        return; // Ignore triggers until PIR settles  
    }  
}  
  
// PIR-triggered deterrents (only when armed AND settled)  
if (motion == HIGH) {  
    digitalWrite(9, HIGH); // Motor  
    digitalWrite(8, HIGH); // Buzzer  
  
    if (isDark) {  
        digitalWrite(11, HIGH); // Blue LED only at night  
    }  
  
    delay(5000);  
  
    digitalWrite(9, LOW);  
    digitalWrite(8, LOW);  
    digitalWrite(11, LOW);  
  
    delay(500);  
}
```

Appendix B – Simulation Video

The simulation video demonstrates the system behaviour over a 30-second run, including the arming delay, motion activation, and day/night output logic. The video corresponds directly to the Simulink model described in Section 5 and the results presented in Section 6.