Backward walking via descending control in response to mechanical touch

Daniel Abraham Elmaleh
Imane Jennane

24th May 2024

Controlling Behavior in Animals and Robots
BIOENG-456

Prof. Pavan Ramdya

pavan.ramdya@epfl.ch

TAs: Alfred Stimpfling & Tommy Lam

Contents

Abstract

Abstract

Most land animals normally walk forward but switch to backward walking upon sensing an obstacle or danger in the path ahead. A change in walking direction is likely to be triggered by descending “command” neurons from the brain that act upon local motor circuits to alter the timing of leg muscle activation. Here we identify descending neurons for backward walking in Drosophila—the MDN neurons. MDN activity is required for flies to walk backward when they encounter an impassable barrier and is sufficient to trigger backward walking under conditions in which flies would otherwise walk forward.[1]

This study will focus on the approch used by flies in order to avoid obstacles. For this, we designed a virtual environment with curved walls that will trigger the upper part of the antennae, more precisely, left and right Arista of the Drosophilia. This sensory input causes the fly to change direction randomly, allowing it to continue walking without colliding with the walls.

1 Introduction & Background

Backward walking in response to mechanical touch is a survival mechanism observed across various species, from simple invertebrates to more complex mammals. This behavior allows an organism to retreat rapidly when faced with obstacles or potential threats. The mechanisms behind this action involve intricate neural circuits and sensory feedback systems. In Drosophila melanogaster, or fruit flies, this type of locomotion is essential for avoiding danger and navigating complex environments.

The neural basis of backward walking has been an area of considerable interest. Studies have shown that this behavior is controlled by descending neurons from the brain that send commands to local motor circuits in the thoracic ganglia. For instance, a study by Bidaye et al. (2014) identified specific descending neurons, known as MDNs (moonwalker descending neurons), that are critical for triggering backward walking in Drosophila. Activation of these neurons causes the flies to walk backward, while inhibiting these neurons prevents the flies from walking backward, demonstrating the important role these neurons play in controlling backward locomotion.[1]

In another significant development, Gunel et al. (2019) introduced DeepFly3D, a deep learning-based approach for 3D limb and appendage tracking in Drosophila. This technology enables precise measurement of limb positions, in order to study complex behaviors like backward walking. By utilizing multiple camera images, DeepFly3D can automatically detect and correct pose estimation errors, providing a detailed understanding of the neural control of locomotion.[2]

The study by Bidaye et al. (2014) provided a foundational understanding of the neural circuits involved in backward walking. They employed optogenetics to selectively activate and inhibit neurons, elucidating the roles of MDNs and MANs (moonwalker ascending neurons) in locomotor control. Their findings highlighted that while MDNs initiate backward walking, MANs promote sustained backward movement by inhibiting forward walking circuits.[1]

Gunel et al. (2019) extended this knowledge by applying advanced imaging and tracking techniques to study Drosophila's locomotion in greater detail. Their DeepFly3D system allowed for the accurate tracking of limb movements in three dimensions, facilitating a more comprehensive analysis of how neural commands translate into physical movements.[2]

Despite these advancements, several questions remain unanswered:
1. How do the sensory signals from mechanical touch integrate with descending control pathways to produce coordinated backward walking?
2. What specific neural circuits are involved in the initiation and maintenance of backward walking?
3. Are there additional neurons or neural circuits that play a role in backward walking, beyond the MDNs and MANs identified so far?
4. How do environmental factors such as terrain complexity and presence of multiple obstacles influence the neural control of backward walking?

