Ecole Polytechnique Fédérale de Lausanne

NeuroMechFly Student project

NeuroMechFly 2.0 Navigation: Enabling Spatial Awareness through the Implementation of Bio-Inspired Path Integration Mechanisms

---

Jean Cordonnier, Flore Munier-Jolain

January 5th 2024

Contents

Abstract

This report brings to light the different steps we had to follow to try to enhance the navigational and path integration capabilities of the NeuroMechFly 2.0 virtual fly model. Through this project we used a numerical simulation to underlying mechanisms controlling the navigational and path integration behavior of a fly. This involved implementing new navigation capacities that enabled the fly to move in precise directions relative to its own position, for example to find a food source and go back to its initial point like an ant would. The development of these types of artificial controllers is an essential step in the biorobotics approach of neuroengineering to understand neural circuits generating animals' complex behaviors. Then, these understandings about biological intelligence can be used for the design of complex bio-inspired robots. This multidisciplinary project, combining neuroscience and robotics, allowed us to tackle the following question: How can we design a robust robotic controller allowing our virtual fly to maintain a spatial awareness, a crucial behavioral aspect?

To implement this spatial processing, the first step was to design a task which serves as a baseline to implement our controller and allows us to measure and determine the success of what we implemented. During the first phase of our task the virtual fly has to undertakes a random walk until a certain point. Afterward, the fly has to navigate back to its starting position by precisely following a given heading vector. This vector has to be computed using biologically relevant fly path integration mechanisms applied during its walk.

The primary challenges in this project is to find a biologically relevant way to integrate the fly current heading, without dependence on simulation parameters and, instead, using descending information. To address this problem we first tried to identify the best descending parameters allowing us to extract the fly's heading.We established multiple regressions connecting the change in fly heading with specific descending information that the fly autonomously extracts through sensory input or neural processing. These simple regressions allow the imprecise computation of return heading but encountered difficulties in integrating complex trajectories. While serving as a valuable foundation, these analyses offer an initial stage for the development of more sophisticated and accurate models.Our final model enables the virtual fly to engage in a random walk while simultaneously integrating its current heading to a certain point. This integration allows self-awareness of its orientation, enabling the fly to navigate back to its initial point. However to better fit realistic behaviors, we can refine and elaborate on our task by introducing multiple food sources. This modification requires the fly to remember the locations of these food sources and consistently update this information over the course of the experiment.

1 Introduction

Understanding how brains control behavior remains with lot of unknown. From the function of individual neurons to how animals interact with their surroundings, it's a big challenge. To tackle this, we need simulation frameworks that can explore these complexities. NeuroMechFly 2.0 [1] is one such tool, building on previous models to better understand how the nervous system guides behaviors.

Our objective is to construct a numerical simulation, allowing to investigate the mechanisms controlling the fly's navigation and path integration behaviors. We focused on incorporating advanced navigation skills enabling the virtual fly to move precisely in accordance with its environment.

Generating a random walk is a crutial preliminary step before studying mechanisms controlling the fly navigation. It can be seen as foundational aspect of our project. Then, based on this, the virtual fly needs to be able to integrate its path, enabling self-positioning within its environment, represented here by the arena.

To create a realistic model, we must train an algorithm capable of predicting fly orientation based on descending signals comprehensible to the fly. The selection of this specific signal is a important points of the implementation as it can significantly influence the accuracy of predictions. Representing fly orientation also plays a crucial role, as multiple vectors are available in the simulation to do so, each carrying advantages and disadvantages.This training is realized based on a large data set created

from long simulations of fly random walks. The initial approach involves a straightforward regression, allowing to link relevant fly signals and changes in fly heading. A robust relationship between these factors would enable the prediction of the return heading, guiding the fly back to its initial point after a random walk. However, this approach may encounter limitations due to the inherent noise in simulation signals (trajectory, heading vectors, velocity vectors, etc.) and the non-instantaneous update of fly position following the initiation of descending signals.

All of these encourage us to go further in our way to predict the fly's orientation and to explore new biologically relevant method that could helps us understand better the mechanisme behind the fly's path integration mechanism.

2 State-of-the-art

Understanding animal behavior is crucial in the fields of neuroengineering. It involves unraveling complex mechanisms that have evolved over time, allowing animals to adapt efficiently to their environment. It serves as a rich source of knowledge in neurobiology and provides inspiration for developing efficient robotic systems. Many animal behaviors involve the need for orientation and steering within their environment. This can explain why neural mechanisms underlying navigation capabilities are highly studied in neuroscience and most especially in Drosophila melanogaster due to its genetic simplicity, easy handling, conserved neural circuits, robust behaviors, optical accessibility... All together these studies explore the complexity of multi-sensory processes.

Insects exhibit a remarkable aptitude for navigation, relying on various cues such as the sun, sky, olfactory signals, wind, and self-motion to traverse diverse environments. Recent work by Currier et al. [2] shows that this fundamental behavior is influenced by multiple sensory modalities, including in flies. While the role of vision in fly navigation is well-established, contributing to spatial orientation and directional guidance, there is growing interest in understanding how flies maintain spatial awareness in the absence of visual cues, in dark environments. This study explores how mechanosensation, facilitated by antennal mechanoreceptors detecting wind and sound, and odors detection may play a crucial role in influencing navigation behaviors. While the role of proprioception in navigation behavior is not fully understood, this review [3] outlines that recordings from Drosophila navigation circuits indicate its potential role in keeping the fly on track even without external landmarks. Proprioceptive information, related to the sense of body position and movement, seems to help maintain an internal sense of direction within the central complex when flies navigate in the dark. Some other studies focus more on the potential impact of corollary discharge, a crucial brain function that helps animals differentiate between self-generated and external signals [4], in navigation tasks. Indeed, as show by this research [5], in motor control, corollary discharge affects how sensory information is processed during movement. Specifically, in the fly's visual system, visual neurons use corollary discharge signals to subtractively compute and eliminate components caused by self-motion. This prevents unwanted visuomotor reflexes during flight and stabilizes movements while walking. Ongoing research efforts are directed towards understanding the neural mechanisms that generates these signals to control fly navigation. The NeuroMechFly model can serve as a valuable model for testing and improving hypotheses related to these neural processes.

In this paper from Seelig et al. [6], researchers investigated Drosophila navigation by conducting in vivo experiments utilizing two-photon calcium imaging to gain understanding of neural processing. In the experiments, a Drosophila melanogaster was head-fixed in a virtual reality arena, allowing the measurement of brain activity while walking on a ball. Thanks to this study it has been shown that flies can combine a continuous path integration with potentially intermittent landmark-based orienting to navigate in many different environments. These information are continuously processed to give rise to a dynamic compass-like representation of the fly's orientation. Then it has been shown that a

specific type of neuron present in a Drosophila melanogaster brain region, called ellipsoid body, uses a dynamic bump-like activity to neurally code this animal's orientation. The ring attractor model is generally used as a theoretical framework to explain this fly brain process. Indeed, this proposed ring attractor model, characterized by limited local excitation and flat long-range inhibition, was identified consistent with the observed neural activity pattern in the research conducted by Kim et al. in their paper [7].

Some other researchers focused more on path integration principles and how animals use cues as landmarks for spatial awareness. This paper written by Mitchell et al. [8] investigates how multiple external cues are combined and integrated in order to facilitate orientation behavior. Through experiments conducted on dung beetles, the authors claim that cue integration, weighted sum of vectors, is influenced by cue contrast, such as the gradient in light intensity produced by the sun. This contradicts previous works, which suggested that cue vectors were weighted according to their reliability.

In addition to possessing a sense of direction, the fly also has the ability to remember spatial information. In this paper [9] Behbahani, Amir H et al. discovered that flies uses path integration to memorize the location of a food site, a crucial skill for their survival. In this study, researchers used a kernel density estimate (KDE) based on back-and-forth runs to evaluate spatial memory. Once a food source is located, the fly is able to self localized relative to the food source with path integration using different cues such as the odometry and its compass navigation. When the fly encounters two food sources, instead of memorizing all locations, the fly adapts its integrator, centering it between the two food sites. The study proposed the central complex as a candidate locus for these computations.

The study of animal behavior, particularly in simpler organisms like Drosophila, has provided valuable insights for neuroengineering and robotics. Research on fly navigation mechanisms, such as Seelig et al.'s work using advanced imaging, reveals the brain's dynamic processes. Mitchell et al.'s findings challenge conventional ideas about cue integration in dung beetles. Additionally, Behbahani et al.'s study on fly spatial memory demonstrates how the fly can remember its position relative to a memorized zone. All these studies will help us in improving the navigational and path integration features of our virtual fly model. Our main challenge will be to integrate all these biological mechanisms in the implementation of our virtual controllers while addressing the existing limitations of the model.

3 Methods

3.1 Turning controller

To implement these new navigational capabilities, we based our implementation on the pre-programmed turning controller described in the paper presenting NeuroMechFly 2.0 [1]. This pre-programmed Hybrid Turning NMF controller is based on a CPG-based and sensory feedback-based robust hybrid controller allows the fly to turn thanks to two coefficients. This 2D representation $[\delta_{\mathrm{L}},\delta_{\mathrm{R}}]$ of descending signals allows to modulate the amplitude and direction of leg CPGs on each side of the body.

3.2 Random walk

The first compulsory element needed to follow our implementation was to generate a random walk allowing the fly to evolves randomly in it environment seeking a food source for example. As described by Yang et al. in their paper [10], steering and turning in Drosophila involve an asymmetric tripod walk characterized by multiple leg gestures. This includes increasing the stride length of the three legs on the outside of the turn and decreasing the one for the inside legs. Our random walk algorithm is derived from this principle.

We based our implementation on four different normal distributions, presented bellow. We choose four different distribution in order to fit four different leg movements needed for the fly to freely evolves in the environment by turning, going straight forward ... Here the four different Normal distributions:

- $\mathcal{N}(1.25, 0.3)$ defined to generate large amplitude leg movement needed for external legs when turning
- $\mathcal{N}(1.0, 0.3)$ defined to generate normal amplitude leg movements needed for strait forward walk
- $\mathcal{N}(0.40, 0.3)$ defined to generate small amplitude leg movements needed for internal legs when turning
- $\mathcal{N}(-1.25, 0.3)$ defined to generate backward leg movements in the internal legs required for executing large turns.

We developed the following random walk algorithm with these following steps:

Step 1: The algorithm randomly chooses a value for δ_{L} according to these four different normal distributions and their associated probabilities.

Step 2: Then the second coefficient is computed "based on the first one". This means that, normal distribution probabilities for this second coefficient are updated in a manner to increase the probability of turnings. With a completely random choice we would have almost all the time a straight forward walk.

- If $\delta_{\mathrm{L}} \geq 1.0$, we used the following probabilities, increasing the fly probability to turn right:
  - $\mathrm{P}(\mathcal{N}(1.0, 0.3)) = 0.1$
  - $\mathrm{P}(\mathcal{N}(1.25, 0.3)) = 0.1$
  - $\mathrm{P}(\mathcal{N}(0.4, 0.3)) = 0.5$
  - $\mathrm{P}(\mathcal{N}(-1.25, 0.3)) = 0.3$
- If $0.4 < \delta_{\mathrm{L}} < 1.0$, we used the following probabilities, increasing the fly probability to go straight:
  - $\mathrm{P}(\mathcal{N}(1.0, 0.3)) = 0.4$
  - $\mathrm{P}(\mathcal{N}(1.25, 0.3)) = 0.2$
  - $\mathrm{P}(\mathcal{N}(0.4, 0.3)) = 0.2$
  - $\mathrm{P}(\mathcal{N}(-1.25, 0.3)) = 0.2$
- If $-1.0 < \delta_{\mathrm{L}} < 0.4$, we used the following probabilities, increasing the fly probability to turn left/go straight:
  - $\mathrm{P}(\mathcal{N}(1.0, 0.3)) = 0.05$
  - $\mathrm{P}(\mathcal{N}(1.25, 0.3)) = 0.9$
  - $\mathrm{P}(\mathcal{N}(0.4, 0.3)) = 0.05$
  - $\mathrm{P}(\mathcal{N}(-1.25, 0.3)) = 0.0$
- If $\delta_{\mathrm{L}} \leq -1.0$, we used the following probabilities, increasing the fly probability to turn left:
  - $\mathrm{P}(\mathcal{N}(1.0, 0.3)) = 0.1$

- $P(\mathcal{N}(1.25, 0.3)) = 0.3$
- $P(\mathcal{N}(0.4, 0.3)) = 0.5$
- $P(\mathcal{N}(-1.25, 0.3)) = 0.1$

Step 3: Then, the algorithm compute the absolute difference between these two coefficients to assess the "turning state" of the fly. If a significant difference exceeding a predefined threshold of 0.65 is observed, we assume that this coefficient combination corresponds to a turn. In the other hand, if the absolute difference is below the threshold, the "turn" state is set to False.

Step 4: Finally, a counting step variable is established, regulating the duration for which this combination of coefficients is applied. Indeed, in order to get a smooth and stable walk, facilitating directional changes for the fly, it requires to maintain this coefficient combination constant for a significant number of steps within the range of 1000 to 10000 steps. To define this variable, we decided to use a gamma distribution with distinct parameters based on whether the coefficient combination is associated with a "turning state" being true or not.

- If turning state is True: $\Gamma(140, 65)$
- If turning state is False: $\Gamma(50, 30)$

Step 5: The controller receives δ_L and δ_R coefficients for a specific number of simulation steps, defined by the counting step variable, before the algorithm resets and initiates again.

This random walk implementation allows the fly to change its trajectory through two different mechanisms. The first mechanism facilitates small and smooth turns, using minimal amplitudes for internal legs and larger ones for external legs. The second mechanism enables the fly to execute complete turns, leading to a significant trajectory change by using positive descending drive for external legs and negative drive for internal legs. The detailed explanation of this turning behavior is provided in the following section.

3.3 Turning behavior

In the specified task, the virtual fly is required to execute an almost full turn by the end of its exploratory walk before returning to its starting point and also during its random walk. The goal is to closely replicate the fly turning behavior to simulate biologically relevant complete turns as described in H.Yang et al. paper [10] and as observed with the fly located at the far right of the Appendix [Video 1]. During a "pivot" steering maneuver we still observe a tripod rhythmic limb movement pattern in which the internal legs are pushed backward, while the external legs execute a forward swing movement. This turning mechanism is schematized in the Fig 1

Initially, before our exploration of biologically relevant turning behavior, our approach involved inducing smooth turns in the fly. This was done by applying a very small positive descending drive to the internal leg and a larger positive drive to the external leg (e.g., $[\delta_L,\delta_R] = [0.2, 1.2]$). However, as shown in the results section, this method lead to more gradual changes in direction rather than distinct turns. Consequently, we retained these particular descending drive combinations in the random walk algorithm to facilitate smooth directional changes.

