Modelling male-female fly courtship chasing behaviour

Eugenia Chiappe Lab Rotation work[*]

Francisco Moreira de Azevedo

June 2023

1 Abstract

Studying animal behavior is challenging due to the inability to know the moment to moment state and goal of the animal. Decision making tasks constrain the animal's behaviour by reducing it into limited possible outcomes (correct, incorrect, miss, etc.) in order to understand how the animal might be transforming the information available to it. However the representations and algorithms the animal might be using are most often implicitly assumed. For example in any freely moving decision-making task there's a possibility that an animal might be using information about its own body to predict choice instead of transforming external inputs. An example would be a mouse using its own body movement, instead of the stimuli, to estimate elapsed time in a time categorization task.

In contrast during natural behaviours such as courtship, hunting and escape we know animals' explicit goals - mating, feeding, survival, respectively. This inherently reduces the behaviour to a specific immediate objective, amenable to our understanding, without the need for a task with inherent human biases concomitant with anthropocentrism.

In the case of fly courtship behavior, the male pursuits the female by minimizing the angle between its body and the back of the female, while also minimizing the distance. It achieves this by using its forward, sideways and rotational velocity. Despite clarity in the inputs and outputs of this behavior, the algorithm and it's mechanistic implementation are unclear. One added challenge is that according to data from Miguel Paço (unpublished) (also other papers?), flies react to changes faster than what's achievable by their visual processing (40-80ms) [1], hence suggesting a predictive or memory driven internal model.

To address this challenge, we used a control theory approach in which the current error angle is fed back, continuously updated and acted upon, to model fly courtship chase

[*] Final presentation **here**. Code and any other documentation housed **here**

behavior. A similar approach has been used in which the authors controlled the error angle via angular velocity only, using a Proportional (P) and Derivative (D) controller [4]. Despite this, evidence from Miguel Paço (unpublished) shows that flies consistently use their sideways velocity to minimize the error angle. Taking this into account we used two controllers in parallel to control the error angle: one for the fly's angular velocity, another for its sideways velocity, keeping the forward velocity matched to that of the instantaneous velocity of the female. By comparing the model's behavior when using both controllers separately or together against real fly data we can understand which algorithms the fly might be using.

2 Problem statement

During chasing behavior the male approaches the female from its anterior part of the body. This chase can be seen geometrically, in which the objective of the male is to:

1. minimize the **angle** $e = \lambda - \gamma$ between its orientation and the anterior part of the female (or target T);

2. minimize the **distance** d from the female.

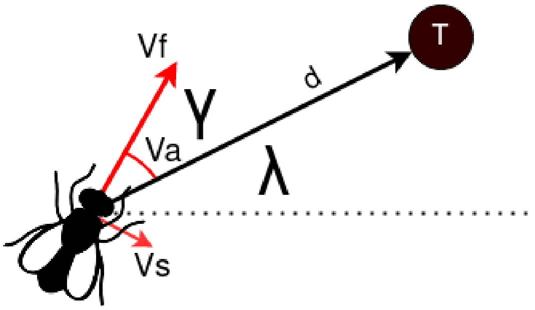


Figure 1: Geometric representation of inputs and outputs of fly chase

To achieve its objective the fly can move in two dimensions (decomposable into forward and sideways velocity, V_f and V_s, respectively) and rotate on itself (rotational velocity V_a). Throughout this work V_f has been chosen to match the target's instantaneous velocity, hence one of the objectives is not covered. In this work we also neglected the head's movement relative to the body as well as the eye movement which has been shown to occur in drosophila [3].

In this work we used a control theory approach to first simulate a virtual chase agent and secondly model real data of fly chasing an experimenter-controlled magnet.

3 Background on insect courtship behaviour

The first application of control theory to insect courtship behaviour was performed in the 70s by Land and Collet [5]. By inspecting behavior they suggested that houseflies chase their mates using a PD controller, in which the proportional component responds irrespective of target's position on the retina, whereas the derivative component is only used when the target is on the front field of view (probably corresponding to the FOV in which there is binocular overlap). This is advantageous since otherwise the proportional term would be small when the error is small and the control of the error would be slower without the derivative component.