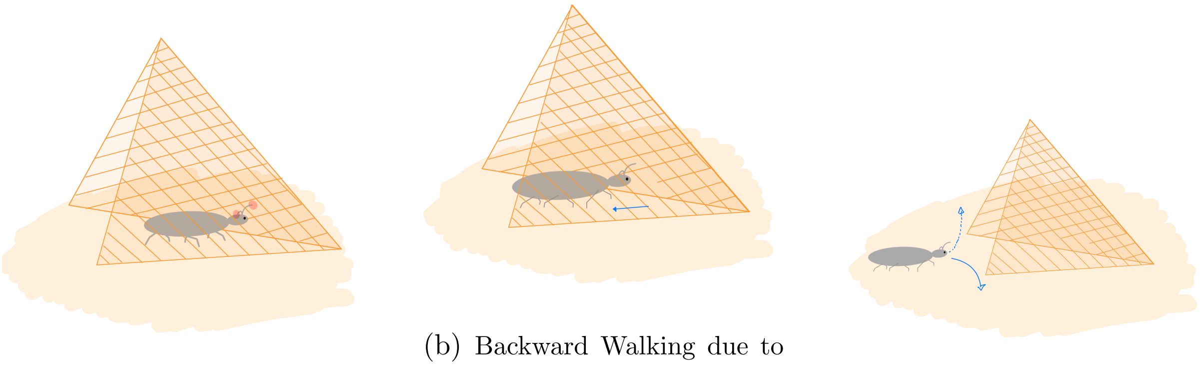
2 Methods

2.1 Scenario

2.1.1 MuJoCo Tools

To set up an appropriate environment where the fly can move in random directions and then collide with an object, we aimed to construct two inclined planes that form an angle. This setup is designed to simulate a scenario where the fly encounters an obstacle and adjusts its movement accordingly.

The imagined setup is illustrated in the following figure:



(a) Contact with arena

(b) Backward Walking due to contact

(c) Random turning direction

Figure 1: Steps of the algorithm: (a) Contact with the arena, (b) Initiation of backward walking due to contact, (c) Random turning direction following the backward walking.

We used the plane creation and arena creation functions to produce our desired environment, specifically the root-element.worldbody attribute of the 'FlatTerrain' class. Additionally, we created new camera definitions in the XML file to capture the arena during the debugging process, which helped us understand the effects of various parameters. During this process, we encountered a bug, which we describe later in the report.

Although we could have used antennal deflection as sensory input or built a more complex environment using derivatives of the applied force to distinguish between regular contact and encountering an obstacle, we chose our solution for its simplicity and intuitive, realistic aspect. Similar to how humans sense force as sensory input, our approach emphasizes practicality.

2.2 Sensory

2.2.1 NeuroMechFly Mechanosensors Involved in Backward Walking

In NeuroMechFly, we focused on the antennae as the primary mechanosensors for detecting obstacles and initiating backward walking. Specifically, the Left Arista Sensor (LArista) and the Right Arista Sensor (RArista), located on the upper part of the antennae, were chosen for their sensitivity to mechanical touch. These sensors effectively detect physical contact with obstacles in the environment, providing crucial feedback to trigger the backward walking response.

We utilized contact forces detected by placing a force sensor on the antennal extremities, known as the arista. To visually indicate contact, we implemented a color modification of the arista to red upon detection of contact, which is visible in the simulation. A challenge we encountered was that the head and front legs often touched the wall first. We resolved this by maximizing the inclination of the obstacle plane and ignoring sensors other than the arista. This approach ensured that the simulation only considered the obstacle after contact by the arista, occurring after the front legs had already made contact. This method allows the simulation to represent a discriminatory sensor input, provoking backward walking only upon antennal touch.

We set a threshold for the sensory input, determined empirically by running the simulation several times without triggering backward walking. This process allowed us to calibrate the arista sensors' response accurately.

3 Control

When the algorithm initiates, it places the fly in a random position within the environment. Following this, the fly begins to walk forward. Depending on its initial position, it will eventually collide with an object in the environment, either sooner or later. In most cases, the fly's position at the moment of collision is not perfectly symmetrical. This asymmetry means that one of the fly's antennae will make contact with the object before the other.

This initial contact triggers a specific process where the fly responds by walking backward. This behavior is crucial for the algorithm's functionality and is based on the fly's natural response to obstacles.