Then for our second approach aiming to obtain complete turns, we applied a strategy similar to the one used for inducing smooth turns in the previously explained random walk. However, in this case, we introduced a negative sign for the small internal leg descending drive. To implement this, we assigned the following descending drives: -0.3 for the internal legs and 1.2 for the external legs. The

minus sign allows to get an internal step with a small amplitude backward movement. This approach demonstrates results that are not highly effective.

Since the turning movement pattern is also a coordinated tripod pattern, it can be modeled using a CPG controller. This is why we decided to base our final implementation on the Hybrid Turning NMF controller, using distinct turning coefficients $[\delta_L,\delta_R]$ = [+-1.2, -+1.2]

- **Coefficients signs:** The sign of the descending signal controls the direction of stepping by influencing the sign of the intrinsic frequency of leg CPGs. This explain why, in our turning behavior implementation, the internal legs are associated with a negative descending drive. This allows to mimic the biological legs movement where they are pushed backward.
- **Amplitude:** To observe a rapid and clear turning behavior we need to set our coefficients to large absolute value, while still maintaining biologically relevant leg movements. Furthermore, to keep a coherent synchronization between the left and right legs, we decided to maintain identical absolute values for both coefficients. As a consequence, we have settled on an absolute value of 1.2 for this example but more generally absolute values can be comprised between 1.0 and 1.2.

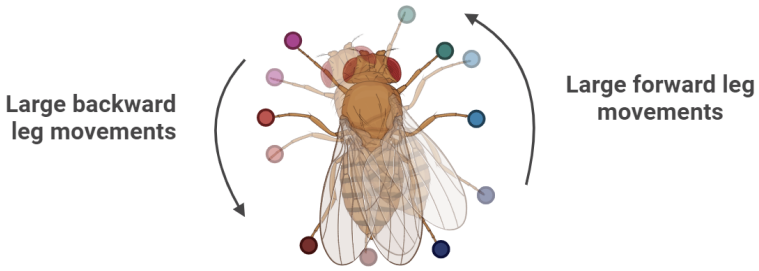


Figure 1: Diagram illustrating the mechanism that governs the rapid and direct turning behavior in flies

To get a closer look to the robustness of our implemented turning behavior, we successively repeated three simulations using the same set of drives: $[\delta_L,\delta_R]$ = [0, 0] for 2'000 steps and $[\delta_L,\delta_R]$ = [-1.2, 1.2] for 10'000 steps. At the beginning of each new simulation, we reset the nmf object without resetting the CPG network, as would occur during the random walk. The resulting trajectories and CPG phases for all legs were then plotted for analysis.

3.4 Path integration groundtruth

Throughout our entire project, we have to deal with headings, vectors representing the fly orientation during the entire simulation. To gain a more in-depth understanding of the simulation parameters at our disposal, we decided to retrace/integrate the fly's trajectory by using these simulation parameters. To do so we integrated these parameters throughout the simulation, with a cumulative sum of their (x,y) values. Then, we plotted the integrated trajectory alongside the ground truth trajectory obtained from the simulation. We realized this experiment by using two different parameters:

- (x,y) values of the fly's orientation pointing toward the direction that the fly is facing (`obs['fly.orientation']`)
- (x,y) values of the fly's velocity (`obs['fly][0][:2]`)

Both of these vectors show limitations associated with the overall simulation implementation. Indeed, orientation alone is not sufficient, this is why we explored alternative solutions for representing the fly's heading.

3.5 PID controller

In order to control the fly direction we designed a simple PID controller that was designed to choose the descending signal according to the goal the fly is supposed to reach.

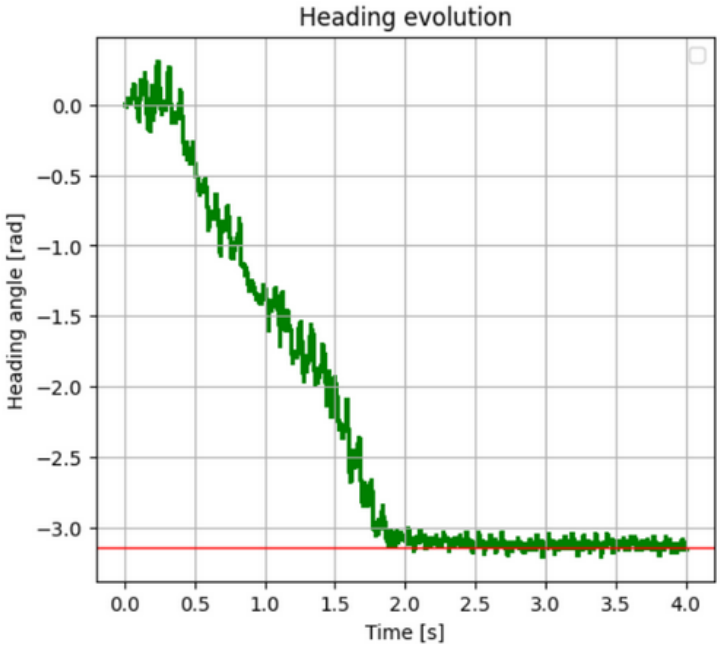


Figure 2: Trajectory of a full turn around monitored by a PID

The controller scales the two descending signals based on the norm and the sign of the goal angle. In the end we only used the proportional part of the controller as having sharper turns was essential for more efficient turnings.

3.6 Heading change regression

During its navigation, the fly must integrate its heading to develop spatial awareness. However, in reality, Drosophila are not simulated object, they have to extract their own heading by using their biological capabilities, including natural sensors and neural processing. As a consequence, our final implemented model needs to be designed to exclusively integrate the fly path by relying on biologically relevant signals. In the case of fruit flies, these signals are descending signals, neural signals originating in higher brain centers and traveling downward through the nervous system to influence motor outputs such as navigation or even proprioceptive signals.

As a first approach, we attempted to train linear regression models to try to establish a robust relationship between information accessible to the fly, such as descending information or proprioceptive data, and a change in heading direction. Developing such a relationship would enable our virtual fly to extrapolate its heading based on these parameters.

3.6.1 Random trajectories data base

To construct these regression models, the initial step is to generate an large dataset derived from multiple walking simulations. We realized ten distinct simulations using our random walk algorithm programmed to generate complex trajectories, with multiple turns and directional changes to enrich our training database. Throughout these simulations, we collected multiple simulation parameters

which would serve as training data for our models. Finally we obtained a large CSV file, containing all the following parameters for each of the final 1,360,000 steps (8 simulations of 150'000 steps and 2 simulations of 80'000 steps):

- Fly's position
- Fly's orientation
- Fly's velocity
- Fly's cumulative velocity
- Drive descending coefficient (2D parameter)
- Contact forces measured at the Tarsus 5
- Leg position measured at the end effectors
- Swing/Stance phase transitions based pre-programmed steps and CPG phases

3.6.2 Sliding window

To create robust and meaningful linear regression models using biologically relevant signals and to handle the slow timescale of turning, we decided to apply a sliding window principle. This approach involves the calculation of the sum or mean of a specific parameter within the window, which is then used to evaluate its relationship with the change in heading that we will describe in the next paragraph. This fixed-size window is moved incrementally to the next step and this incrementally until reaching the end of the simulation.

By computing means or sums over consecutive steps, the model captures the dynamics and trends in "biological parameters", providing a more nuanced understanding of their influence on heading changes over time. Moreover, the sliding window introduces a smoothing effect by aggregating parameter values over a defined window. This helps to reduce noise or fluctuations in the data, contributing to a more stable and representative input for the regression model.

This sliding window approach can be tuned by adjusting an important parameter: the window size. Indeed, the window may significantly influences sensitivity to change. In our case, we will have to choose sufficiently large window size because in our virtual fly simulation, the fly is set with consistent descending drive parameters for several steps (in the thousand's range), and turns are not instantaneous due to the slow timescale of turning. To select the most suitable window size for our different regression tasks, we realized a grid search by computing the Pearson correlation coefficient between the heading change and the predictor variable. Finally, we chose the one associated with the highest correlation. Additionally, we can also vary the number of data points that are either summed or averaged inside a window. To make the computation faster without loosing to much information we decided to sum/average one out of every 10 values within each window.

3.6.3 Heading changes

For our regression task and more generally for any path integration task, it is essential to measure the variation in the fly's orientation across a specified number of steps. To address this, we designed an algorithm allowing to compute the mean heading on the few steps before the interval of interest and the mean heading during this interval. Then, a comparison is possible between these two averages. More precisely, we compared the mean heading computed over the preceding window, concluding at the initiation of the window of interest, with the mean heading computed over the entire window

of interest, as depicted on the Fig 4. This analysis provides valuable insights into the correlation between the parameter and heading changes. As we described in the "path integration ground truth" paragraph, we can use two possible "orientation vectors" to quantify this heading change:

- (x,y) values of the fly's orientation pointing toward the direction that the fly is facing
- (x,y) values of the fly's velocity

To decide between these options, we relied on graphical analysis. Indeed, we generated graphs with the fly's trajectory and the various heading change vectors across multiple windows, each made of 2000 steps. Our selection process involved choosing the parameters that best captured the trajectory (those most tangential to the trajectory).

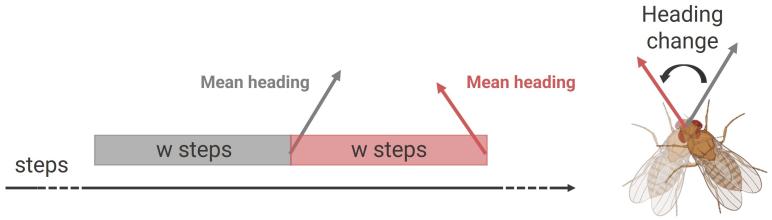


Figure 3: Diagram illustrating the computation of changes in heading when using sliding windows

3.6.4 Drive difference VS heading change

Model fitting

The first parameter selected for generating our regression model for heading change is the descending drive input. This input corresponds to the two "turning coefficients" provided to the controller, enabling the fly to turn.

We computed the averages for individual step differences between the two coefficients using a specified-size sliding window. Additionally, we determined the sum of each coefficient using the same sliding window. Then we moved the window to the next step forward, recomputed the averages and the sum, and repeated this process in an iterative manner until reaching the end of our simulation data. In parallel we computed the corresponding heading change for all these windows, as outlined in the "Heading changes" section.

We then generated two types of plots. The first provides insights into the overall global pattern connecting heading change and driving coefficients. It displays δ_L sums on the x-axis and δ_R sums on the y-axis. For each combination, we represented the corresponding heading change using a color gradient. The second plot is more useful for visually assessing the possible regression between heading changes and descending drive, as it represents the mean difference between δ_L and δ_R coefficients (over our sliding windows) on the x axis and the corresponding heading change on the y axis. This plot enables us to overlay a regression line for further analysis.

Predictions using only predicted vectors

Finally, we used the established regression model to predict angles associated with heading changes for each 4,000-step sliding window within a test trajectory. Indeed, we realized a tenfold iteration in which we used data from 9 trajectories to "fit" the linear regression. Then, we assessed the model's performance on the remaining test trajectory, finally giving predictions for 10 simulations. Notably, two of these simulations were shorter, comprising only 80,000 steps each.

For predicting the heading vector, we initiated the process with the first heading vector matching the groundtruth (mean velocity vector). Then, at intervals of 4,000 steps, we used the predicted angles corresponding to these steps to calculate a new heading vector based on the previously predicted one. Next this predicted vector is normalized by being multiplied by the norm of the corresponding groundtruth mean velocity vector. This normalization assumes that the fly can perceive and integrate its speed magnitude. While the specific neural circuits enabling the fly to sense its speed haven't been clearly defined, this paper [11] provides evidence that flies can adjust their walking speed using an algorithm tuned to visual motion speed. This suggests the fly can regulate its speed, implying the potential for processing speed-related information. This sequential procedure continued, advancing by 4,000 steps each time, until reaching the end of the simulation. It is important to note a limitation of this approach: predictions must conclude at the last step divisible by 4,000. This constraint arises from the fact that our model link descending drive differences with heading changes obtained after 4,000 steps. For instance, after 2,000 steps, the same descending drive values would be linked to a smaller heading change, introducing a limitation in the predictive accuracy. Consequently, there may be trajectories where the exact end of the simulation is not reached, resulting in a potential maximum deviation of 0.4 seconds.

To quantify the errors associated with these predictions, at each 4,000-step prediction point, we calculated the "predicted" return heading vector by cumulatively summing all the previously predicted vectors. Then, we compared this cumulative vector with the actual return heading vector connecting this point to the initial trajectory point. This approach enables the derivation of an error angle, measured in degrees, every 4,000 steps, providing a quantitative assessment of the prediction accuracy for our task which can be represented on a graph.

Predictions using predicted angles and groundtruth headings

To try to get a better prediction accuracy, we explored an alternative approach by predicting heading vectors using predicted angles. However, instead of using the predicted previous vector to compute a new one, we used the groundtruth heading vectors as the previous vector. This aimed to mitigate error transmission throughout the trajectory. While this method is not entirely biologically relevant, it provides a more accurate assessment of the predictive model's performance in angle computation.

3.6.5 Step length VS heading change

Model fitting

In this second approach implemented to predict changes in direction, we applied proprioception principles. Given that the fly's nervous system can interpret proprioceptive signals, we decided to use the position of the legs within the fly's reference frame to estimate vectors representing a change in heading.

In our predictive process, the initial step is to transform the simulated leg positions, which we used for regression extraction, from the arena reference frame to the fly's body reference frame. We only retained the x and y coordinates of the legs, neglecting the z-coordinate as it has no relevance for our next predictions. Then, we used the acquired swing/stance indices loaded from the simulation, extracted from the states of CPGs. This allowed us to selectively retain leg position both before and after each swing phase, facilitating the extraction of stride length information for each leg.

Finally, for each leg we computed two different features, that we will used for the predictive model:

- The average y distance for each stride length calculated using the same sliding window as employed for the drive difference.
- The mean y position calculated across all time steps, not limited to just before and after the swing, using the same sliding window as applied to the drive difference.