Focusing on the Drosophila melanogaster model, Robert Cook [2] did the first careful analysis showing XYZ.

The most recent model [4] suggested for drosophila courthsip behavior is a PD controller on angular velocity. In this paper Sten et al sustain specific claims:

1. P1 neurons release and reflect a dynamic state of sexual arousal but decayed when animals transiently stopped courting despite still having future "enduring" arousal;

2. Furthermore, activating P1 neurons induces courtship similar to spontaneous natural chasing. It also induces LC10a activity similar to the ones evoked by natural chasing (Fig.3E);

3. LC10a's activity reflects changes of target on the retina - phase of fluctuations in LC10a match those of target. LC10a's activity gain increases when courting (but not when moving);

4. Optogenetic manipulation (Fig.2G-H) show LC10a's sufficiency for ipsilateral orienting behaviors. Anterograde tracing shows that downstream synpatic partners innervate lateral accessory lobe (LAL) neurons (Ext.Fig.9G-K).

5. During spontaneous courtship P1 activity leads LC10a's by 500ms (Ext.Fig.10A).

6. LC10a neurons respond identically to targets of different sizes Ext.Fig.8M-N, consistent with behavioral data [6].

There are also details about the model important to our work, namely:

1. Most importantly the model implements a Proportional controller TODO CHECK WHAT IS DONE WITH THE OUTPUT OF THE NEURONS, as a code snippet (Appendix I) from the **source code** of the paper shows.

2. The model is composed of LC10a left- and right-FOV tuned integrate and fire neurons. These neurons' receptive fields cover **each** around 10.5^{o} of the FOV of the fly, together they span the whole FOV. The output is the subtraction of one population's activity from another (a form of circuit competition).

3. There is a binocular overlap of 15^{o} at the center (implemented as 30^{o} in source code though) which is necessary for the model to exhibit "predictive" behavior (Ext.Fig.11H).

4. AOTu activity (serving as a proxy of LC10a axonal boutons) exhibit direction selectivity (Ext.Fig.11) which is vital for following a target in a time-locked manner, otherwise the model is delayed, phase wise, relative to real fly's data.

5. Including P1 activity as a scale of input current to LC10a neurons changes gain of model output yielding a better match to data (Fig.4I). P1 activity influences LC10as via a continuous-gain, not via threshold (Fig.4J).

6. P1 activity also changes the model's behaviour qualitatively from continuous control to one closer to discrete bouts, similar to the animal's behavior Ext.Fig.13)

7. Ext.Fig.3H-I show that V_a has a maximum of 15 rad/s (860^o/s) and V_s of 35 mm/s (although with different distributions) in agreement with data from Miguel Paço;

During a talk at Cosyne 2023, Matthew Collie (Wilson lab, Harvard) et al extend the work by Sten et al by explicitly modelling courtship behaviour under a control theory approach. Since a proportional controller delayed with regards to reference is not enough to track a reference in a time-locked manner, they state that adding a derivative controller compensates for it. Importantly enough on the plots shown, the derivative component precedes the proportional one.

They go on to suggest AOTU019 as a neuron that might implement derivative control. To test this they suppress AOTU019's activity - as a result, flies are still able to engage in courtship, however it causes the pursuer to lag target motion, as if the derivative component were removed; hence AOTU019 are necessary for faithful pursuit. Furthermore AOTU019's activity scales with target speed, which means it could be coding for the derivative of the error angle. Lastly, activity peaks the most closest to the midline, where the target is in the region of binocular convergence. Evidence from Nuno Rito (unpublished) shows that males cannot pursuit targets when the center of their eyes is coated with paint. Joining these two pieces of evidence one can speculate that the derivative component is perturbed when the region of binocular convergence is coated with paint, moreover this should be reflected in AOTU019's activity.