To accurately determine which antenna is affected, the algorithm compares the force exerted on each antenna. This comparison is essential because the response of the fly depends on identifying the antenna experiencing the greater force. To achieve this, the algorithm calculates the Euclidean norm of the force's (x, y, z) coordinates for each antenna. The Euclidean norm provides a scalar value representing the magnitude of the force applied to each antenna.

By comparing these magnitudes, the algorithm can identify the antenna that undergoes the greatest force. This precise identification allows the fly to react appropriately, ensuring that it walks backward in response to the collision. This process is repeated each time the fly encounters an obstacle, allowing for continuous interaction with the environment.

We used a simple decision tree that first checks the forces, comparing them to the threshold and to each other. Based on this comparison, the turning controller imposes directionality by modifying the intrinsic frequencies and inverting them. Furthermore, to imitate the behavior of real flies walking backward (as described in the paper [2]), we modified the amplitude of the joint angles of the 7 degrees of freedom (DOF) of each leg, but only for the front and back legs. Specifically, we amplified the coxa pitch of the hind legs and diminished the coxa pitch of the front legs. Apart from making them a contact force sensory input, we did not modify the antennae for obstacle sensing.

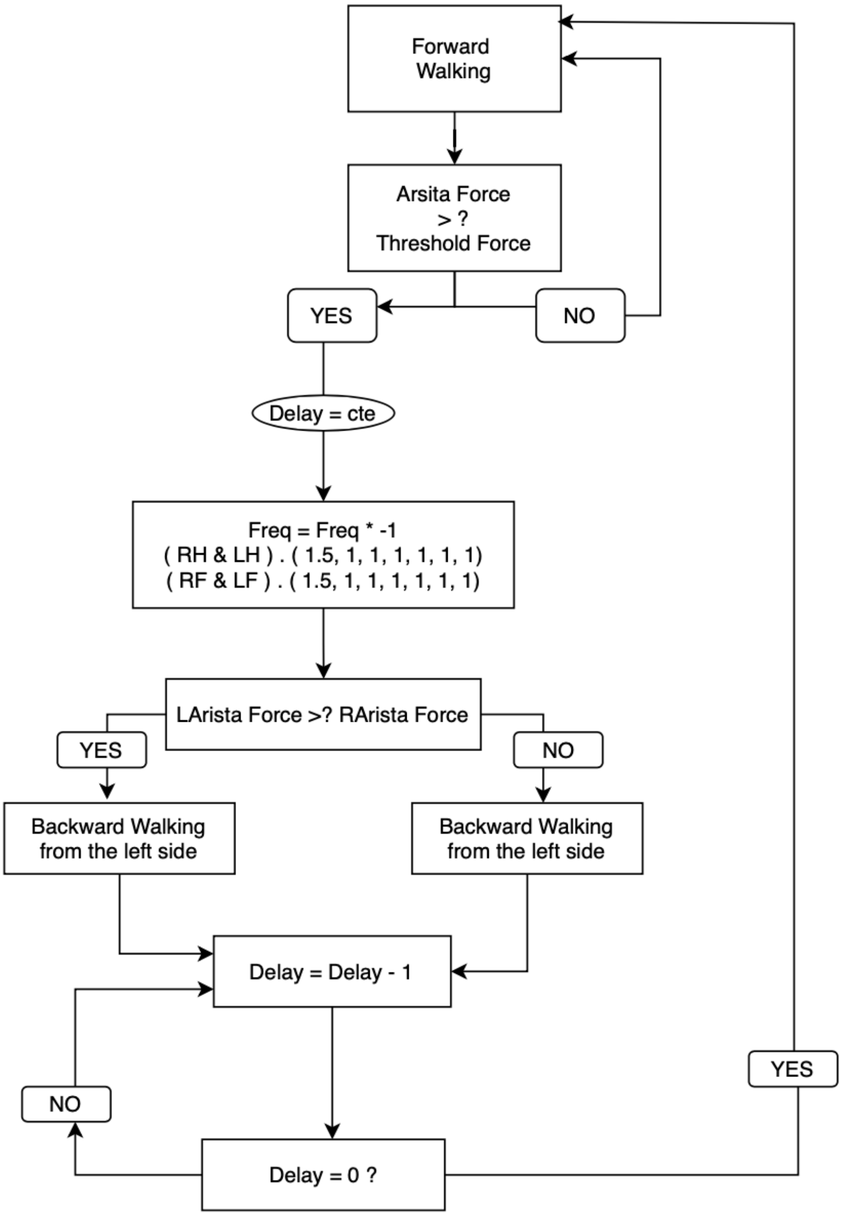