Similar to the first predictive model, we calculated in parallel the corresponding heading change for all these windows. Finally, a linear regression was used using these twelve features to establish an association with the corresponding heading change.

Predictions

Regarding the concrete aspect of predicting precise heading vectors, we used the identical method described in the "Drive difference VS heading change" section.

3.6.6 Models comparisons

Finally in order to statistically compare results obtained using these two models, we realized a paired-sample t-test. This aimed to determine whether there is a significant difference between the means of return heading errors computed using descending drives and using proprioceptive signals related to step length. We realized two different comparisons:

- descending drive method vs. proprioceptive method with predictions using only predicted vectors
- descending drive method vs. proprioceptive method with predictions using predicted angles and groundtruth headings

3.7 Deep learning approach

We trained a LSTM-based network to predict the difference in position in the fly's frame. To train it we used the data-set described in 3.6.1. The network takes as input the descending signal over the last 3000 steps and output the difference in position in the fly's frame for x and y (we trained x and y separately) from the position 3000 steps ago. The architecture of the network is the following: We did not complicated it too much as we wanted to avoid over-fitting. One of the main source of

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 3000, 2)]	0
lstm (LSTM)	(None, 16)	1216
dense (Dense)	(None, 8)	136
dense_1 (Dense)	(None, 1)	9

Figure 4: Architecture of our LSTM network

error was the influence of the change in heading over the the frames transformation despite having a moving average. It resulted in the lateral predictions being less precise than the medial's ones. We did not output any sequence resolution for the LSTM as it never converged toward a solution, since the complexity was too high regarding the number of inputs.

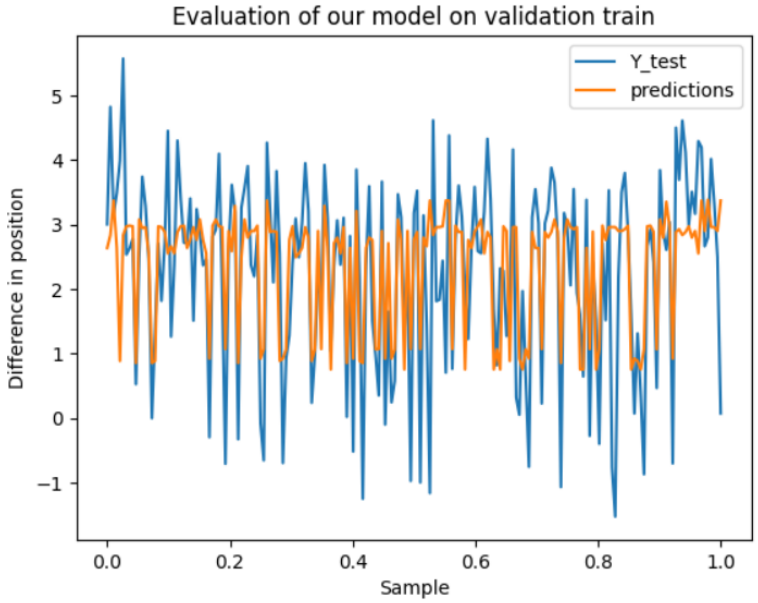


Figure 5: Evaluation of our LSTM network on our validation data. We can already sees that the network struggles to predcit the maximas and minimas with efficiency

4 Results

4.1 Random walk

On this Fig 6 , we can observe different trajectories created by our random walk algorithm. We notice many full turns associated with winding routes. We intentionally programmed the algorithm this way to have a complete training dataset covering all possible fly trajectories for our ongoing analysis. The videos corresponding to these three simulations can be found in the Appendix: [Video 2], [Video 3], and [Video 4].

This random walk algorithm is flexible and can be adjusted to the desired type of "walk." We can easily adjust these characteristics (turns amplitudes, turns frequencies ...) by manipulating the parameters of the various normal distributions and their corresponding probabilities.

Figure 6: Trajectories obtained with the random walk algorithm

4.2 Turning behavior

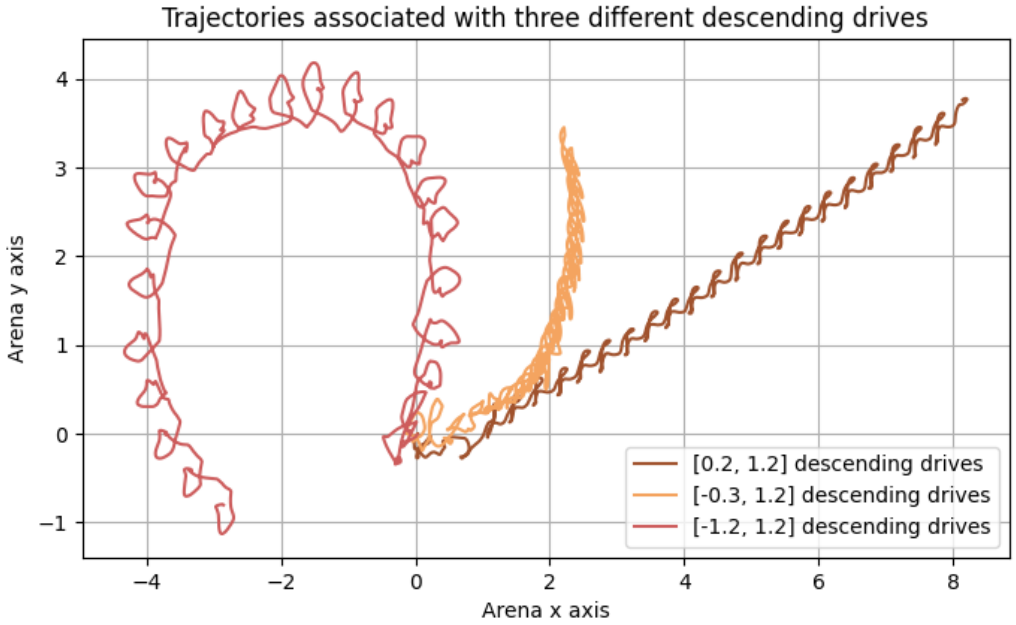


Figure 7: Trajectories obtained after the simulation of 20'000 steps using three turning coefficient combinations: [0.2, 1.2], [-0.3, 1.2] and [-1.2, 1.2]

As mentioned in the method section, the descending drives of [0.2, 1.2] induce a change in direction but not a real turn. Indeed, the fly does not proceed in a completely straight path (aligned

with the spawn direction) but rather transitions smoothly to the left, as observed in the Fig 7 and in [Video 5] of the Appendix, instead of executing a complete turn.

For the turning coefficients [-0.3, 1.2], the simulated movement demonstrates a relatively large and unconstrained turn. After 20,000 steps, the change in direction approaches approximately 90° as observed in the Fig 7 and in [Video 6] of the Appendix.

For the coefficient combination [-1.2, 1.2], the simulated turns executes a more tightly controlled and efficient turn. After 20,000 steps, the change in direction nearly reaches 270° (Fig 7). However, the trajectory corresponding to the [-1.2, 1.2] parameter range exhibits noticeable noise, reflecting an unstable gait characterized by alternating right and left movements for balance. The perceived efficiency of this second approach is confirmed by the accompanying [Video 7] of the Appendix and aligns more closely with the steering pattern described in the paper by Yang et al. [10].

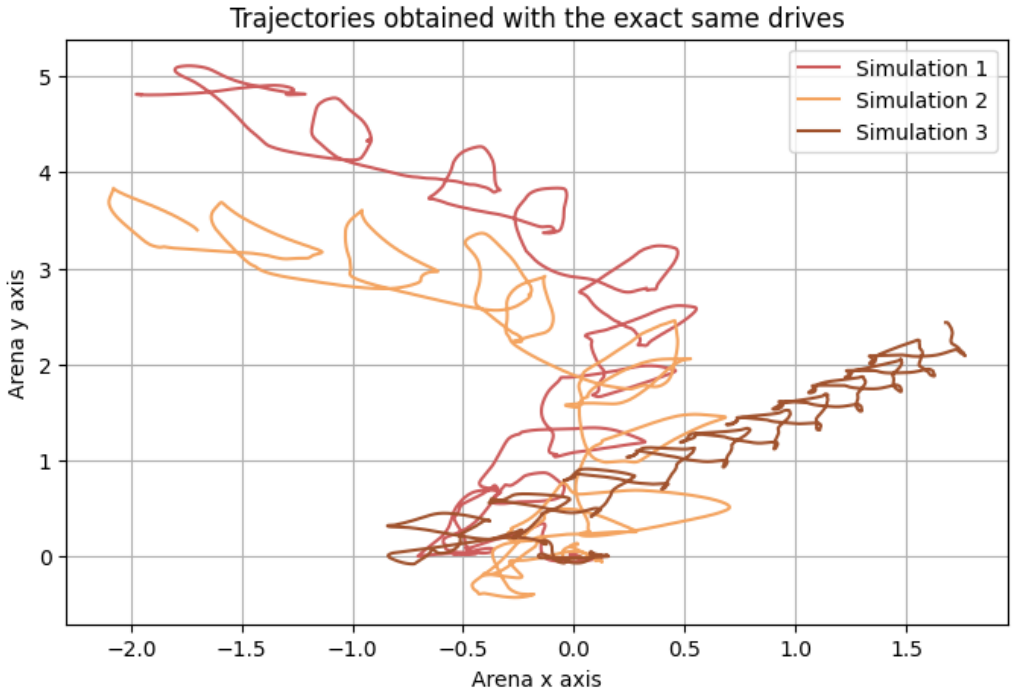


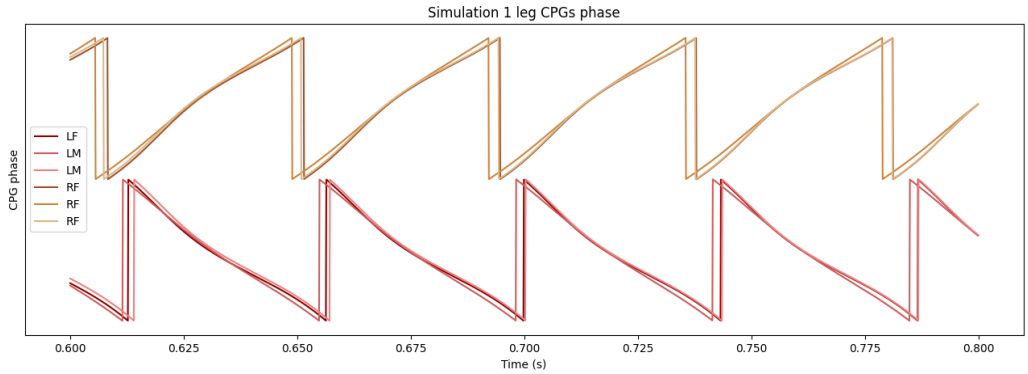
Figure 8: Trajectories of the three consecutive and identical simulations (δ_L,δ_R] = [0, 0] for 2'000 steps and [δ_L,δ_R] = [-1.2, 1.2] for 10'000 steps)

Fig 8 illustrates that with the exact same descending drives given to the simulation model, we can obtain different outcomes. Simulation 3 shows a slight deviation to the left, in contrast to the other two simulations that display pronounced curved trajectories associated with a left turn. Upon examination of Fig 9 and CPG phases, what differs in Simulation 3 is the presence of nearly perfect in-phase CPGs among legs on the same side and between the two sides. A more detailed observation of the events can be made with [Video 8] in the Appendix.

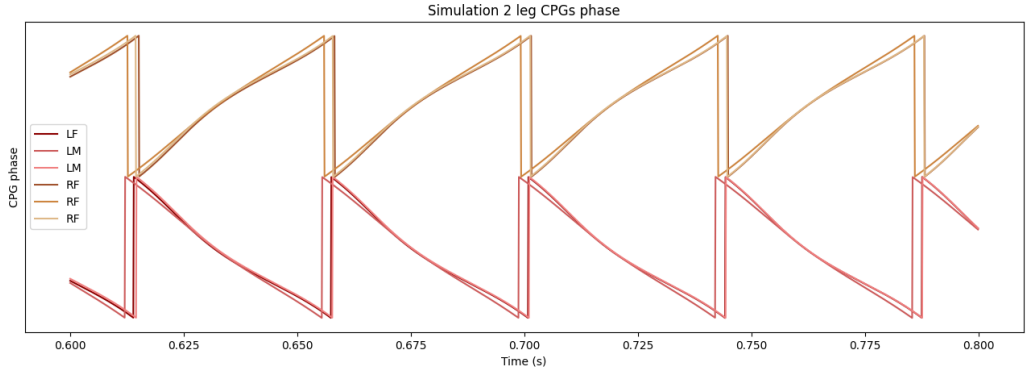
4.3 Path integration

On this Fig 10 we can observe the trajectories retraced by using orientation vector integration and velocity vector integration.

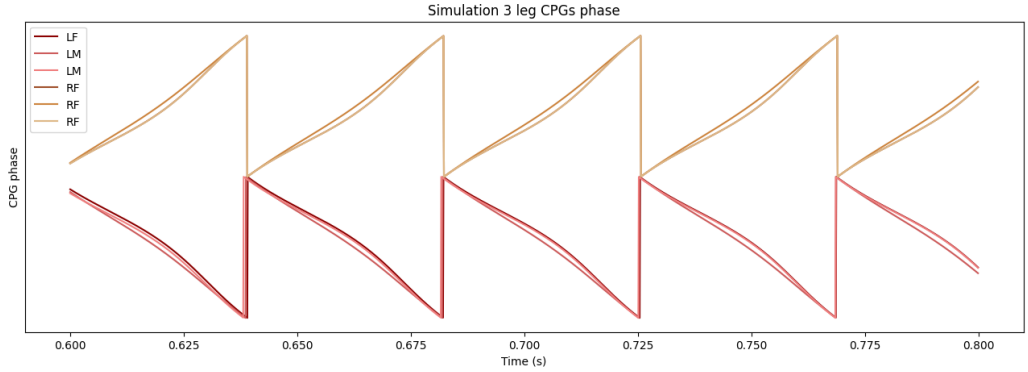
In the left graph, we can observe the two types of simulation vectors of interest. The orientation vectors appear to align more closely with the general trajectory, even if we can observe here a noticeable tendency for them to deviate outward from the trajectory. This observation underscores the limited



(a) Simulation 1



(b) Simulation 2



(c) Simulation 3

Figure 9: CPG phases for all 6 legs during three consecutive and identical simulations (δ_L,δ_R] = [0, 0] for 2'000 steps and [δ_L,δ_R] = [-1.2, 1.2] for 10'000 steps). The plotted data is zoomed and presented within the interval of 6'000 to 8'000 steps.

