Neurorobots of the Deep: Dragonfish

Introduction

Animals in the deep ocean exhibit remarkable adaptations to survive and thrive in their harsh environments. One such example is the dragonfish, a predator with a unique body structure that is adapted to life in the deep sea. Its sleek, dark black body provides camouflage in the absence of light, while its bioluminescent lure attracts unsuspecting prey. Inspired by these traits, we designed a robot that mimics the dragonfish's predatory and defensive behaviors, incorporating key principles of neurorobotics to create an adaptive and reactive system.

Our robotic dragonfish transitions through multiple behavioral states, each driven by sensory input and internal factors such as satiety. It begins in a waiting phase, mimicking the dragonfish's "lazy" (or patient) hunting strategy by waving its "lure" and using a color sensor to identify prey. As it feeds and its hunger diminishes, the robot switches to an exploration phase, where it navigates its environment using a distance sensor to avoid obstacles. If it encounters a "predator," the robot immediately transitions to a defending/withdrawal phase, fleeing away to safety while expending energy in the process. By integrating these states, we aimed to replicate the decision-making and energy management strategies of a real dragonfish.

In building this robot, we applied several neurorobotic design principles, categorized into three major themes:

1. Embodiment and Reactions

   To achieve sensory-motor integration, the robot relies on inputs from a color sensor to detect prey and predators, and a distance sensor to avoid obstacles. These inputs directly influence the robot's motor actions, enabling it to respond dynamically to its environment. For example, the robot detects white or purple prey objects, calculates rewards, and performs precise mouth movements to "catch" its prey. The robot also has the ability to achieve multitasking and event-driven processing. The robot continuously monitors its environment for prey, predators, and obstacles while maintaining smooth transitions between states. These event-driven processes allow it to perform complex behaviors, such as deciding whether to eat or flee based on satiety and external stimuli.

2. Adaptive Behavior

   The robot uses a satiety score as a value system to transition between states and guide its decision-making. Initially, it prioritizes eating to reduce hunger. As the score rises, the robot becomes less inclined to eat, eventually shifting to exploration. This system mimics the natural balance between hunger-driven foraging and energy conservation.

3. Behavioral Trade-offs

   The robot balances invigorated and withdrawn, prioritizing survival when faced with threats, such as large "predator" objects represented by yellow targets, and exploration when faced without threats. Additionally, its satiety level governs the shift between active and passive behaviors, representing the decision of foraging and defending. When hungry, the robot passively hunts and focuses on energy conservation and accumulation. As it reaches satiety, its behavior becomes more active, focusing on exploration and energy consumption.

By incorporating these principles, we created a robot capable of dynamic decision-making, closely resembling the behavior of its biological counterpart. To evaluate the robot’s performance, we simulated the dynamic conditions a dragonfish might experience in its deep-sea environment. Due to the robot's reliance on sensors, prey and predator objects were not placed in fixed locations. Instead, objects were introduced directly in front of the robot’s color sensor at appropriate times during the trials. This approach ensured reliable detection and allowed us to closely observe the robot's responses to specific stimuli.

The experimental design aims to evaluate the robot's ability to mimic the adaptive and predatory behaviors of the dragonfish in a simulated deep-sea environment. The primary goal is to test how effectively the robot transitions between behavioral states (waiting, exploring, defending, and prey interaction) based on environmental stimuli and internal conditions. The independent variables include the type of stimuli presented to the robot, such as prey objects of varying sizes and colors (white, blue, and purple) and predator objects (yellow). The dependent variables include the robot’s responses, such as satiety score updates, and the obstacle avoidance success rates.

Methods

Robot Design

The robot was designed to replicate the dragonfish's appearance and predatory behavior, combining functional and aesthetic elements. Each core component of the robot was carefully constructed to contribute to its mobility, sensory processing, and behavioral execution.

<u>Movement System</u>

The robot's movement is powered by two large motors, each attached to a wheel and connected to ports F and E on the LEGO Spike Prime hub. While real dragonfish rely on muscle-driven move in water, the robot simulates movement through its wheels, enabling it to move forward, backward, and make precise turns. The motors are programmed to execute these movements in response to environmental stimuli and internal state changes.

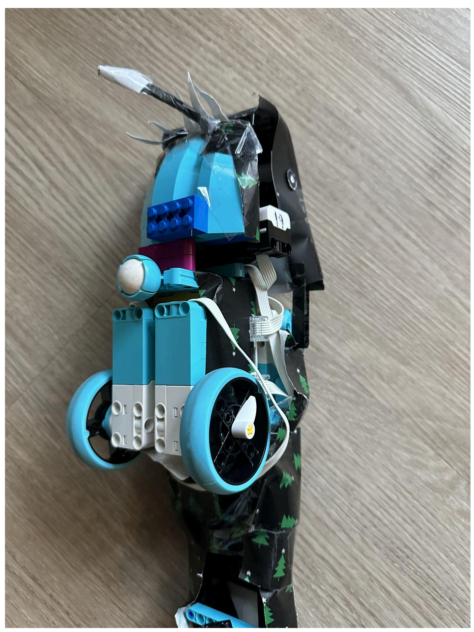

Figure 1 - Motors are attached to run the robot

<u>Sensory Module</u>

Two key sensors form the robot's sensory module, both mounted at the front of the hub:

1. Color Sensor: Attached to port A, the color sensor detects prey (represented by colored objects) and predators. It plays a crucial role in determining the robot's response, including prey capture or predator avoidance.
2. Distance Sensor: Positioned beside the color sensor and connected to port B, the distance sensor detects obstacles and enables the robot to navigate its environment safely.