Figure 2: Decision Diagramm

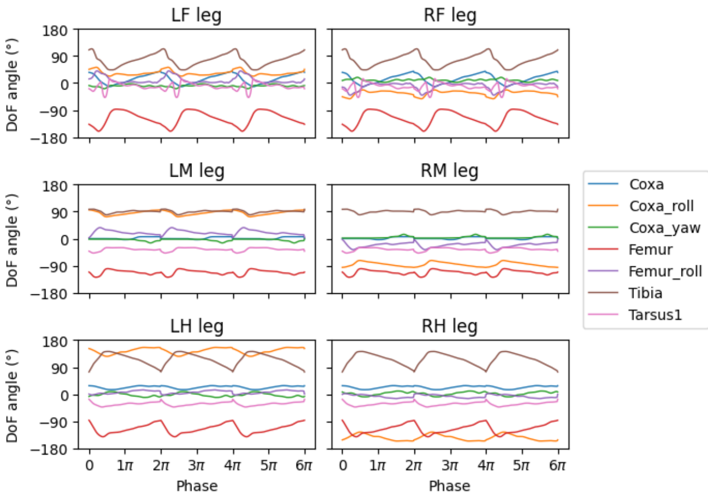
Algorithm 1 CPG Simulation

```
Initialize delay_after_collision, force_threshold, is_backward_walking, delay_counter
for each simulation step do
    observation ← get_observation(current_step)
    if left_contact_force > force_threshold or right_contact_force > force_threshold then
        if not is_backward_walking or (delay_counter < delay_after_collision and delay_counter > 0) then
            is_backward_walking ← True
            reverse(cpg_network_frequencies)
            if right_contact_force > left_contact_force then
                change_color("RightSensor", red)
                action ← [1.2, 0.4]
            else if left_contact_force > right_contact_force then
                change_color("LeftSensor", red)
                action ← [0.4, 1.2]
            end if
            delay_counter ← delay_counter - 1
        end if
    else if is_backward_walking then
        delay_counter ← delay_after_collision
        is_backward_walking ← False
        reverse(cpg_network_frequencies)
        change_color("LeftSensor", blue)
        change_color("RightSensor", blue)
    else
        is_backward_walking ← False
        reset_sensor_contacts()
        change_color("LeftSensor", blue)
        change_color("RightSensor", blue)
    end if
    execute_action(action)
end for
```

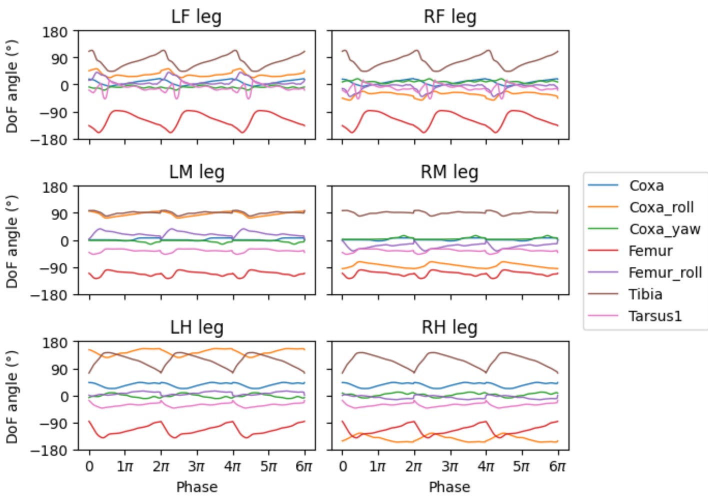
4 Results

The findings from our analyses are summarized in the following figures. Each figure is accompanied by a clear and detailed legend.