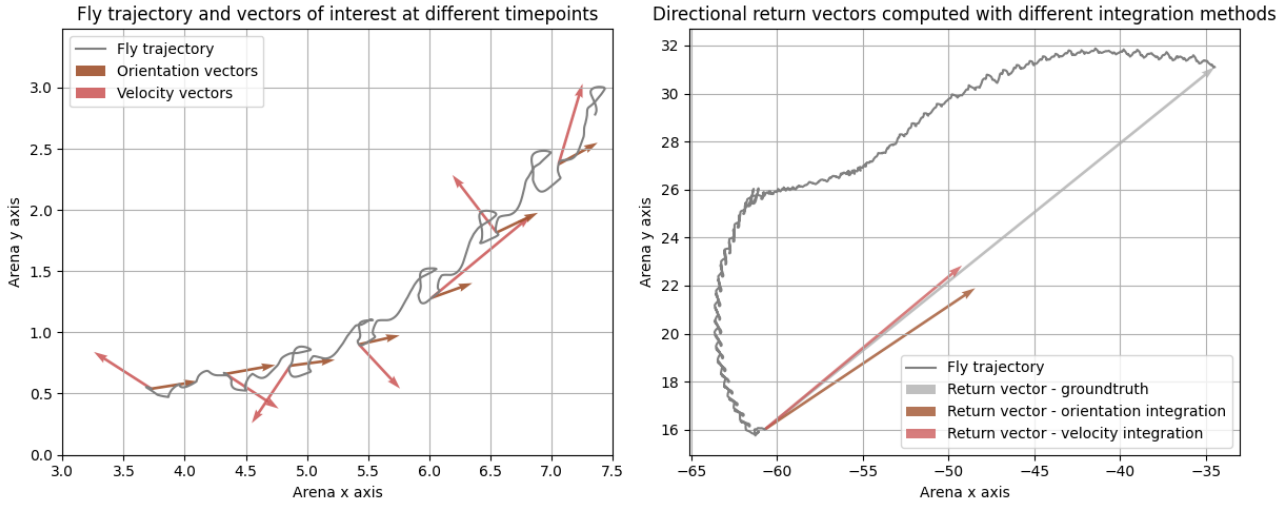


Figure 10: (left) velocity vectors and orientation vectors extracted from the simulation over a segment of the trajectory (3'000 steps). Both velocity and orientation vectors were plotted at intervals of 450 steps. (right) fly trajectory of 17'000 steps with some computed return vectors. The ground-truth vector simply connects the last point of the trajectory to the initial point. The velocity return vector corresponds to the negative (x, y) values obtained from the cumulative sum of velocity components throughout the trajectory. Similarly, the orientation return vector corresponds to the negative (x, y) values derived from the cumulative sum of orientation components over the trajectory.

precision of these vectors in accurately representing the fly's specific heading. Regarding the velocity vectors, we can observe that they are "very noisy" and non-constant. These vectors precisely follow the noisy trajectory and the rhythmic and balanced walk of the fly at every time step.

The direct correlation between velocity vectors and trajectory explains why the cumulative sum of velocity vectors over the entire trajectory results in a quasi exact return vector to the initial point, as depicted in the graph on the right. On this same plot, we can observe that the cumulative sum of orientation vectors over the all trajectory gives a less accurate return vector to the initial point.

4.4 PID controller

- Graph showing the difference between the goal heading and the real fly heading
- Quantify the controller accuracy

4.5 Heading change regression

4.5.1 Heading changes

In Fig 11, the mean orientation vectors and velocity vectors are depicted in dark, on the left and right sections respectively, and calculated over the preceding 2,000 steps. Additionally, we plotted in a lighter shade the vector corresponding to the mean of the previous window, enabling the observation of the heading change associated with the most recent 2,000 steps.

When dealing with mean orientation vectors and velocity vectors, we faced a similar challenge as discussed in the "Path integration" section. Specifically, during smooth trajectories, both orientation and velocity vectors are tangential to the trajectory, with orientation appearing even more precise. However, within turns, velocity vectors exhibit greater tangential alignment to the trajectory in comparison to orientation vectors, which tend to point outside the trajectory.

For both vectors, we determined the return heading vector (in dashed line) by cumulatively summing these mean vectors. The "velocity" return vector closely aligns with the ground truth return vector, while the alignment of the "orientation" return vector is less precise. We can also notice that using the cumulative sum of mean vectors for computing the return heading, instead of the cumulative sum of all vectors, does not significantly affect the observed return heading vector in Fig 10.

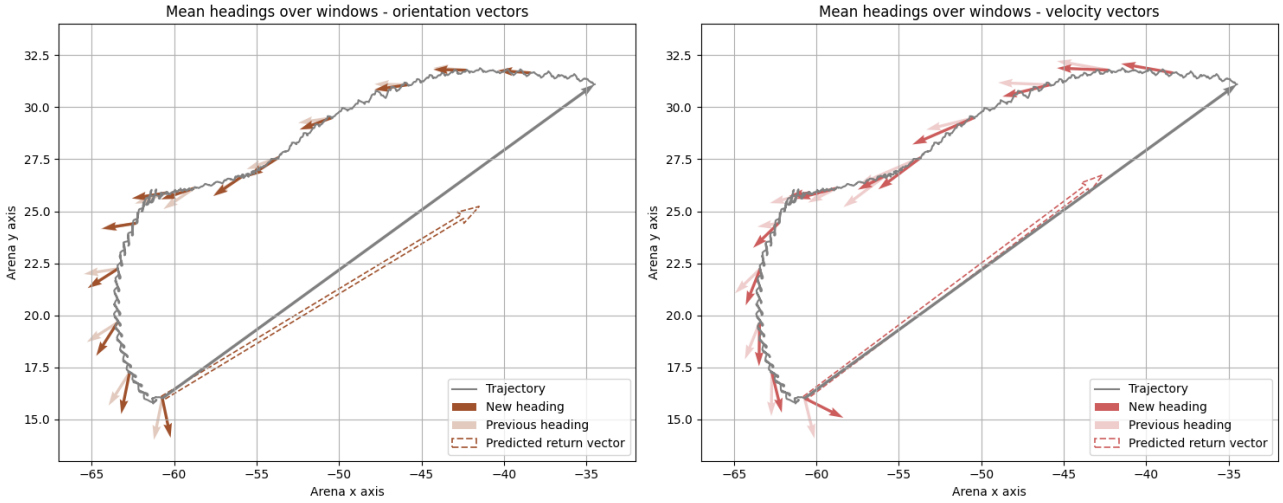


Figure 11: Mean headings were calculated over windows of 2,000 steps. The resulting new heading vectors, in dark, is compared to preceding one, in light, to observe changes in heading. This computation was conducted using both orientation vectors (left) and velocity vectors (right). Predicted return vectors were computed using a cumulative sum of these mean vectors and compared to the groundtruth.

4.5.2 Drive difference VS heading change

Model fitting

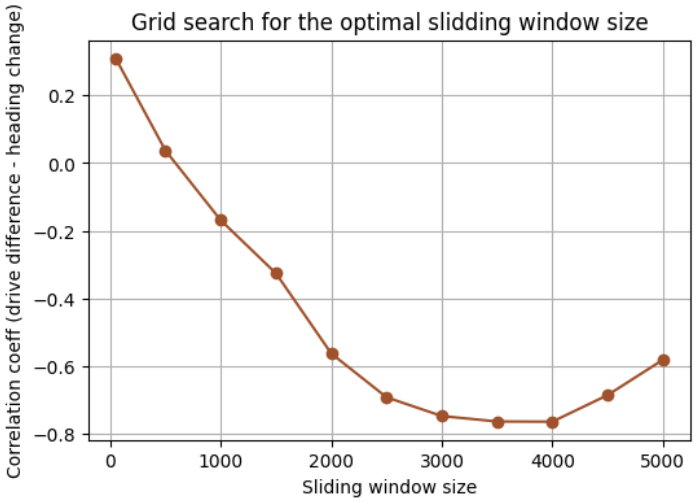


Figure 12: Grid search calculating the Pearson correlation between the mean difference between left and right drive coefficients across windows and the heading change in degrees before and after each window. The correlation coefficients were computed for 11 window sizes using a single simulation of 150'000 steps and are presented in the graph.

For this regression task, the first step is to identify the most effective window size. Fig 21 illustrates the grid search aimed at determining the optimal parameter size. From this search we want to maximized the absolute value of the Pearson correlation coefficient between the drive difference and the computed heading changes over sliding windows. The curve shows a rapid decrease until a window size of 3000-4000, reaching a plateau before beginning to increase again. Based on this plot, we decided to use sliding windows of size 4000.

In Fig 13, the left subplot shows, on the x and y axes, the sum of left and right drives over a 4000-step window. The colormap represents the corresponding heading change angle for these drive combinations, using a diverging color gradient: grayscale indicates positive angles (left turn), and reddish tones indicate negative angles (right turn). We can clearly distinguish different zones, with the upper left corner predominantly colored in grey, the lower right corner mainly in red, and the upper right section displaying a lighter shade. Despite some observable reddish and grey lines in the middle upper and lower parts, representing inconsistent walk patterns, these zones demonstrate the coherence of our data with the model observations: larger left drives, coupled with negative or small right drives, result in right turns, and vice versa. This "pseudo" segmented plot indicates the potential for developing a model that uses drives to predict heading changes.

The next step was to fit a linear regression for this predictive task. In the right part of the Fig 13 we can observe a scatter cloud, where each point represents a mean drive difference calculated over a 4000-step window, corresponding to a heading change on the y-axis. Despite the presence of noticeable noise and several outliers in the scatter cloud, a linear pattern can be distinguished, intersecting approximately at the point (0,0). Negative differences between Left and Right drives are associated with positive directional changes (trigonometric angles), while positive differences correspond to negative angles, representing heading changes (anti-trigonometric angles). To fit to this linear pattern, we applied a linear regression. This regression enables us to correlate a heading change with each descending drive difference for our regression task.

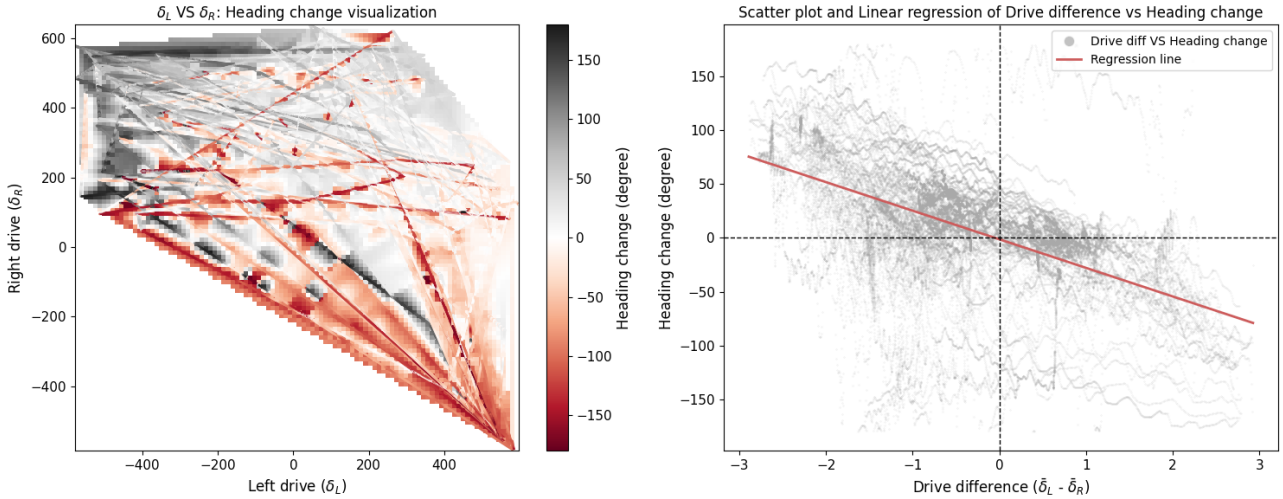


Figure 13: (left) the left subplot shows the sum of left and right drives across a 4000-step window on the x and y axes associated with a colormap indicating the heading change angle associated to each drive combination. (right) this right subplot shows a scatter plot representing the drive difference versus the associated heading change, fitted with a linear regression.

Predictions using only predicted vectors

In Fig 14 (top), quantifying the errors linked to these predictions, it is apparent that predicted return vectors exhibit significant divergence and considerable variation throughout the trajectories in many simulations. Although some simulation lines stay within a reasonable range of -50° to 50°, others diverge, reaching values larger than -180° (we choose a modulo 2π instead of π to better align with the realistic context of the situation). On average, however, the resulting return vectors tend to remain within a reasonable range around 0°.

To examine the cause of divergent predictions, let's focus on Simulation 2 in Fig 15a. We can observe that the error angle drops at the second prediction point to approximately -100°. Upon closer observation of the initial steps through the zoomed view (middle), a sharp turn is apparent. The predictive model struggles to handle this sharp turn, as the subsequent heading direction "fails to follow this turn" accurately. The problem arises from the fact that, with this method, the following heading is predicted based on an angle reported to the inaccurately predicted vector. Consequently, the error propagates along the trajectory, requiring time to diminish, as illustrated in the error angle plot. It becomes apparent that the error decreases before encountering another sharp turn where it drops again.

In Fig 15b, we present a simulation characterized by a smoother trajectory, without sharp turns. The error angle, depicted in the figure, consistently stays within a reasonable range, with a minor drop reaching -50° due to an almost complete turn. The final return vector could drive the fly in a coherent orientation to find its initial point.