Together, these sensors mimic the "eyes" of a dragonfish, allowing for simultaneous analysis of color and distance data.

Figure 2 - Layout of the sensory module, featuring the color sensor and distance sensor for prey detection and obstacle avoidance.

Mouth Mechanism

The robot's mouth is controlled by a medium motor linked to port D. This motor drives a model constructed from beams, biscuits, and technic frames, designed to simulate the dragonfish’s jaw. The mouth opens and closes during prey capture, mirroring natural feeding behavior. The jaw is represented by connecting a full shell and a right shell, which is immovable.

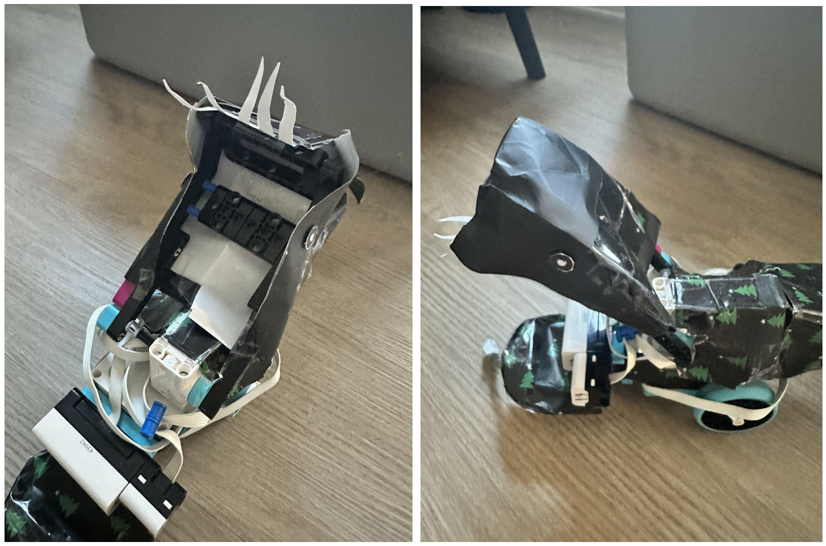

Figure 3 - Design of the mouth mechanism, simulating the dragonfish's jaw for capturing prey.

Aesthetic and Structural Features

To enhance realism, the robot's body is covered in black wrapping paper, representing the dragonfish's dark coloration for camouflage. A string with a white sticker is attached below the mouth, serving as a bioluminescent lure to attract prey. Additionally, a tail structure composed of long beams adds balance and visual resemblance to a real dragonfish.

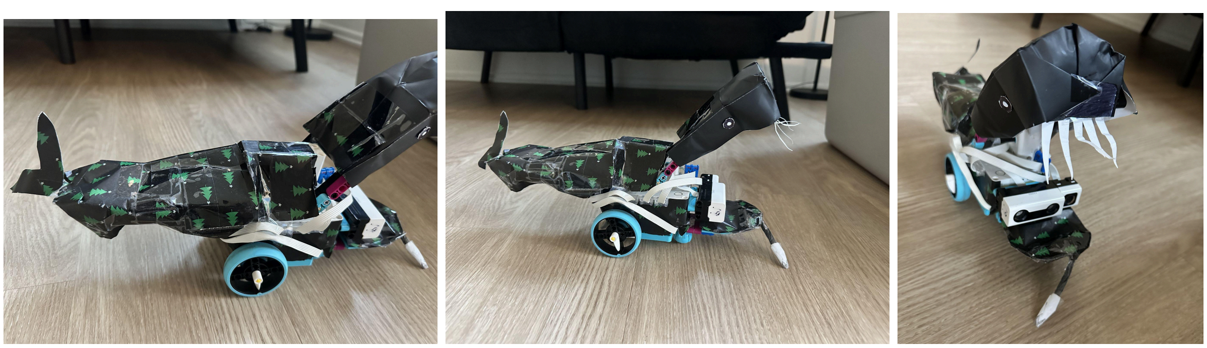

Figure 4 - Complete robot design highlighting aesthetic features, including the dark camouflage and bioluminescent lure for prey attraction.

Robot Workflow and Flowchart

The robot operates based on a state machine model, transitioning dynamically between four primary states: **WAIT**, **PREY**, **RUNAWAY**, and **EXPLORATION**. The state transitions are dictated by sensory inputs (from the color and distance sensors) and internal parameters (such as satiety score).

1. After initialization, the robot starts in the waiting phase, where it performs waltz-like movements and scans for prey using its color sensor.
2. Upon detecting prey, the robot uses its satiety score to determine whether to attempt prey capture. A positive decision transitions the robot to the prey interaction phase; otherwise, it continues waiting.
3. If the satiety score reaches 10, the robot automatically transitions to the exploration phase, where it navigates its environment while consuming energy over time. It uses the distance sensor to avoid obstacles during this phase.phase
4. Predator detection is active throughout the robot's operation. If a yellow object is detected at any time, the robot immediately transitions to the defending/withdraw phase, performing a retreat to a safe distance.
5. After completing a retreat, the robot resumes its prior state, continuing its behavior as dictated by its satiety score or current environment.
6. As the satiety score decreases due to exploration or retreating, the robot eventually returns to the waiting phase, restarting the process.