We can notice in those figures the slight difference in amplitude of the joint angles for the 2 front and hind legs. cased artificially to mimic the real behavior in backward walking flies.



(a) DoF angle for forward walking

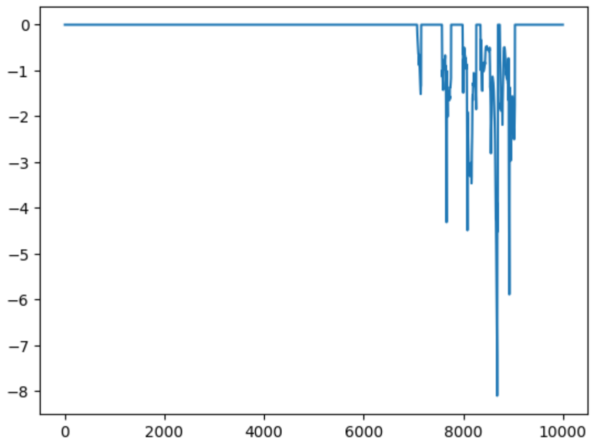


(b) DoF angle for backward walking

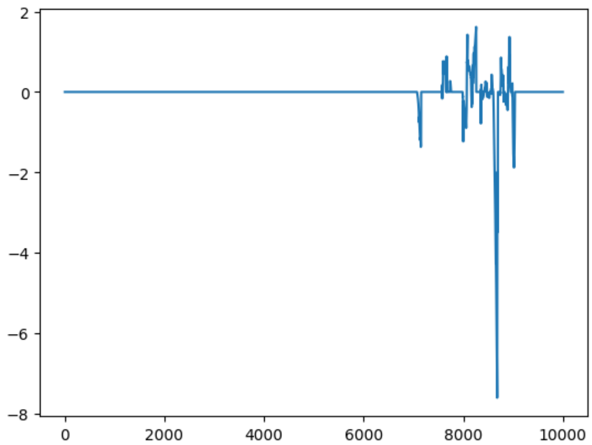
Figure 3: Angles DoF

We have plotted the outcome of our simulation in which we tested the response of the antennae sensors when colliding with the plane. The goal was also to identify and approximate a useful threshold for triggering the backward walking.

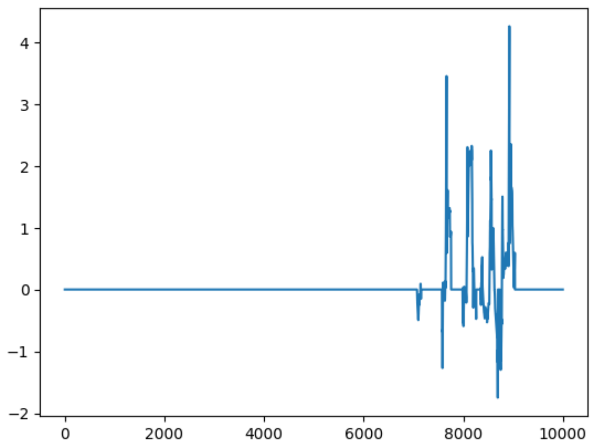
The plots are as follows for the x,y,z force sensors of the left antenna:



(a) Force r.t x



(b) Force r.t y



(c) Force r.t z

Figure 4: Forces felt by the antenna when colliding.