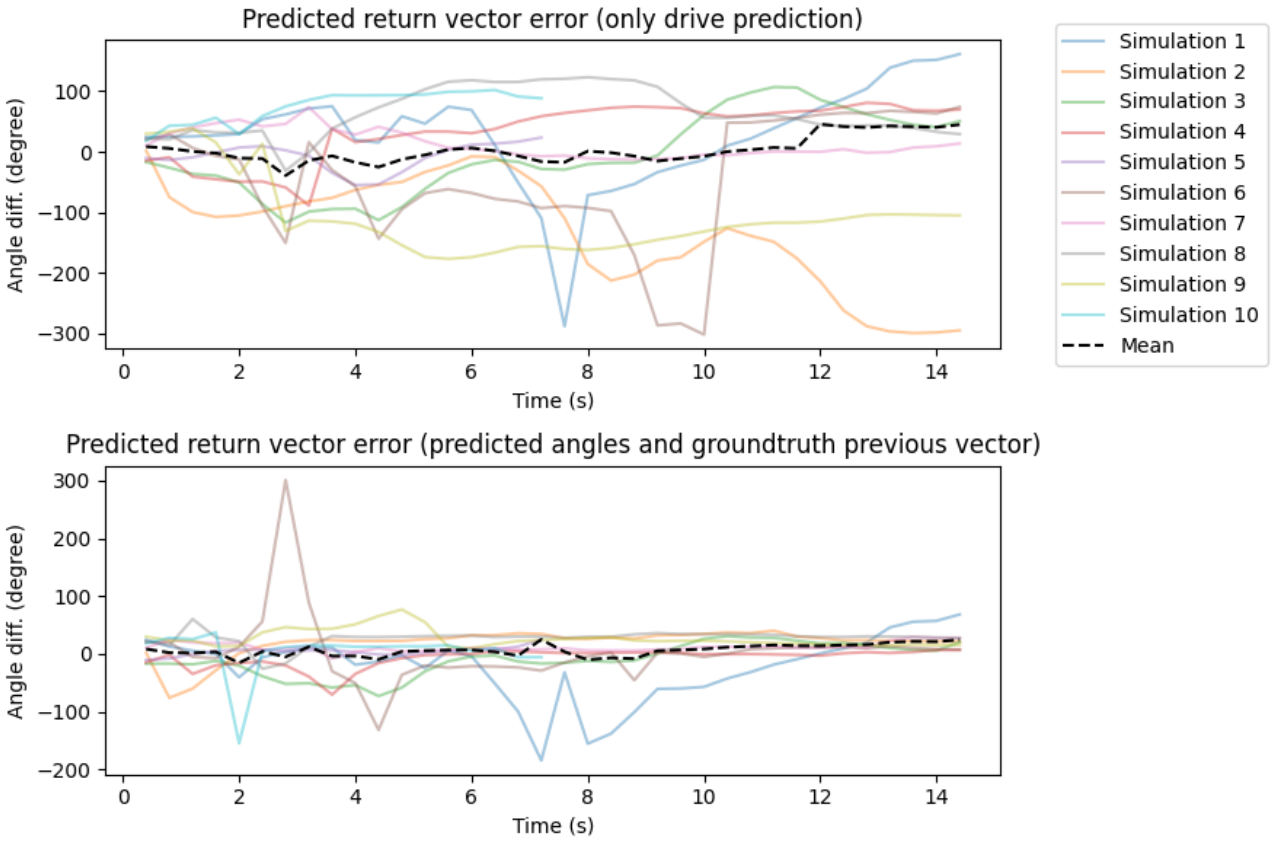


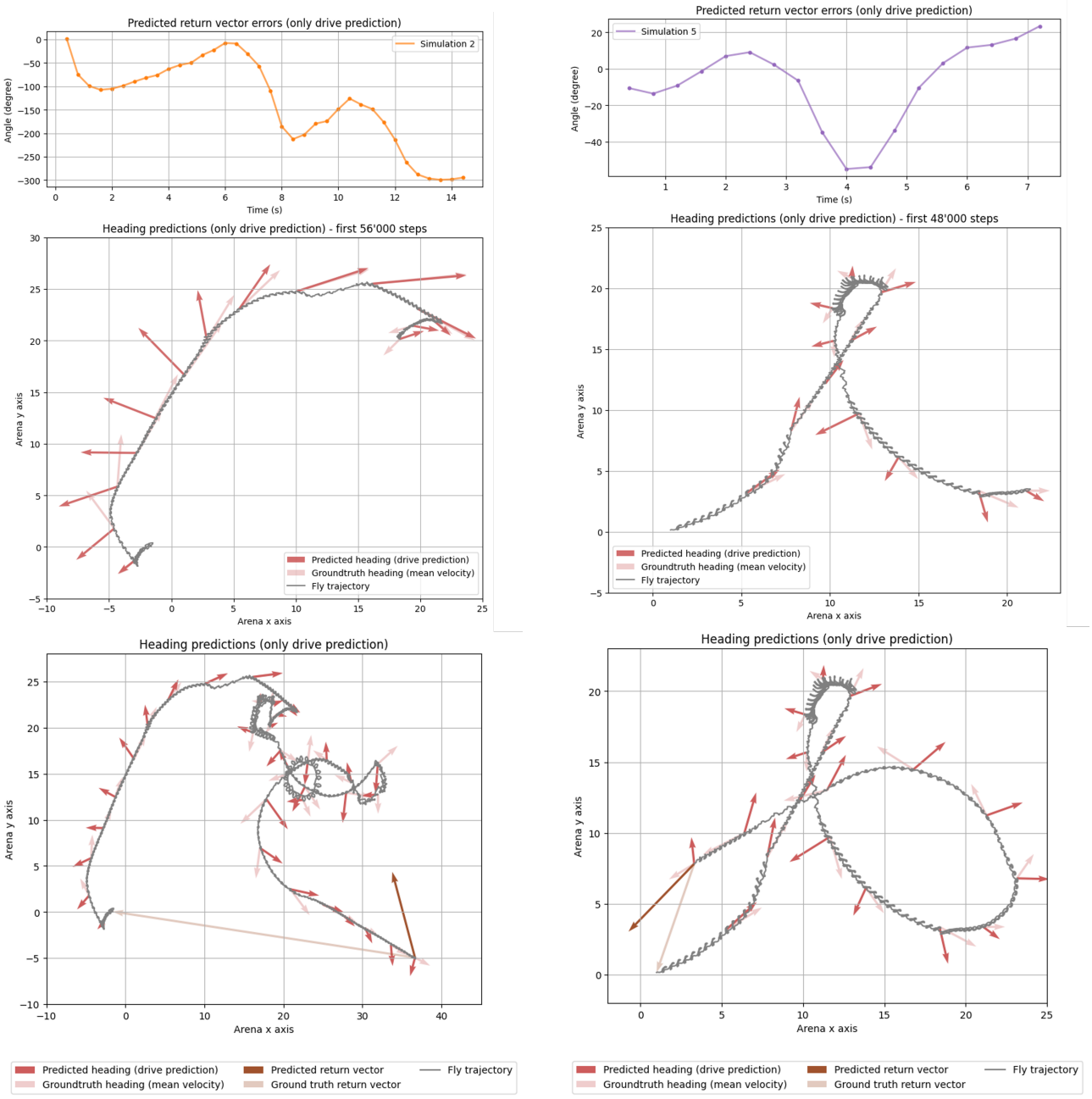
Figure 14: (top): Angle differences, computed at 4,000-step intervals, between the "predicted" return vector obtained from heading vectors predicted (using predicted angles and previously predicted vectors) and the groundtruth return vector across 10 test simulations. The average angle difference is visually presented as a dashed line. (bottom): Same angle differences, but the "predicted" return vector is determined using predicted heading vectors based on predicted angles and the groundtruth previous angle.

Predictions using predicted angles and groundtruth headings

The issue identified in Fig 15a drived us to adopt the second method described in the method section. This approach involves using the groundtruth heading to apply the predicted angle and obtain the predicted heading vector. In the Fig 14 (bottom), we clearly observe better results with less variability and errors remaining closer to 0° compared to the Fig 14 (top). Errors in two simulations (6 and 1) still briefly deviate but are promptly rectified.

In the Fig 16a, we examine the events of simulation 6. At the beginning of the simulation (middle), we can observe a complex trajectory, marked by multiple turns in different directions, corresponding to the divergent zone in the angle error plot (top). This complexity may explain the challenges faced by the predictive model in accurately predicting angles, even with the "correct previous vector" correction. Notably, towards the end of the trajectory, we observe smoother curves associated with precise return vectors (top), as that errors from the initial stages did not propagate throughout the remainder of the trajectory with this method.

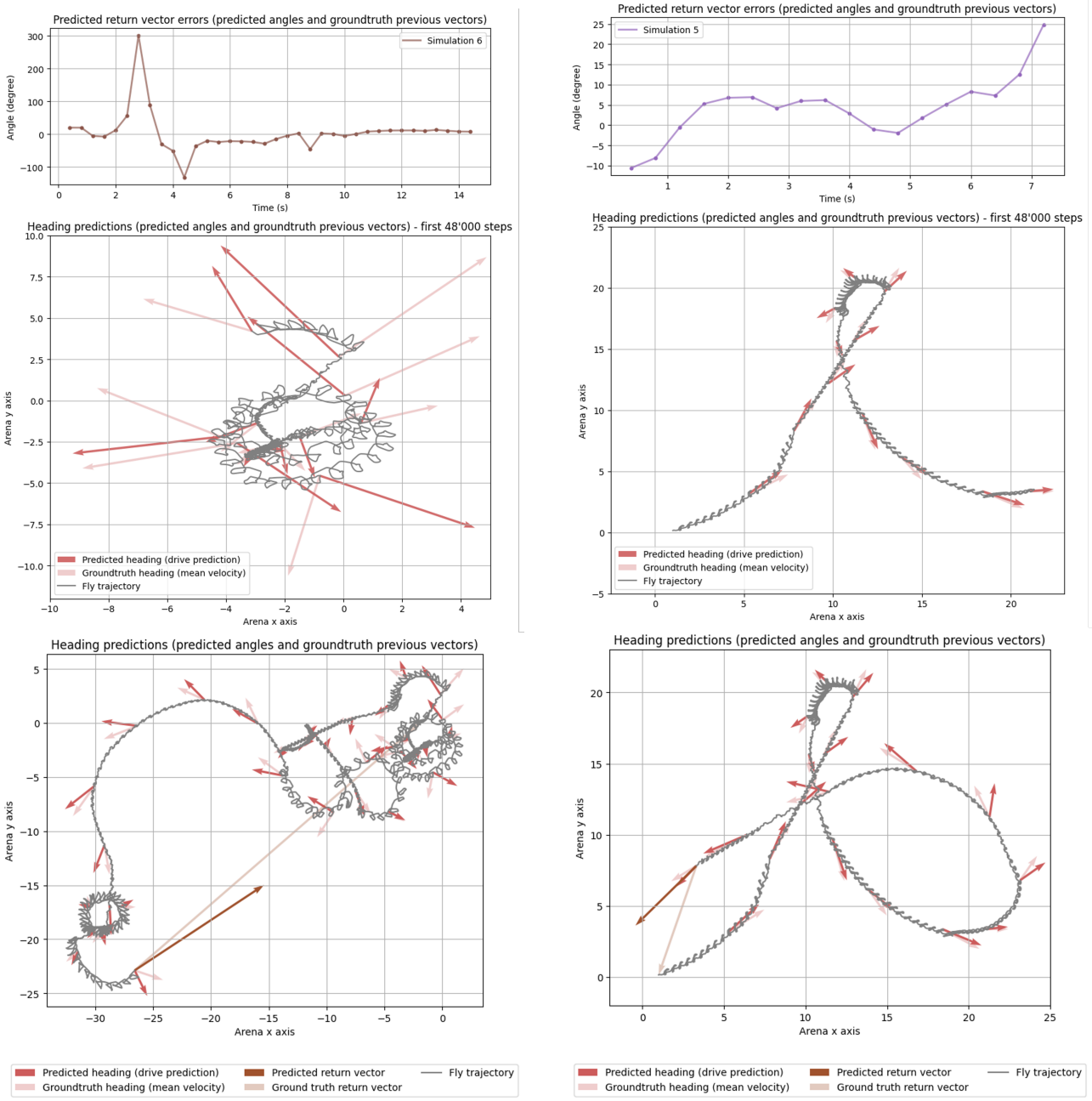
We also plotted the simulation 5 with this second method. In the Fig 16b we can observe very precise predicted headings (in darker red) leading to precise return vectors (top).



(a) Simulation 2

(b) Simulation 5

Figure 15: (top): Angle difference between "predicted" return heading vector and groundtruth reported from Fig 14. (middle): Zoom on an interesting trajectory part, with in darker red the predicted heading vectors and in lighter the corresponding groundtruth headings (mean velocity vectors). (bottom): Entire trajectory with predicted and grounthruth headings and "predicted" and groundtruth last return vector.



(a) Simulation 6

(b) Simulation 5

Figure 16: (top): Angle difference between "corrected/predicted" return heading vector and groundtruth reported from Fig 14. (middle): Zoom on an interesting trajectory part, with in darker red the corrected/predicted heading vectors and in lighter the corresponding groundtruth headings (mean velocity vectors). (bottom): Entire trajectory with corrected/predicted and grounthruth headings and "corrected/predicted" and groundtruth last return vector.

4.5.3 Step length VS heading change

Model fitting

To implement this predictive model, we decided to maintain the same sliding window size of

4'000 steps, to facilitate a more direct comparison with the previous model. The using of multiple features to fit this model, however, renders the visualization of our linear regression more complex, this is why we directly observe results of predictions using this model.