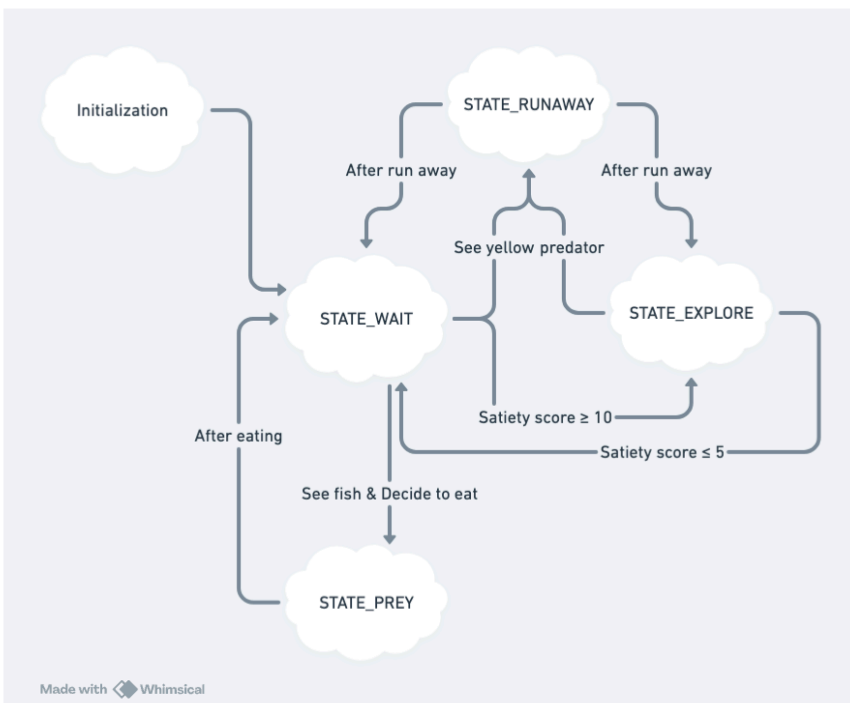

Figure 5 - Flowchart with state transitions and continuous predator detection

Code Design

The robot's behavior is implemented in Python, leveraging asynchronous functions to handle real-time sensor inputs and manage state transitions seamlessly. Each phase has a dedicated function, allowing modular implementation and testing.

Waiting Phase

The waiting phase simulates the dragonfish's behavior of holding its position in water currents while enticing prey with its bioluminescent lure. The robot performs "waltz-like" movements to mimic this underwater oscillation, such that the robot can move for not to somewhere far away. During this phase, the color sensor actively scans for potential prey or predators.

The function should_prey determines whether the robot will consume detected prey based on its current satiety score. The decision-making is influenced by a custom-designed probability function, which decreases non-linearly from 1 to 0 within the 5–10 satiety score range:

```
def check_probability(score):
        delta = (score - THRESHOLD_LOW) / (THRESHOLD_HIGH -
THRESHOLD_LOW)
    probability = 1 / (1 + math.exp(10 * (delta - 0.5)))
    return probability
```

For low scores (closer to 5), the probability of prey capture approaches 1, while for higher scores (closer to 10), the probability approaches 0.

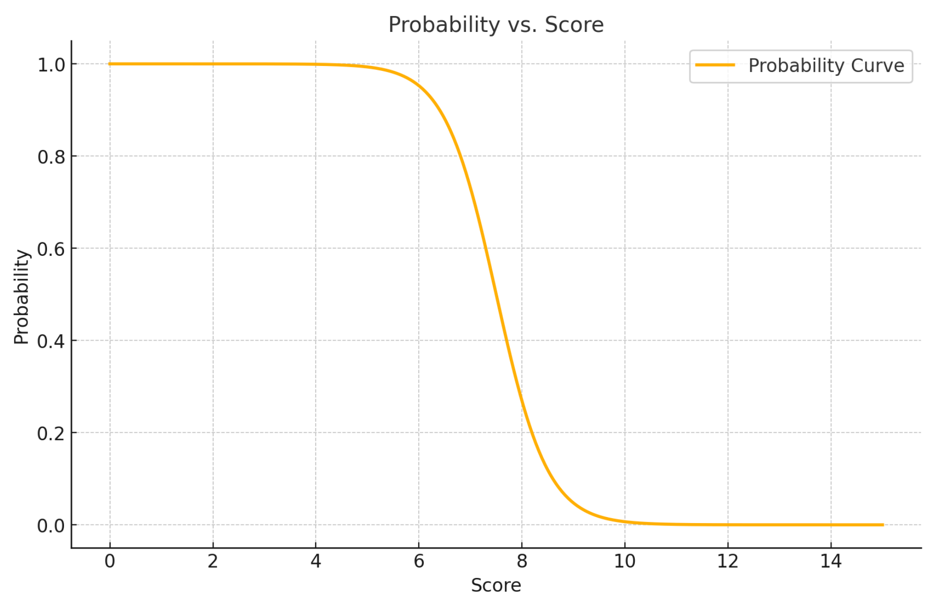

Figure 6 - Prey capture rate as a function of satiety score.