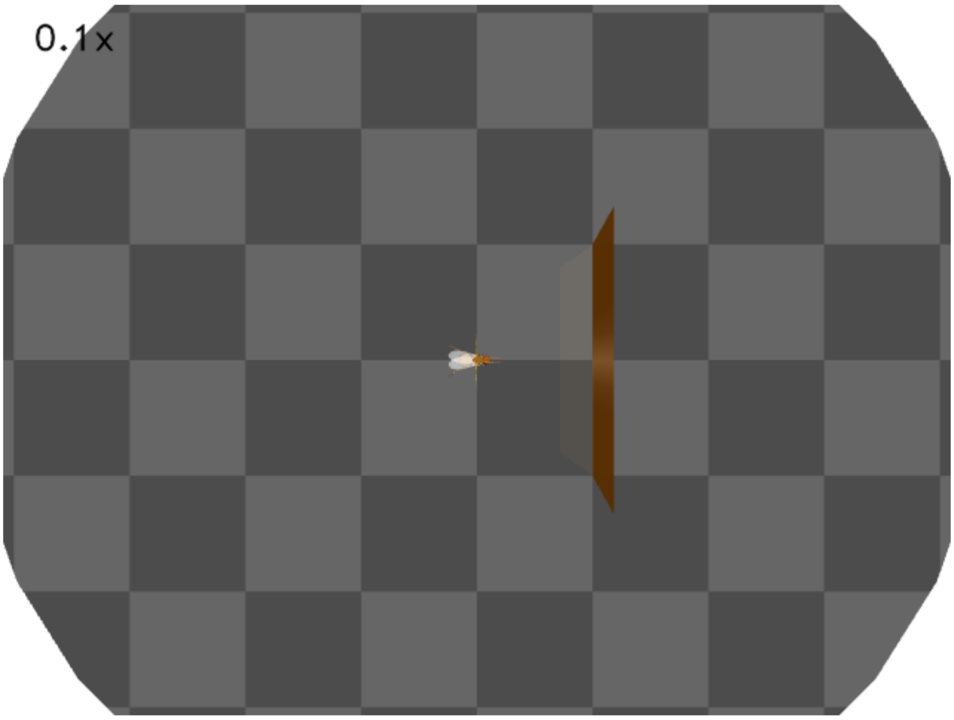


Figure 5: Arena figure

5 Future Work

An interesting next step would be to reverse engineer prerecorded fly movements by tracking their joint positions and replicating these movements for backward walking. Furthermore, integrating the NeuroMechFly (NMF) simulation with a neural model of the fly's connectome could allow us to test the responses of actual descending neurons, creating a highly realistic simulation to validate experimental data. It would also be beneficial to test other types of obstacles and sensory feedback from the antennae to verify the resulting behaviors.

6 Discussion

6.1 Challenges & Issues

During the course of our project, we encountered several significant challenges that impacted our progress.

Firstly, we faced considerable difficulty in creating an inclined plane for the fly. The fly consistently responded unpredictably, sometimes unexpectedly flying away, which was very surprising and led to crashes in our setup.

Due to this issue, we attempted to construct any kind of plane that would not cause problems. Consequently, we ended up creating a plane with an angle greater than 90 degrees, which resulted in the fly's legs touching the plane first instead of its antennae.

After several attempts, we finally succeeded in constructing an inclined plane towards the fly, and strangely enough, it worked. We still do not have a clear explanation for this successful outcome, but it allowed us to proceed with our experiments.

References

[1] Salil S. Bidaye - Christian Machacek - Yang Wu - † Barry J. Dickson†‡. *Neuronal Control of Drosophila Walking Direction*. SCIENCE AAAS, 2014.

[2] Joaõ Campagnolo - Pavan Ramdya - Pascal Fua1† Semih Günel1 - Helge Rhodin1 - Daniel Morales2. *DeepFly3D, a deep learning-based approach for 3D limb and appendage tracking in tethered, adult Drosophila.* eLife, 2019.