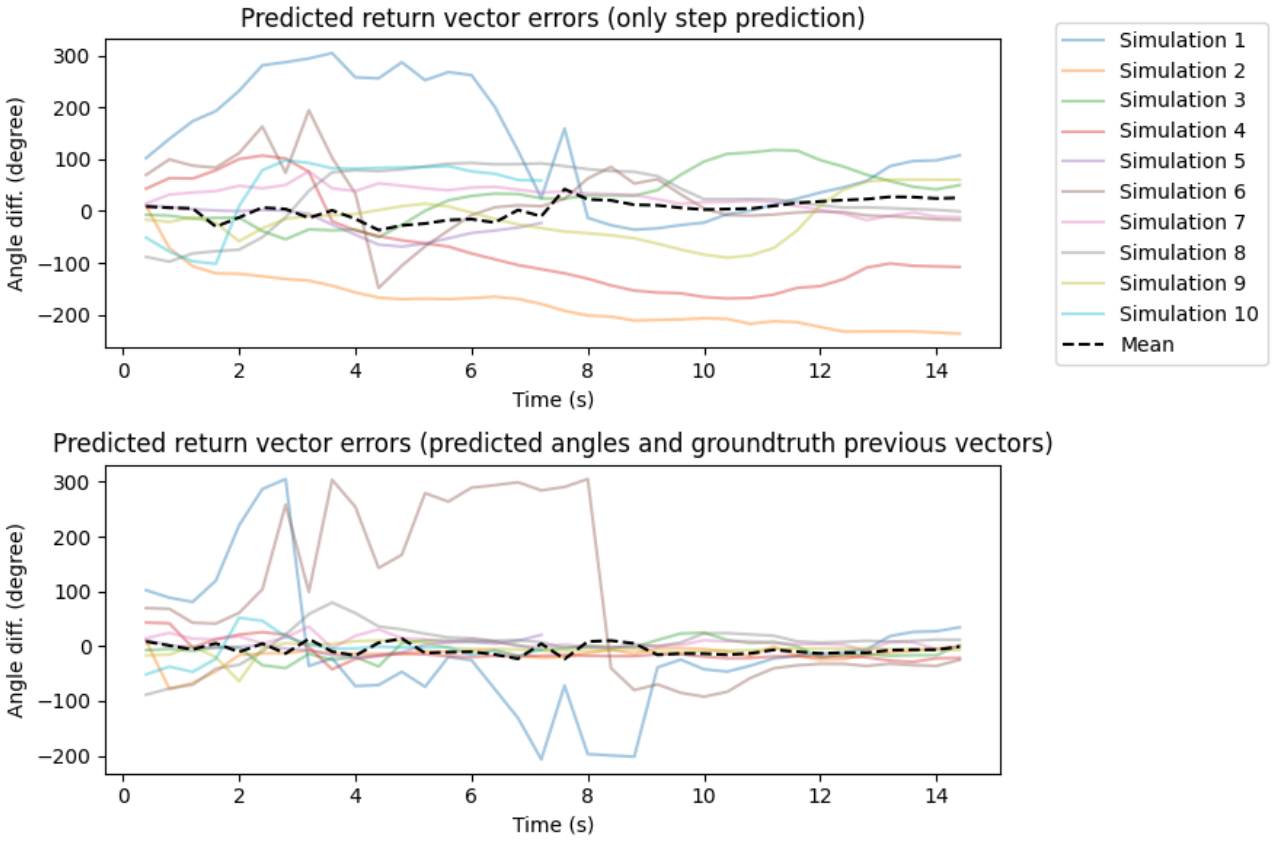
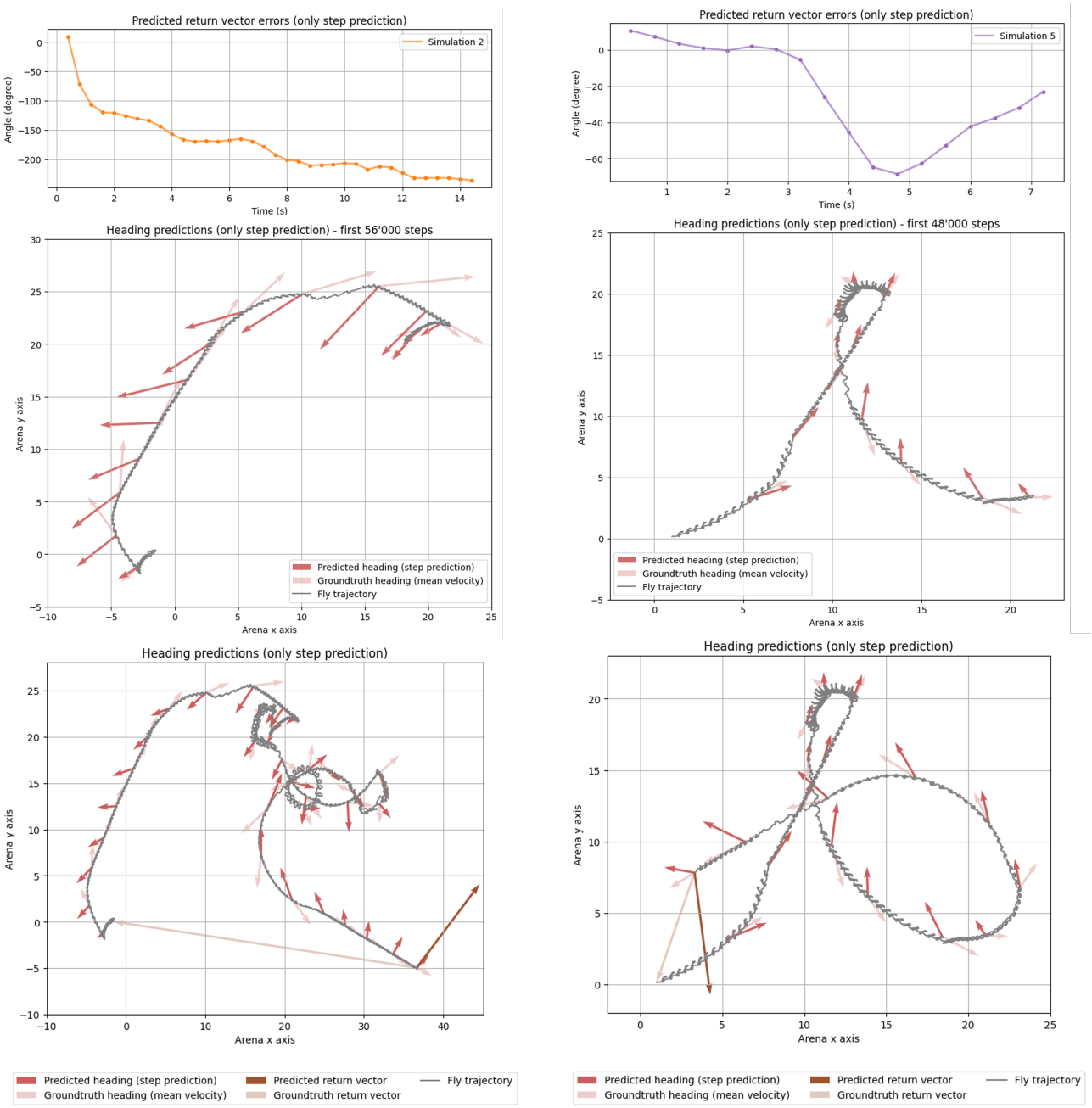


Figure 17: (top): Angle differences, computed at 4,000-step intervals, between the "predicted" return vector obtained from heading vectors predicted (using predicted angles and previously predicted vectors) and the groundtruth return vector across 10 test simulations. The average angle difference is visually presented as a dashed line. (bottom): Same angle differences, but the "predicted" return vector is determined using predicted heading vectors based on predicted angles and the groundtruth previous angle.

Predictions using only predicted vectors

As for the model using descending drives, the top section of Fig 17, which illustrates the error in predicted return vectors, displays considerable divergence and variability, fluctuating between -200° and 300°. We can note that in comparison to Fig 14 (top), the variations here appear more continuous with fewer abrupt changes. A significant distinction is also observed in that relatively large errors are evident from the first 4,000-step interval (simulations 1-4-6-8-10).

We also focused our attention to simulation 2, depicted in Fig 18a, where we also observed a decrease in the error angle at the second prediction point, attributed to a sharp turn. However, using this method, the error persists and propagates throughout the entire trajectory without being compensated. With this method, the "smooth" simulation 5 also shown a better result but with a larger drop at the level of the end of the turning behavior compared to the one observe in Fig 15b.



(a) Simulation 2

(b) Simulation 5

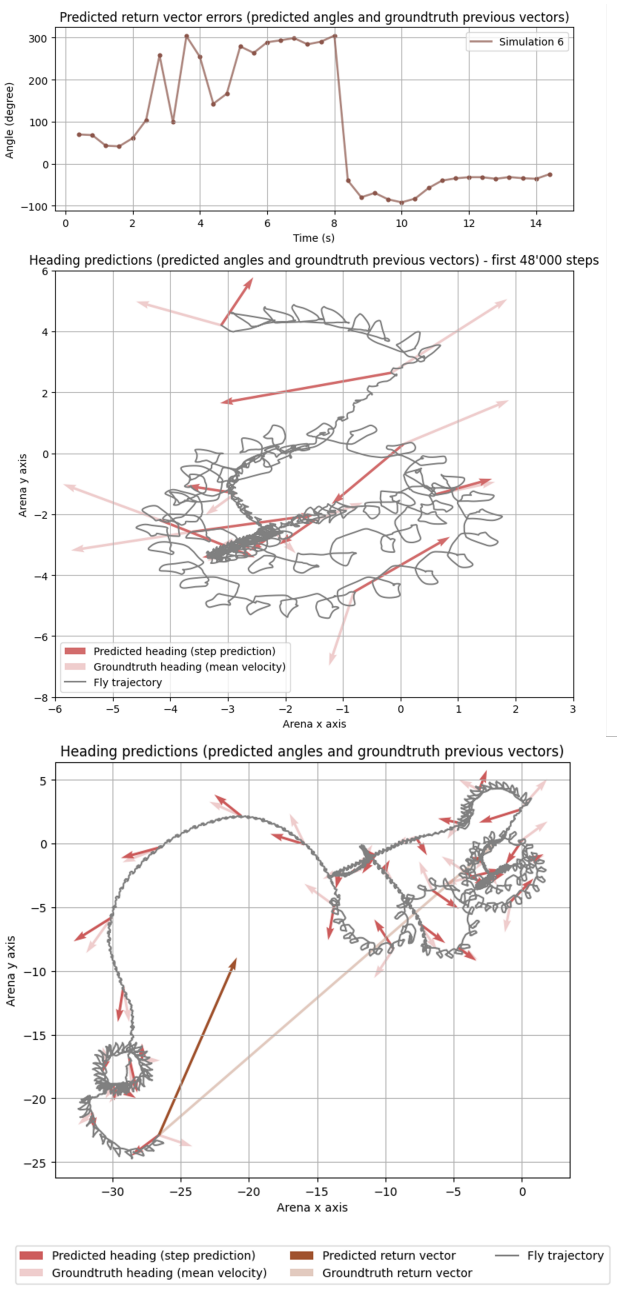
Figure 18: (top): Angle difference between "predicted" return heading vector and groundtruth reported from Fig 17. (middle): Zoom on an interesting trajectory part, with in darker red the predicted heading vectors and in lighter the corresponding groundtruth headings (mean velocity vectors). (bottom): Entire trajectory with predicted and grounthruth headings and "predicted" and groundtruth last return vector.

Predictions using predicted angles and groundtruth headings

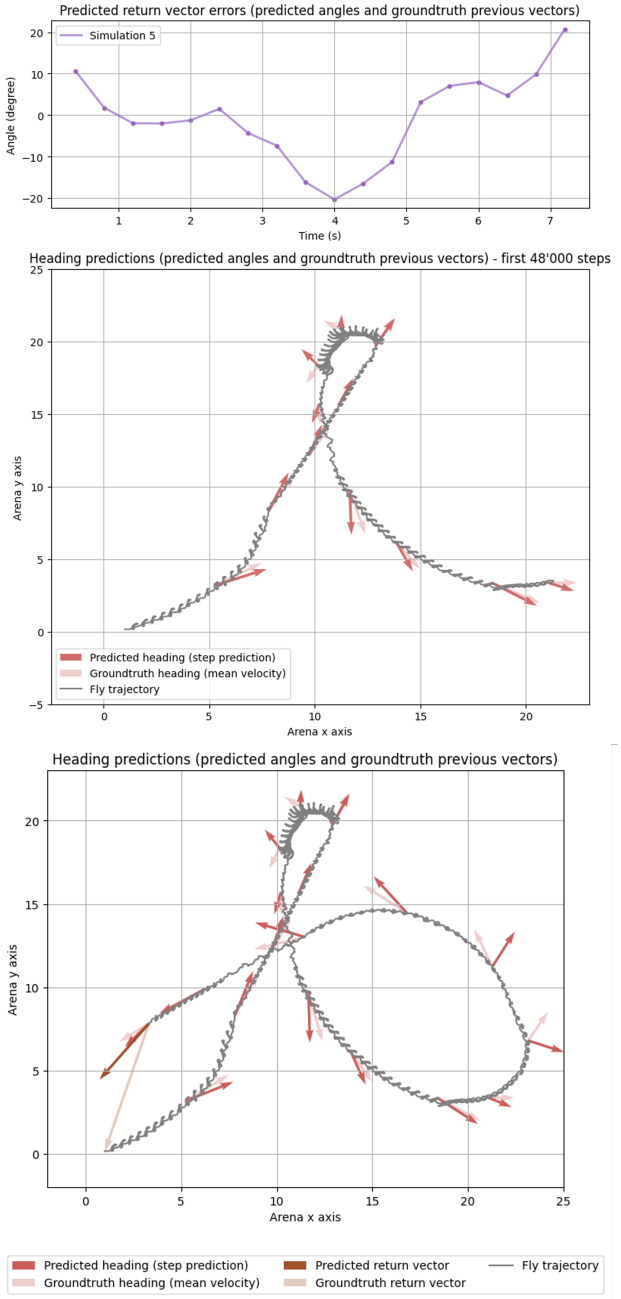
We then implemented the second method, using the "groundtruth heading" correction for the predicted heading vector. Similar to the descending drive predictive model, this significantly improved

our predictions, as illustrated in Fig 17 (bottom). Here, errors remain relatively large for the initial predictions, but subsequently, for most simulations, angle errors stay close to 0°. However, notable deviations are observed in simulations 1 and 6.

Indeed, as shown in Fig 19a, the initial segment of the simulation is complex, with multiple turns and inaccuracies in angle predictions using this model. Regarding the predictions for simulation 5 with the "corrected" method, notable improvements are evident, even though there are still incorrectly predicted headings in turning trajectories.