Prey Interaction

In the prey interaction phase, the robot simulates prey capture by opening and closing its mouth mechanism.

```
# Stop the robot immediately
motor_pair.stop(motor_pair.PAIR_1)
await runloop.sleep_ms(1000)
# Open mouth to prepare for eating
motor.run_for_degrees(port.D, 70, 100)# Mouth opens
print("Mouth opened to catch prey.")
await runloop.sleep_ms(1000)
# Quickly close mouth to simulate catching prey
motor.run_for_degrees(port.D, -150, 150)# Mouth closes
print("Mouth closed, prey caught.")
await runloop.sleep_ms(2000)
# Wait for 5 seconds to simulate eating
print("Eating prey... Waiting for 5 seconds.")
await runloop.sleep_ms(5000)
# Open mouth again to indicate prey has been eaten
motor.run_for_degrees(port.D, 40, 100)# Mouth opens
print("Mouth opened, prey has been eaten.")
update_satiety(current_fish)
print("Current satiety score: %i." % satiety_score)
await runloop.sleep_ms(3000)
```

The update_satiety function then adjusts the satiety score based on the detected prey's color:

```
def update_satiety(color):
    """
    Update satiety score based on the detected fish size.
    """
    global satiety_score, current_state
    if color in fish_rewards:
        satiety_score += fish_rewards[color]
        print("Satiety updated. Current score: %.2f" % satiety_score)
        # Check if satiety exceeds the high threshold
        if satiety_score > THRESHOLD_HIGH:
            current_state = STATE_EXPLORE
            print("Dragonfish is full. Switching to STATE_EXPLORE.")
```

```
        else:
            print("Unknown fish color (%d). No update." % color)
```

This process ensures realistic feeding behavior and triggers the next state based on updated hunger levels.

Exploration Phase

In the exploration phase, the robot navigates its environment autonomously while consuming energy to reduce its satiety score. The distance sensor detects obstacles, and the robot adjusts its movement based on the distance to ensure effective navigation.

```
# No obstacle detected, explore normally
forward_duration = 3000
left_speed = random.randint(150, 300)# Random speed for the left wheel (minimum 150)
right_speed = random.randint(150, 300)# Random speed for the right wheel (minimum 150)
motor_pair.move_tank(motor_pair.PAIR_1, left_speed, right_speed)
print("Exploring - Left speed: %d, Right speed: %d" % (left_speed, right_speed))
await runloop.sleep_ms(forward_duration)
```

The following logic is implemented for obstacle avoidance:

1. If no obstacle is detected, the robot moves forward with random wheel speeds to simulate exploratory movement.
2. If an obstacle is detected within 300 units, an adjustment factor is calculated based on the distance, ensuring that closer obstacles result in sharper turns:

```
adjustment = int((300 - distance) / 2)
base_left_speed = random.randint(150, 300)
base_right_speed = random.randint(150, 300)
left_speed = max(150, min(300, base_left_speed - adjustment))
right_speed = max(150, min(300, base_right_speed + adjustment))# Ensure bounds [150, 300]
motor_pair.move_tank(motor_pair.PAIR_1, left_speed, right_speed)
print("Turning - Left speed: %d, Right speed: %d" % (left_speed, right_speed))
await runloop.sleep_ms(500)# Adjust movement for 0.5 seconds to allow frequent checks
```

This approach ensures dynamic and adaptive navigation.

Defending/Withdraw Phase

The defending/withdrawal phase is triggered whenever the robot detects a predator (yellow object), regardless of its current state. The robot performs an immediate sharp turn followed by a rapid retreat:

```
turn_direction = random.choice([-1, 1])
motor_pair.move_tank(motor_pair.PAIR_1,     turn_direction     *     300,
-turn_direction * 300)
print("Turning sharply in direction: %d" % turn_direction)
await runloop.sleep_ms(1000)
forward_speed = int(500)
motor_pair.move_tank(motor_pair.PAIR_1, forward_speed, forward_speed)
print("Running away at speed: %d" % forward_speed)
await runloop.sleep_ms(random.randint(2000, 3000))
motor_pair.stop(motor_pair.PAIR_1)
await runloop.sleep_ms(1000)
```

Once the retreat is completed, the robot returns to its prior state and resumes normal operation.

```
# Switch back to exploration after running away
   if satiety_score <= 5:
       prior_state = current_state
       current_state = STATE_WAIT
   elif prior_state == STATE_EXPLORE:
       current_state = STATE_EXPLORE
       print("Switching back to STATE_EXPLORE.")
   elif prior_state == STATE_WAIT:
       current_state = STATE_WAIT
       print("Switching back to STATE_WAIT.")
```

Experimental Approach

The experiment was conducted in a controlled environment to replicate the dragonfish's deep-sea habitat. The robot's behavior was tested across multiple trials, with data collected to analyze state transitions, decision-making accuracy, and environmental interactions.

Setup

The testing area was to simulate real-life conditions of the dragonfish. Obstacles, such as water bottles and smartphones, were placed randomly to create a dynamic terrain. Colored objects were used to represent prey and predators:

- White objects represented small prey with a reward of +1 satiety score.
- Blue objects represented medium prey with a reward of +2 satiety score.
- Purple objects represented large prey with a reward of +3 satiety score.
- Yellow objects represented predators, triggering the defending/withdraw phase with no satiety gain.

Objects were introduced dynamically in front of the robot's sensors to mimic real-world encounters, ensuring consistent and reliable detection.