(a) Simulation 6

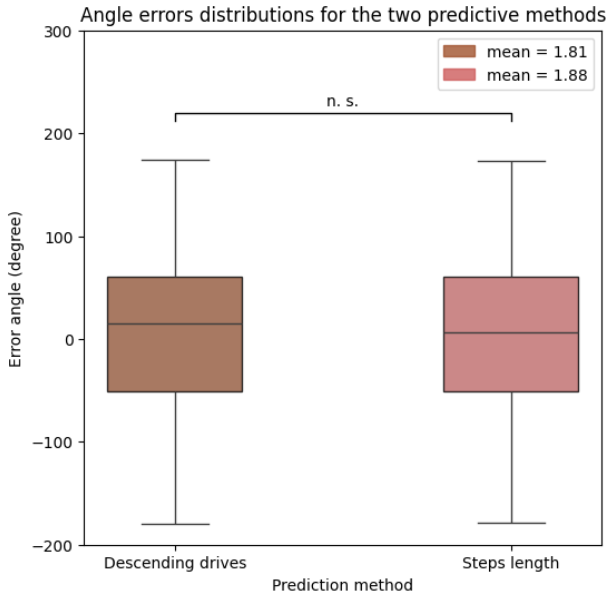


(b) Simulation 5

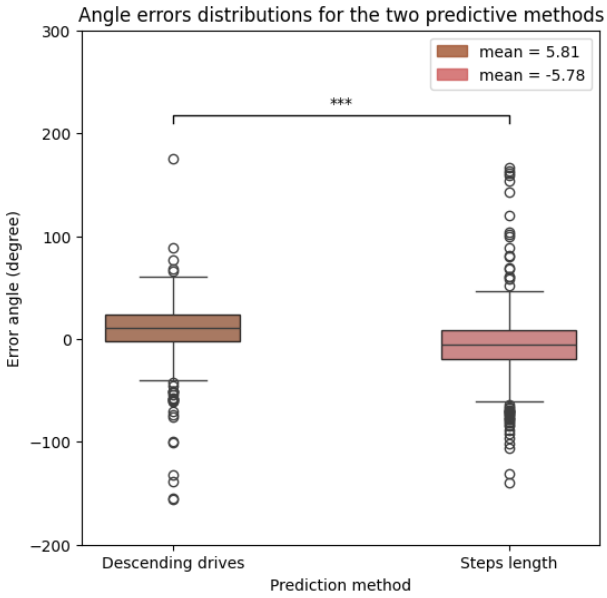
Figure 19: (top): Angle difference between "corrected/predicted" return heading vector and groundtruth reported from Fig 17. (middle): Zoom on an interesting trajectory part, with in darker red the corrected/predicted heading vectors and in lighter the corresponding groundtruth headings (mean velocity vectors). (bottom): Entire trajectory with corrected/predicted and grounthruth headings and "corrected/predicted" and groundtruth last return vector.

4.5.4 Models comparisons

In Fig 20a, both methods show a similar error distribution characterized by a large range between the minimum and the maximum. The paired-sample t-test gives a p-value above the significance level (0.05), indicating insufficient evidence to reject the null hypothesis which states that there is no



(a) Using only predicted vectors



(b) Using predicted angles and groundtruth headings

Figure 20: Boxplot representing distributions of all obtained angle errors for returning vector (Fig 14 and Fig 17) for the different conditions. A `ttest_rel` was performed to compare means of results obtained with descending drive and step length method ($p < 0.001$ '***', $p < 0.01$ '**', $p < 0.05$ '*', $p \geq 0.05$ 'n. s.')

significant difference between these two groups.

The Fig 20b depicts retracted boxplots when using predicted angles and groundtruth headings. A reduced interquartile range indicates that values within the middle 50% of the error data set are closely clustered, suggesting lower variability. The statistical test reveals a p-value below the significance level (0.05), leading to the rejection of the null hypothesis. Consequently, a significant difference between the means of the paired samples can be concluded.

4.6 Final simulation

For our final simulation we created a class that managed the different modules. The simulation was separated in different stages.

- 3000 step of inaction, waiting for the fly to be stable
- Start of the random walk, for an arbitrary time
- Start of the PID to return at the origin
- Near the origin, at $\sqrt{2}$ mm from the depart point, stop

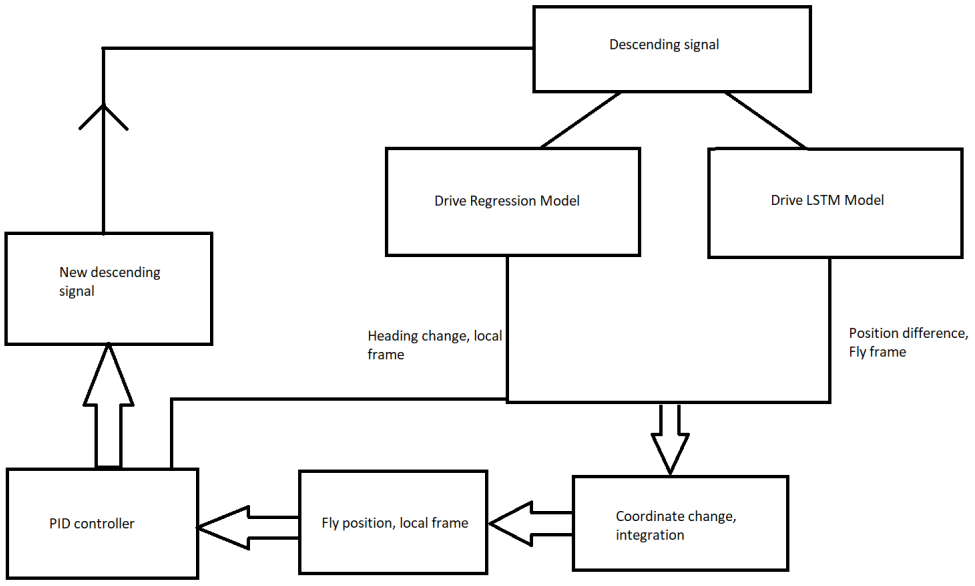


Figure 21: Modules of the final simulation for the return. At each iteration the descending signal is updated, allowing us to make new prediction every 100 steps. Note: As we faced problem with the continuity of the drive regression model, we had to use the ground truth heading and not the estimated one

We got the following trajectories:

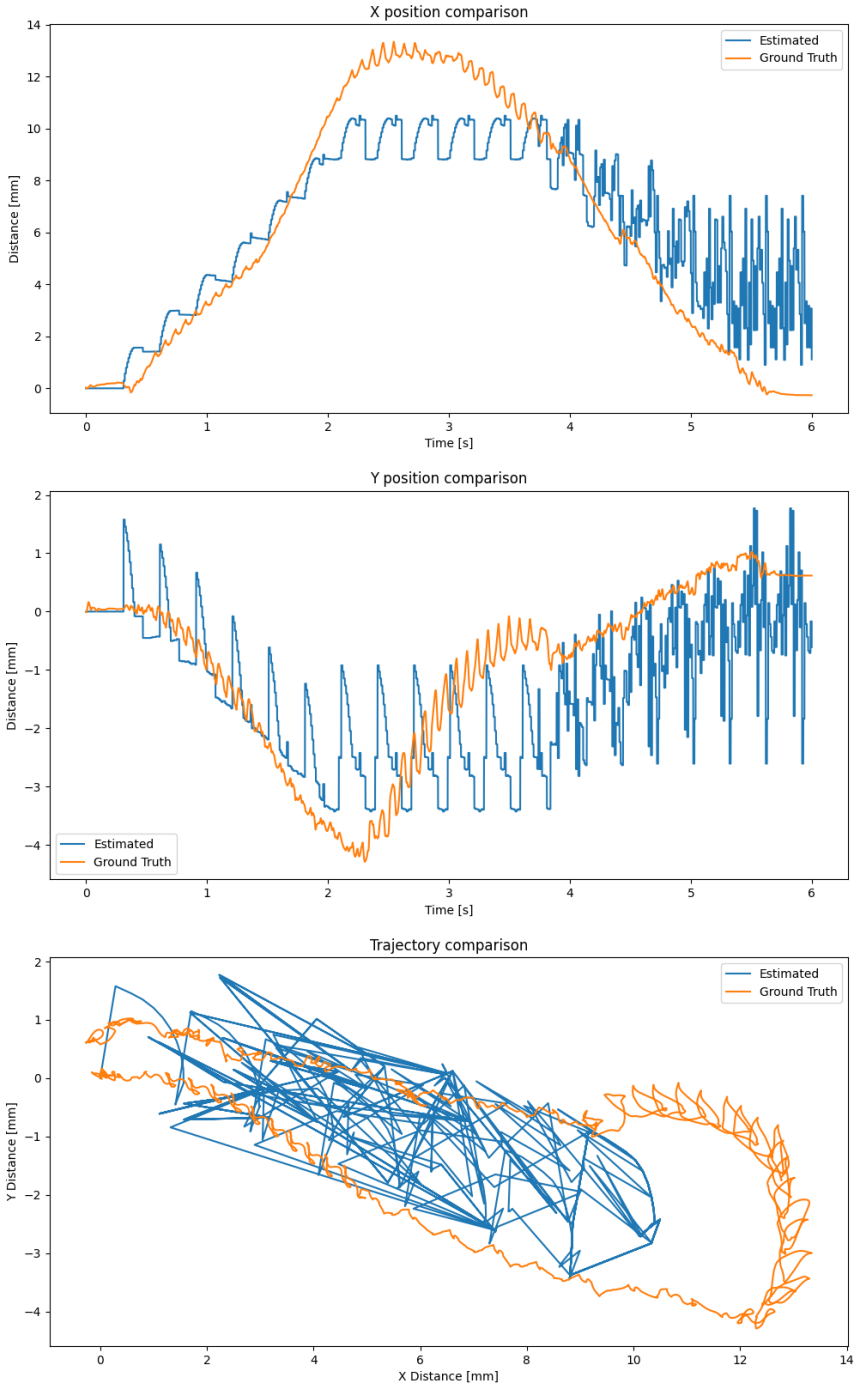


Figure 22: Trajectories of the final simulation where the fly does not manage to reach the target. Due to a big drift in the Y coordinates, the Drospohilia ended at (-4,4), instead of (0,0) even though the fly thought she had reached her goal. Appendix [Video 10]

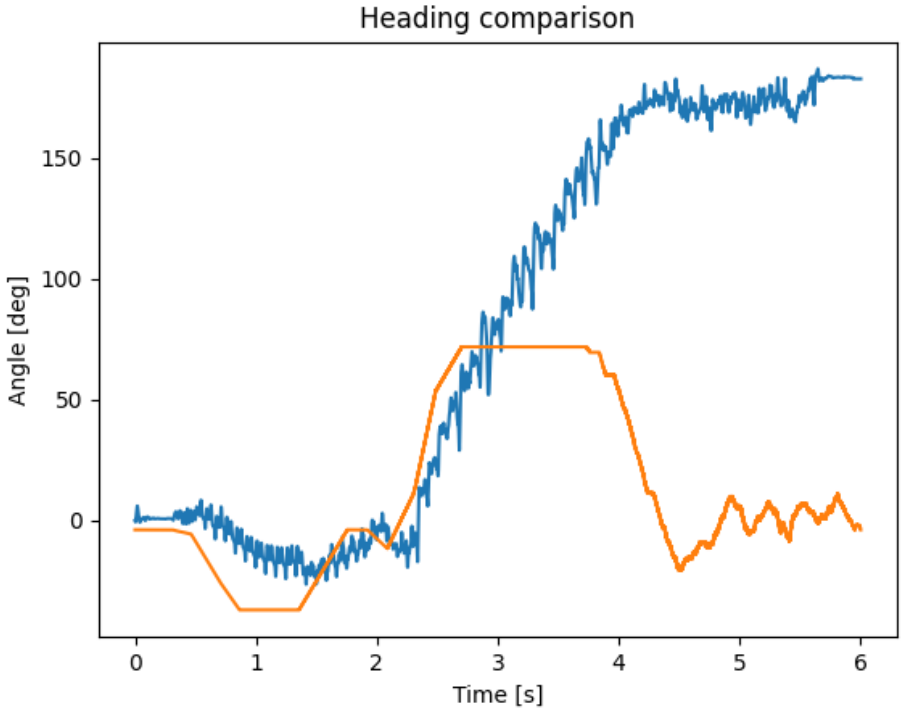


Figure 23: Ground truth heading(blue) and estimated one (orange, drive regression) during the same simulation as Fig 22. As we can see here the estimation model is limited to 80 (and -80)

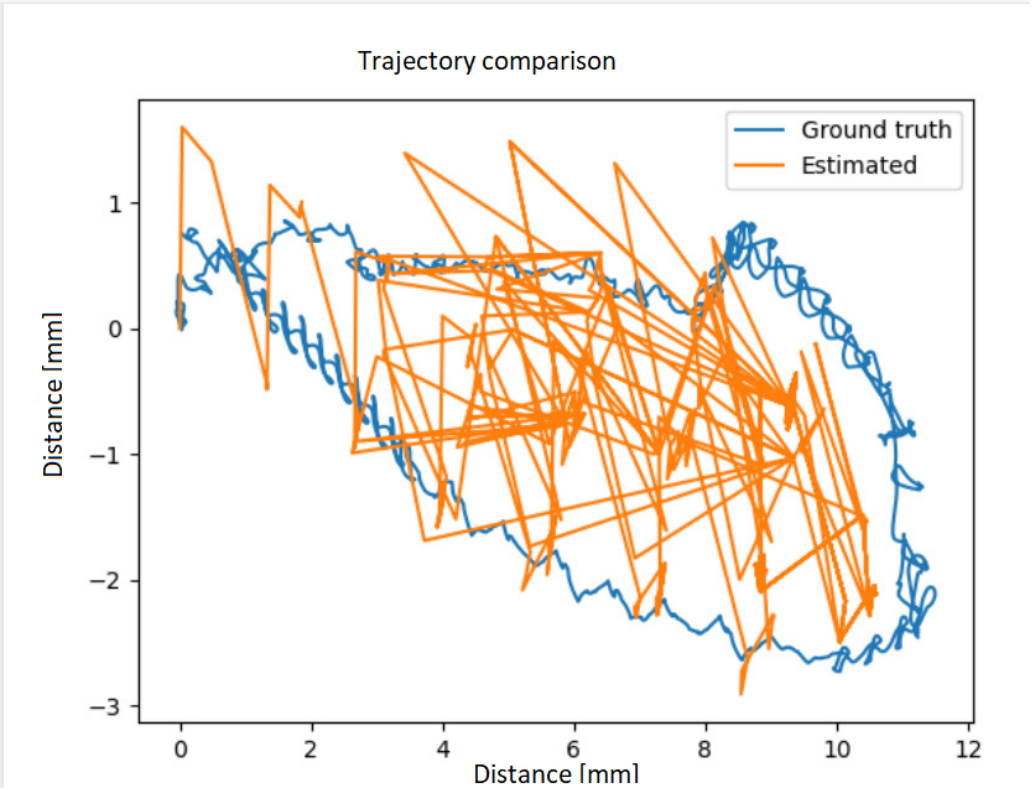


Figure 24: Successful simulation, the fly manages to reach the end. We can observe that the estimated trajectory is still very chaotic but thanks to the interaction between the heading and the fly position estimation the fly comes back to the starting point. Appendix [Video 9]

5 Discussion

5.1 Turning behavior

Regarding the descending drives [0.2, 1.2], as previously discussed, these coefficients are more indicative of a smooth direction change rather than a distinct turning behavior, suitable for incorporation into the random walk algorithm. In the case of [-0.3, 1.2], observation of Fig 7 and [Video 3] in the Appendix reveals that the fly achieves a slight and incomplete turn but with a static and unnatural movement. We can also notice that using such methods with very small amplitudes for internal leg movements could result in a leg's blockage, causing the leg to become stuck below the fly's body. To address this problem, we can implement an artificial step triggered by contact force sensors when a stuck leg is detected.

Subsequent to our results analysis, it becomes evident that the descending drive combination of [-1.2, 1.2] leads to a turning behavior most closely similar to the biologically relevant turn observed in [Video 1] of the Appendix, showing a real fly turning. However, it is crucial to note that a problem arises when controlling turning behavior using this approach, as turns obtained with identical descending coefficients are not consistently uniform and can give different turning behaviors. The efficacy of the turning behavior depends on the fly's internal state (CPG state) at the initiation of the turn. This lack of robustness in our turning behavior could be explain by the fact that all along the simulation CPGs can start synchronizing between legs of the same side and between sides. This synchronization results in an incoherent walking pattern and restrains the propulsive phase of the fly, and preventing turning behavior.

We tried to achieve a more robust and biologically relevant turning behavior, but our attempts were unsuccessful. Unfortunately, resource constraints limited our ability to address this issue. Indeed, we faced limitations inherent in the overall implementation of the NeuroMechFly model and the data governing NeuroMechFly locomotion. Since we couldn't invest more resources in resolving these issues and as we could manage to continue our project with such turning behavior we decided to keep these implementations for the rest of our project.

5.2 Path integration problem

As depicted in Fig 10, the velocity vector aligns precisely with the noisy trajectory of the fly. However, due to the loop-like patterns in the fly's noisy trajectory, the cumulative summation of velocity vectors averaged out their variability, enabling the integration of a precise return vector. This integrated return vector is not always an exact match with the ground truth, but it consistently remains in a close proximity.

While the orientation vectors appear to represent the overall shape of the trajectory, aligning with tangential vectors, the orientation from the simulation does not consistently match the actual fly heading, particularly during turns. These inaccuracy accumulates across the entire trajectory, resulting in a cumulative sum of these orientation vectors that leads to a less accurate return vector.

Additionally, the velocity vector incorporates information about the fly's speed, providing insights into the intensity of its movement and probably enhancing the precision of the cumulative sum. In contrast, the orientation vector ignores changes in velocity, potentially contributing to its cumulative sums that may not accurately represent the overall movement trajectory.

Following the analysis of these two vector types, we decided to use the velocity vector as the fly heading indicator for the rest of our project.

5.3 Heading change regression

5.3.1 Heading changes

This analysis indicates that using the mean velocity vector computed over a specific window of steps is coherent and allows the integration of more accurate heading changes throughout trajectories, without disturbing the integration of the return heading vector, compared to the mean orientation vector. The noise in velocity vectors is effectively smoothed out through averaging across the sliding window.

This section comes to validate our findings in the path integration section and confirms our decision to use velocity vectors and their means to represent heading changes between various points along the trajectories for our regression task.

5.3.2 Drive difference VS heading change

Upon observing Fig 13, we identified a distinct pattern associating descending drives with heading changes. However, we described the presence of numerous outliers, deviating significantly from the fitted linear regression. These outliers suggest that, for a specific combination of descending drives (depicted here as a difference between left and right drives), the fly's behavior can vary. This lack of robustness, already discussed in the "Turning behavior" section, likely contributes significantly to the inaccuracies observed in the results section.

In the analysis of Fig 13, we also mentioned that the linear regression intersects the (0,0) point. This observation is significant in our data context, indicating that when the left and right descending drives are equal, the fly proceeds straight ahead without altering its direction. Fig 13 also indicates that when $\delta_L - \delta_R > 0$, the fly turns left, and when $\delta_L - \delta_R < 0$, the fly turns right. This is consistent

with what we explained in the turning behavior section, where internal legs have a smaller or negative descending drive compared to the external ones.

After discussing Fig 14 and 15-16, what comes out of this analysis is that the major challenge for the biologically relevant prediction method lies in accurately predicting sharp turns. These turns are not consistently predicted, leading to error propagation throughout the remainder of the trajectory and resulting in highly imprecise return headings. Examining the initial stages of Simulation 2, as previously described, [Video 2] shows that these "sharp turns" correspond to transitions between different descending drive combinations. This suggests that to improve our predictions, a crucial aspect may lies in updating the random walk implementation to gradually alter descending drives. The improved performance observed for smoother trajectories further supports this. A more detailed study of actual fly displacements is should be done to enhance the implementation of abrupt directional changes.

5.3.3 Step length VS heading change

Following the analysis of Fig 17 (top) and Figs 18, it can be deduced that the method relying on proprioceptive signals, only relying on biologically relevant predictions, does not lead to better results and has difficulties to integrate complex turning trajectories. Notably, in Simulation 2, the observation that angle errors tend to exhibit greater stability after the sharp turn (Fig 18a), compared to case involving descending drives (Fig 15a) could suggest a more precise prediction for smooth trajectories. Indeed, error variations after a sharp turn region could be explain by prediction inaccuracies.

When observing raw predicted angles, excluding the influence of propagated error, in Fig 17 (bottom), it becomes evident that this model delivers quite accurate return headings, with the exception of the two erratic trajectories, simulations 1 and 6. The rest of the simulation error line remain very close to 0°. Fig 19a illustrates the inaccuracy of heading predictions in the "multiple turns" region within the initial 8 seconds and also the better accuracy for smoother regions at the end of the simulation.

5.3.4 Models comparisons

Both predictive models lead us to the same conclusion: the major problem encountered lies in the inaccurate prediction of heading during turning behaviors, resulting in imprecise final integration. Fig 20a illustrates that both predictive regression methods, using descending drives and step length, lead to similar unsatisfactory results. This is mainly explained in addressing turning behaviors, leading to the propagation of integration errors.

Initially, we might presume that using proprioceptive signals could lead to better results, as relying on more "concrete" signals avoiding the lack of robustness of the descending drives signals to control turning behavior. Fig 20b reveals a notable difference in mean results obtained through these distinct methods. The proprioceptive prediction appears to have a smaller mean error, with the median of the error distribution closer to 0°. Furthermore, the influence of outliers caused by simulations 1 and 6, significantly impacts this distribution. All these observations suggest that the prediction method using step length could potentially improves path integration for smooth trajectories.

As we already mentioned, a more in-depth examination of fly displacement patterns is necessary for updating our predictive approaches. Additionally, we should consider that inaccuracies in predictions may be associated with the computation of signal information used for model fitting, especially within the sliding window. The presence of "sharp junctions" responsible for abrupt turns makes prediction more challenging. A deeper understanding of the "weight" assigned to each element within the sliding window could potentially lead to improvements, enhancing the robustness of the data.

5.4 Final simulation results

In the end we managed to get a fly coming back around the starting point but that has a hard time coming back right on it consistently. Despite the quality of the regression predictions, we did not managed to include it in our final simulation as it was not made to support high rotation drive signal. We are still happy that, despite our position prediction being noisy and subject to error accumulation, the combination with the heading allow the fly to walk back toward her spawning place.

5.5 Further objectives

By the end of our project, we have only reach the initial phase of a comprehensive exploration into various approaches for providing NeuroMechFly with biologically relevant spatial awareness. Further objectives can be outlined to continue the exploration and development of a robust path integration method.

First, it is essential to enhance the overall random walk and turning behavior to achieve more realistic trajectories, facilitating the prediction process based on these displacements. The existing displacements observed in the various videos still show excessive irregularities.

Following the establishment of an effective path integration mechanism allowing the fly to accurately locate itself, we can introduce environmental complexities, such as obstacles or attractive zones, to make the task more realistic. Moreover, this could result in a comprehensive model, providing neuroscientists and roboticists with a platform to test and formulate complex hypotheses.

6 Gant chart

6.1 Gant chart: Expectation

The initial Gant chart, created at the project's beginning, has been transferred with Dropbox along with other materials of the Appendix.

6.2 Gant chart: Reality

The final Gant chart, re-updated at the end of the project, has been transferred with Dropbox along with other materials of the Appendix

6.3 Faced problems

In the first phase of the project, we spent time comparing the "orientation" and "velocity" vectors extracted from the simulation. We noticed inconsistencies in both and worked to understand them to finally choose the most suitable vector for our project.

Dealing with the turning behavior was quite a challenge in our project. After the mid-term presentation, we spent a lot of time trying to improve it. We tried to create a new algorithm that combines CPGs and new rules for a more biologically relevant turning behavior. However, we realized the limitations of the simulation, leading us to resolve the issue by using the preprogrammed "Turning controller."

After all of this, we decided to change our approach for "heading prediction." Initially, we planned to use visual cues and complex neural networks to extrapolate the return heading vector. However, upon reflection, we chose to start with a straightforward method, based on linear regressions, to establish a basic prediction model. We were aware that getting conclusive results only from this might be challenging due to simulation limitations, but it serves as a foundation for understanding the task's challenges and exploring more advanced methods later. We chose to explore two distinct linear

regressions based on two sets of features: descending drives and leg proprioceptive signals. However, during this period, we duplicated the effort by concurrently employing different methods for the same task. In the end, we spent a lot of time on this to create the pipeline for such predictive models and we finally only choose one method. This meant the time spent on the other method could have been used for something else. During the final part of the project, the understanding of what causes these inaccurate predictions and the desire and trials to improve these results cost a lot of time.

7 Appendix

Dropbox link: Here

Video 1: "Turning_behavior" is a video capturing the actual evolution of flies within an experimental arena. The video, obtained from the lab, is intentionally slowed down to facilitate a more detailed examination of leg movements during turning behavior.

Video 2: "random_walk_1" shows the first simulation obtained with the random walk algorithm corresponding to the simulation 1 on the Fig 6

Video 3: "random_walk_2" shows the second simulation obtained with the random walk algorithm corresponding to the simulation 2 on the Fig 6

Video 4: "random_walk_3" shows the second simulation obtained with the random walk algorithm corresponding to the simulation 3 on the Fig 6

Video 5: "turn_02_12" shows the change in the fly's direction when the descending drives are set to [0.2, 1.2].

Video 6: "turn_03_12" shows the change in the fly's direction when the descending drives are set to [-0.3, 1.2].

Video 7: "turn_12_12" shows the change in the fly's direction when the descending drives are set to [-1.2, 1.2].

Video 8: "turning_robustness" shows the different trajectories obtained with the same descending drives ($[\delta_L,\delta_R]$ = [-1.2, 1.2]) in three successive simulations (without resetting CPG network)

Video 9:"final_simulation1" shows a successful run of the final simulation

Video 10: "final_simulation2" shows another run where the fly diverged from the starting point.

Code organisation: We uploaded our code on our branch of "flygym-scratch" repository:

notebooks file

- "1_random_walk_generation.ipynb": Notebook allowing to generate a random walk of a given number of step and save important features at each timestep in a csv file in the output file (csv_random_walk.csv)

- "2_prediction_preprocessing.ipynb": Notebook allowing to preprocess random walk data to facilitate the preparation of linear regressions and predictive models. The preprocessed data obtained are store as data frame in a pickle structure (preprocessed_data_w_4000.pkl).
- "3_prediction.ipynb": Notebook allowing to fit linear regression models and explore possible regression using these models. The "preprocessed_data_w_4000.pkl" file is loaded in the DropBox to run this last Notebook with our data.
- "4_path_NMF_test.ipynb": Notebook allowing to launch final simulation and get the plot associated to it. In case of SSH run, the main of the class path_NMF can be used as it produces the same plot
- "5_Lstm_position.ipynb": Notebook used to process the data and train the LSTM networks used during the final simulation.

References

[1] Pembe Gizem Özdil Louise Genoud Femke Hurtak Sibo Wang-Chen, Victor Alfred Stimpfling and Pavan Ramdya. Neuromechfly 2.0, a framework for simulating embodied sensorimotor control in adult drosophila. *Bio Rxiv*, 09 2023.

[2] Tim Currier and Katherine Nagel. Multisensory control of navigation in the fruit fly. *Current opinion in neurobiology*, 12 2019.

[3] John C. Tuthill and Eiman Azim. Proprioception. *Current Biology Magazine*, 04 2018.

[4] Matasaburo Fukutomi and Bruce A. Carlson. A history of corollary discharge: Contributions of mormyrid weakly electric fish. *Frontiers in Integrative Neuroscience*, 07 2020.

[5] Stephanie J. Preuss Chintan A. Trivedi Johann H. Bollmann Mir Ahsan Ali, Katharina Lischka. A synaptic corollary discharge signal suppresses midbrain visual processing during saccade-like locomotion. *Nature Communications*, 11 2023.

[6] Johannes D. Seelig and Vivek Jayaraman. Neural dynamics for landmark orientation and angular path integration. *Nature*, 05 2015.

[7] Sung Soo Kim, Hervé Rouault, and Vivek Jayaraman Shaul Druckmann. Ring attractor dynamics in the drosophila central brain. *Science*, 05 2023.

[8] Marie Dacke Robert Mitchell, Shahrzad Shaverdian and Barbara Webb. A model of cue integration as vector summation in the insect brain. *Proceedings of the Royal Society B*, 05 2017.

[9] Roman A. Corfas Amir H. Behbahani, Emily H. Palmer and Michael H. Dickinson. Drosophila re-zero their path integrator at the center of a fictive food patch. *Current biology*, 10 2021.

[10] Laia Serratosa Capdevila Quinn X. Vanderbeck Atsuko Adachi Richard S. Mann Rachel I. Wilson Helen H. Yang, Luke E. Brezovec. Fine-grained descending control of steering in walking drosophila. *Bio Rxiv*, 10 2023.

[11] Omer Mano Matthew S. Creamer and Damon A. Clark. Visual control of walking speed in drosophila. *Neuron*, 12 2019.