Procedure

1. The robot was initialized in the waiting phase, performing waltz-like movements while scanning for prey with its color sensor. Upon detecting prey, the should_prey function determined whether to consume it based on the satiety score. Successful prey capture transitioned the robot to the prey interaction phase, while rejection kept it in the waiting phase.
2. As the satiety score reached 10, the robot automatically transitioned to the exploration phase. During this phase, the robot moved autonomously, using its distance sensor to avoid obstacles. Randomized speeds for each wheel simulated natural exploratory behavior. Obstacle proximity determined the turning angle, with closer obstacles prompting sharper turns.
3. Predator detection was active throughout the experiment. Encountering a yellow object caused the robot to enter the defending/withdraw phase, executing a sharp turn and retreat. After completing the retreat, the robot resumed its prior state and continued its behavior.
4. Data was recorded continuously, including satiety score changes, stimulus interactions, state transitions, and obstacle navigation outcomes.

Each trial ran until the robot decided not to pray for consecutive 5 times or it collided with the obstacles that prevented it from continuing its movement. Data analysis focused on evaluating the robot's adaptive behavior, decision-making efficiency, and alignment with the dragonfish's biological traits.

Results

The robotic dragonfish's performance was evaluated across five trials, focusing on its behaviors in each state (WAIT, PREY, EXPLORATION, and RUNAWAY) and its ability to transition between these states based on environmental stimuli and internal satiety score. Key performance metrics, such as prey capture decisions influenced by the satiety score and obstacle avoidance during the exploration phase, were quantified and analyzed. These results reflect the robot's alignment with the biological traits of the dragonfish and its adherence to neurorobotic design principles, demonstrating its effectiveness in executing desired behaviors.

State-Specific Behavior Evaluation

1. **Waiting Phase**: During the waiting phase, the robot performed waltz-like movements, simulating the dragonfish's natural behavior of holding position in ocean currents while scanning for prey. These oscillatory motions allowed the robot to maintain its position without significant displacement. The robot's color sensor continuously monitors for the presence of prey or predators, ensuring real-time responsiveness. Upon detecting prey, the robot evaluated its satiety score to determine whether to capture it. At lower satiety scores, the robot consistently decided to capture prey, reflecting a hunger-driven response. As the satiety score increased, the robot became more selective, transitioning to adaptive decision-making that mimics natural feeding restraint in less hungry predators. This phase successfully replicated the dragonfish's patient hunting strategy and demonstrated smooth state transitions based on internal and external cues.
2. **Prey Interaction Phase**: The prey interaction phase was triggered when the robot decided to capture prey. It initiated a series of actions, including stopping its movement, opening its mouth mechanism, simulating prey capture, and closing its mouth. The satiety score was then updated based on the size and type of prey, with larger prey yielding higher rewards. This adjustment ensured accurate representation of energy acquisition during feeding. Following prey capture, the robot returned to the waiting phase, ready to scan for additional stimuli. The mechanical actions of the mouth, combined with seamless integration of sensory input and state transitions, closely mirrored the biological feeding behaviors of the dragonfish. The robot consistently exhibited reliable prey capture mechanics, enhancing the realism and functionality of this phase.
3. **Exploration Phase**: Once the satiety score reached its threshold (10), the robot transitioned to the exploration phase, mimicking a dragonfish's behavior after satiation. In this phase, the robot navigated autonomously with randomized wheel speeds, simulating exploratory movement in a

dynamic environment. The distance sensor played a crucial role in obstacle detection, allowing the robot to calculate adjustments to its path based on proximity. Most obstacles were successfully avoided, but occasional collisions occurred when the robot faced complex or rapid obstacle encounters. These collisions, while rare, demonstrated opportunities for improving sensor calibration and response times. Despite this, the robot effectively reduced its satiety score over time to reflect energy expenditure during exploration, transitioning back to the waiting phase when hunger returned. This phase highlighted the robot's ability to balance energy acquisition and consumption dynamically.

4. **Defending/Withdraw Phase**: The defending phase was activated whenever the robot detected a predator, represented by a yellow object. This phase showcased the robot's ability to prioritize survival by performing immediate retreats. Upon detecting a predator, the robot executed a sharp turn and moved rapidly away from the threat. Following the retreat, the robot seamlessly returned to its prior state, whether waiting or exploring, ensuring continuity in its behavior. The satiety score was reduced during this phase to reflect the energy cost of defensive actions, emphasizing the robot's adherence to the energy management strategies of real organisms. This phase reliably demonstrated the robot's capacity for reactive behaviors, aligning closely with the dragonfish's natural defensive responses in the presence of predators.

Action/Stimuli	Detection	Behavior	Satiety Score	Prior State	Behavior State	After State
White prey	Successful	Captured	1	WAIT	PREY	WAIT
Blue prey	Successful	Captured	3	WAIT	PREY	WAIT
Purple prey	Successful	Captured	6	WAIT	PREY	WAIT
Blue prey	Successful	Captured	8	WAIT	PREY	WAIT
Purple prey	Successful	Captured	10	WAIT	PREY	EXPLORATIO N
Exploration trial 1	N/A	Obstacle avoidance * 3	9	EXPLORATION	N/A	EXPLORATIO N
Exploration trial 1	N/A	Obstacle avoidance * 2, Collide	9	EXPLORATION	N/A	END

Table 1 - Experiment trial 1

Action/Stimuli	Detection	Behavior	Satiety Score	Prior State	Behavior State	After State

White prey	Successful	Captured	1	WAIT	PREY	WAIT
Purple prey	Successful	Captured	4	WAIT	PREY	WAIT
Purple prey	Successful	Captured	7	WAIT	PREY	WAIT
Purple prey	Successful	Captured	10	WAIT	PREY	EXPLORATIO N
Exploration trial 1	N/A	Exploration, Obstacle Avoidance	9	EXPLORATION	N/A	EXPLORATIO N
Yellow predator	Successful	Run away	6	EXPLORATION	RUNAWAY	EXPLORATIO N
Yellow predator	Successful	Run away	3	EXPLORATION	RUNAWAY	WAIT
Blue prey	Successful	Captured	5	WAIT	PREY	WAIT
White prey	Successful	Captured	6	WAIT	PREY	WAIT
Purple prey	Successful	Captured	9	WAIT	PREY	WAIT
Blue prey	Successful	Decide NO	9	WAIT	N/A	WAIT
Blue prey	Successful	Decide NO	9	WAIT	N/A	WAIT
Blue prey	Successful	Decide NO	9	WAIT	N/A	WAIT
Blue prey	Successful	Decide NO	9	WAIT	N/A	WAIT
Blue prey	Successful	Decide NO	9	WAIT	N/A	WAIT

Table 2 - Experiment trial 2

Action/Stimuli	Detection	Behavior	Satiety Score	Prior State	Behavior State	After State
White prey	Successful	Captured	1	WAIT	PREY	WAIT
Blue prey	Successful	Captured	3	WAIT	PREY	WAIT
Purple prey	Successful	Captured	6	WAIT	PREY	WAIT
Blue prey	Successful	Captured	8	WAIT	PREY	WAIT
Purple prey	Successful	Decide NO	8	WAIT	N/A	WAIT
Purple prey	Successful	Decide NO	8	WAIT	N/A	WAIT
Purple prey	Successful	Captured	11	WAIT	PREY	EXPLORATIO N
Exploration trial 1	N/A	Obstacle avoidance * 3	9	EXPLORATION	N/A	EXPLORATIO N
Yellow	Successful	Run away	6	EXPLORATION	RUNAWAY	EXPLORATIO

predator						N
Exploration trial 2	N/A	Obstacle avoidance * 2	5	EXPLORATION	N/A	WAIT
Blue prey	Successful	Decide NO	5	WAIT	N/A	WAIT
White prey	Successful	Captured	6	WAIT	PREY	WAIT
Purple prey	Successful	Captured	9	WAIT	PREY	WAIT
Blue prey	Successful	Decide NO	9	WAIT	N/A	WAIT
Blue prey	Successful	Decide NO	9	WAIT	N/A	WAIT
Blue prey	Successful	Decide NO	9	WAIT	N/A	WAIT
Blue prey	Successful	Decide NO	9	WAIT	N/A	WAIT
Blue prey	Successful	Decide NO	9	WAIT	N/A	END

Table 3 - Experiment trial 3

Action/Stimuli	Detection	Behavior	Satiety Score	Prior State	Behavior State	After State
White prey	Successful	Captured	1	WAIT	PREY	WAIT
Blue prey	Successful	Captured	3	WAIT	PREY	WAIT
Yellow predator	Successful	Run away	1	WAIT	RUNAWAY	WAIT
Purple prey	Successful	Captured	4	WAIT	PREY	WAIT
White prey	Successful	Captured	5	WAIT	PREY	WAIT
Blue prey	Successful	Captured	7	WAIT	PREY	WAIT
Purple prey	Successful	Captured	10	WAIT	PREY	EXPLORATION
Exploration trial 1	N/A	Obstacle avoidance * 2	9	EXPLORATION	N/A	EXPLORATION
Yellow predator	Successful	Run away	6	EXPLORATION	RUNAWAY	EXPLORATION
Exploration trial 2	N/A	Obstacle avoidance * 3, Collide	6	EXPLORATION	N/A	END

Table 4 - Experiment trial 4

Action/Stimuli	Detection	Behavior	Satiety Score	Prior State	Behavior State	After State
White prey	Successful	Captured	1	WAIT	PREY	WAIT
Blue prey	Successful	Captured	3	WAIT	PREY	WAIT
Purple prey	Successful	Captured	6	WAIT	PREY	WAIT
Blue prey	Successful	Decide NO	6	WAIT	N/A	WAIT
Purple prey	Successful	Captured	9	WAIT	PREY	WAIT
Purple prey	Successful	Captured	12	WAIT	PREY	EXPLORATIO N
Exploration trial 1	N/A	Obstacle avoidance * 2	9	EXPLORATION	N/A	EXPLORATIO N
Exploration trial 1	N/A	Obstacle avoidance * 2, Collide	9	EXPLORATION	N/A	END

Table 5 - Experiment trial 5

Prey Capture Decisions and Satiety Score

The robot's decisions to capture prey were strongly influenced by its satiety score, as designed. The analysis reveals a clear relationship between the satiety score and the likelihood of prey capture. At lower satiety scores (0–4), the robot consistently captured prey, reflecting its hunger-driven behavior. As the satiety score increased, the capture rate declined, demonstrating the robot’s selective behavior in line with its programming.

Satiety Score	Total Decisions	Captured	Capture Rate(%)
0-4	12	12	100
5	3	2	66.7
6	5	4	80
7	1	1	100
8	3	1	33
9	11	1	9.1

Table 6 - Summary of prey capture decisions across all trials

The data strongly aligns with the robot’s designed probability function, which predicts a declining capture rate as the satiety score increases. At satiety scores 0–4, the robot achieves a perfect capture rate of 100%, reflecting its hunger-driven priorities. Between satiety scores 5 and 6, the capture rates stabilize at around 70–80%, showing that the robot remains moderately active in capturing prey. At higher satiety scores (8–9), the capture rate drops sharply to 33.3% and 9.1%, indicating a significant decline in prey capture behavior as the robot approaches satiation.

The data point for satiety score 7 shows a 100% capture rate, but this is based on a single decision and is thus not representative. Excluding this outlier, the overall trend remains consistent and smooth, reinforcing the reliability of the probability function. The observed capture rate curve aligns closely with the expected behavior dictated by the robot's internal scoring mechanism, with a strong correlation between actual and predicted values.

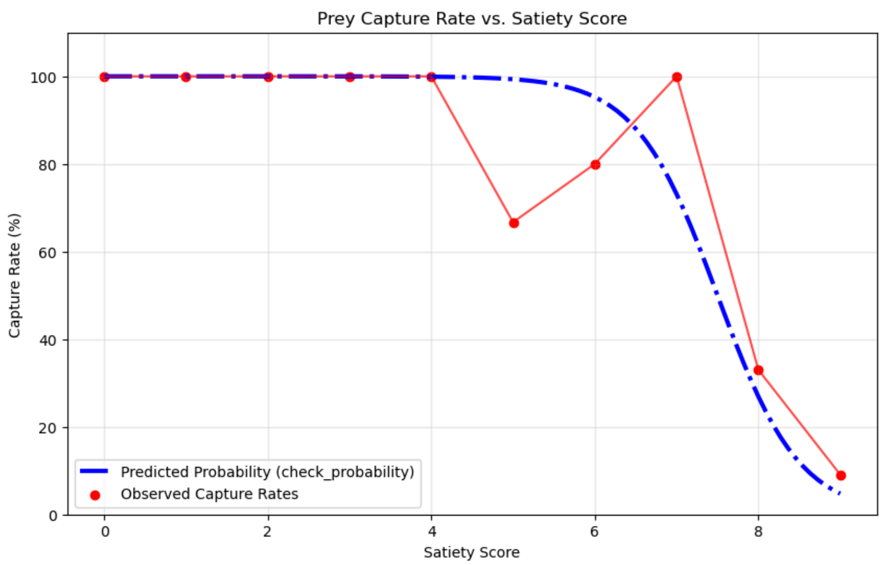

Figure 7 - Comparison between the robot's prey capture rates and the probability function curve.

Obstacle Avoidance Performance

During exploration, the robot's distance sensor enabled it to navigate effectively, avoiding obstacles in most encounters. However, occasional collisions highlighted areas for potential improvement.

- Total Obstacle Encounters: 18
- Successful Avoidances: 14 (77.8%)
- Collisions: 4 (22.2%)

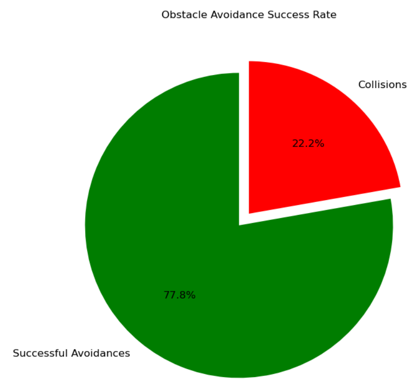

Figure 8 - Obstacle avoidance success rate during the exploration phase. The pie chart highlights a 77.8% success rate versus a 22.2% collision rate.

Conclusions

The results demonstrated that the robot effectively adhered to its programmed behavior, aligning closely with the biological traits of the dragonfish and the principles of neurorobotics. Its ability to transition seamlessly between states, adapt its decision-making based on internal and external cues, and balance behaviors like exploration and defense validated the success of its design.

The project successfully replicated dynamic and biologically inspired behaviors, offering insights into adaptive robotic systems and opportunities for future enhancements.

Discussion and Conclusions

Adherence to Neurorobotic Design Principles

The robotic dragonfish closely adhered to the principles of neurorobotics, particularly in the domains of Embodiment and Reactions, Adaptive Behavior, and Behavioral Trade-Offs, as detailed below:

- **Embodiment and Reactions**
  - **Embodiment**: The robot's structural design mimicked the dragonfish's streamlined body and predatory features, including a functional mouth mechanism for prey capture and a dark exterior for camouflage. Its waltz-like movements during the waiting phase replicated the dragonfish's natural oscillatory behavior in water currents.
  - **Sensory-Motor Integration**: The integration of a color sensor for detecting prey and predators, combined with a distance sensor for obstacle avoidance, allowed the robot to dynamically interact with its environment. These inputs directly influenced motor outputs, enabling immediate, context-appropriate responses.
  - **Multitasking and Event-Driven Processing**: The robot continuously monitored for prey, predators, and obstacles while transitioning seamlessly between behavioral states. This enabled it to prioritize tasks dynamically, such as fleeing from predators during exploration, while maintaining adaptive responses to multiple stimuli.
- **Adaptive Behavior**
  - **Value Systems**: The robot's satiety score functioned as a dynamic value system, driving its transitions between foraging, exploration, and energy conservation behaviors. This score-based mechanism mirrored the energy-management strategies of natural predators.
  - **Learning and Prediction**: While not actively learning, the robot simulates decision-making by responding to satiety levels and environmental stimuli, predicting actions based on pre-programmed logic.
- **Behavioral Trade-Offs**
  - **Invigorated vs. Withdrawn:** The robotic dragonfish exhibited a dynamic balance between invigorated and withdrawn behaviors, adjusting its activity based on internal satiety and external stimuli. In invigorated states, such as prey capture and exploration, the robot actively engaged with its environment, expending energy to acquire resources and navigate obstacles. Conversely, in withdrawn states, such as predator avoidance or waiting, the robot prioritized energy conservation, reducing unnecessary actions while maintaining situational awareness.

This adaptability mirrors biological systems, where organisms transition between high-energy activities and passive states to optimize survival.

- **Foraging vs. Defending:** The robot demonstrated a clear trade-off between foraging and defending behaviors. When conditions were safe, it actively pursued prey, maximizing resource acquisition to increase its satiety. Upon detecting a predator, however, the robot immediately transitioned to defensive behavior, retreating to ensure survival. This prioritization reflects the natural balance in organisms between energy acquisition and self-preservation in response to environmental threats.

By combining these principles, the robot exhibited behaviors and decision-making processes that closely paralleled its biological counterpart, underscoring the effectiveness of neurorobotic methodologies in replicating adaptive systems.

Performance Evaluation

The robot's performance was evaluated across five trials, revealing several strengths and opportunities for improvement in its behavior and responsiveness:

- **Prey Capture Performance:** The robot's decision-making is closely aligned with its programmed satiety score logic. At low satiety scores (0–4), the robot achieved a perfect prey capture rate of 100%, reflecting its hunger-driven priorities. As the satiety score increased, the capture rate gradually declined, dropping to 33.3% at a score of 8 and 9.1% at a score of 9. These trends validated the custom probability function, which ensured a biologically inspired transition from active feeding to selective restraint as the robot approached satiation.
- **Obstacle Avoidance and Exploration:** During the exploration phase, the robot successfully avoided 77.8% of obstacles encountered, demonstrating effective use of its distance sensor and path-adjustment logic. However, 22.2% of encounters resulted in collisions, particularly in scenarios involving consecutive obstacles or walls. This suggests limitations in the robot's spatial awareness and highlights the need for improvements in sensor coverage or multi-sensor integration.

We've also checked the predator avoidance across the trials. The defending phase consistently triggered upon detecting yellow predator objects, leading to immediate retreats. This behavior demonstrated the robot's ability to prioritize survival over other tasks, aligning with the energy-costly but necessary nature of defense mechanisms in biological systems. Post-retreat transitions to prior states were seamless, ensuring continuity in behavior.

Comparison to Biological Organisms

The robotic dragonfish successfully mirrored key aspects of its biological counterpart:

- **Predatory Strategy**: The robot's satiety-driven foraging behavior closely resembled the dragonfish's energy-maximizing hunting strategy, shifting from opportunistic feeding to exploration based on internal energy levels.
- **Survival Responses**: Its rapid retreats in response to predators paralleled natural defensive behaviors, emphasizing the importance of survival over energy acquisition.
- **Energy Management**: The robot balanced energy acquisition (prey capture) with expenditure (exploration and defense), effectively simulating the trade-offs faced by real organisms in resource-limited environments.

Despite these successes, the robot's single-sensor system and limited spatial awareness constrained its ability to navigate complex environments, whereas biological dragonfish rely on multi-modal sensory inputs to adapt to the dynamic conditions of the deep sea.

Trials and Tribulations

- **What Worked**
  - **State Transitions**: The robot transitioned seamlessly between behavioral states, accurately reflecting its programmed logic and environmental stimuli.
  - **Prey Capture Decisions**: The satiety score function operated as intended, producing biologically plausible trends in prey capture rates.
  - **Obstacle Avoidance**: The robot's distance sensor and adjustment algorithm proved effective in most scenarios, with a success rate of nearly 78%.
- **What Didn't Work, Why and How**
  - **Obstacle Navigation in Dense Environments:** Collisions occurred primarily during exploration in areas with densely placed obstacles. The robot's current single-distance sensor struggled to differentiate between obstacles requiring minor adjustments and those necessitating more significant course changes. Implementing dual-distance sensors or improving path-planning algorithms could enhance its spatial awareness and response accuracy.
  - **Sample Size and Prey Capture Trends:** While the robot's prey capture behavior aligned with expectations, certain outliers (e.g., a single decision at satiety score 7) indicated the need for larger datasets to ensure statistical reliability. Future trials with more prey interactions across a wider range of satiety scores would yield deeper insights into its decision-making dynamics.

- **Energy Efficiency in Exploration:** During prolonged exploration phases, the robot occasionally displayed inefficient energy use, such as repeated small adjustments that consumed time and power. Optimizing the exploration algorithm to include rest periods or energy-efficient movement patterns would better reflect real dragonfish behaviors.

Conclusion

The robotic dragonfish effectively adhered to neurorobotic design principles, emulating biologically inspired behaviors such as satiety-driven prey capture, dynamic obstacle navigation, and predator avoidance. Its performance validated the functionality of its decision-making algorithms, showcasing its ability to adapt to environmental stimuli and internal states. Despite challenges such as limited spatial awareness and energy efficiency, the robot demonstrated a compelling capacity to replicate complex natural systems. This project serves as a strong foundation for future innovations in adaptive robotic design, with opportunities for enhancement through multi-sensor integration and refined behavioral algorithms.

Video and Code

Code has been uploaded to Github: https://github.com/AiiiiDannn/Dragonfish.

Video can be found via: https://youtu.be/dhzvL5DYvKk