So far we've focused on controlling the error angle but the other objective for the male is to maintain the distance to the target. Research on beetles etc Beetle paper

Neuronal mechanisms Ann Rev paper for larger context.

Eye VOR eugenia review paper for Integral Windup Reset

4 Results

4.1 Virtual agent modelling - V_a controller

4.2 Effect of controller gains

4.3 Effect of integration window

Text

4.4 Virtual agent modelling - V_s controller

4.5 Virtual agent modelling - Parallel V_a and V_s controller

Effects of different integration windows on the controller

4.6 Model vs Real flies: initial observations

text

5 Conclusions

6 Future work

Problem with derivative kick: https://controlguru.com/pid-control-and-derivative-on-measurement/

Effects of integral reset windup (include VOR literature): https://www.youtube.com/watch?v=ifEsb7BoC

Lag but pursuer uses a predictive linear model to judge the target's direction based on X past positions (would break on cylinders path)

Minimal threshold for activating the derivative part of Va

From sideways velocity spectrum maybe one could guess the optimal window of the controller. If Integral controller integration window is exactly matched to the period of oscillations in target then pursuer chases target in straight line - however does not work if there is integral reset.

Headfix vs Normal

Prediction from covering up eyes effect on derivative

References

[1] Rudy Behnia, Damon A Clark, Adam G Carter, Thomas R Clandinin, and Claude Desplan. Processing properties of on and off pathways for drosophila motion detection. *Nature*, 512(7515):427–430, 2014.

[2] Robert Cook. The courtship tracking of drosophila melanogaster. *Biological Cybernetics*, 34(2):91–106, 1979.

[3] Lisa M Fenk, Sofia C Avritzer, Jazz L Weisman, Aditya Nair, Lucas D Randt, Thomas L Mohren, Igor Siwanowicz, and Gaby Maimon. Muscles that move the retina augment compound eye vision in drosophila. *Nature*, pages 1–7, 2022.

[4] Tom Hindmarsh Sten, Rufei Li, Adriane Otopalik, and Vanessa Ruta. Sexual arousal gates visual processing during drosophila courtship. *Nature*, 595(7868):549–553, 2021.

[5] Michael F Land and Thomas S Collett. Chasing behaviour of houseflies (fannia canicularis) a description and analysis. *Journal of comparative physiology*, 89:331–357, 1974.

[6] Inês MA Ribeiro, Michael Drews, Armin Bahl, Christian Machacek, Alexander Borst, and Barry J Dickson. Visual projection neurons mediating directed courtship in drosophila. *Cell*, 174(3):607–621, 2018.

7 Appendix I - Code snippet from model by Sten et al [4]

Line 12 shows that motion direction is computed as the sign (not the value) of the difference of the position of the female in the male's retina. This is then used as an input in line 19 and 22, yielding *inField* which is an input current used to generate the spikings of the integrate and fire neurons.

```
1   %get distance of female from upper and lower bound of RFs
2   d1 = abs(circ_dist(femaleAngle(hist,:),spatialRFs(k,1)));
3   d2 = abs(circ_dist(femaleAngle(hist,:),spatialRFs(k,2)));
4
5   %get width of RF
6   td = abs(circ_dist(spatialRFs(k,1),spatialRFs(k,2)));
7
8   %check whether female was within spatial RF for each frame
9   presence = (d1 < td & d2 < td)';
10
11  %compute motion direction of target
12  motionDirection = sign(diff(femaleAngle(hist)));
13
14  %constrict to motion direction depending on setting
15  if strcmpi(DIR_SELECTIVITY,'progressive')
16
17      %if neuron is on right side, include when stim moves right
18      if k <= nLC/2
19          inField(k,:) = presence.*[0 motionDirection'<0];
20      %%if neuron is on left side, include when stim moves left
21      else
22          inField(k,:) = presence.*[0 motionDirection'>0];
23      end
```

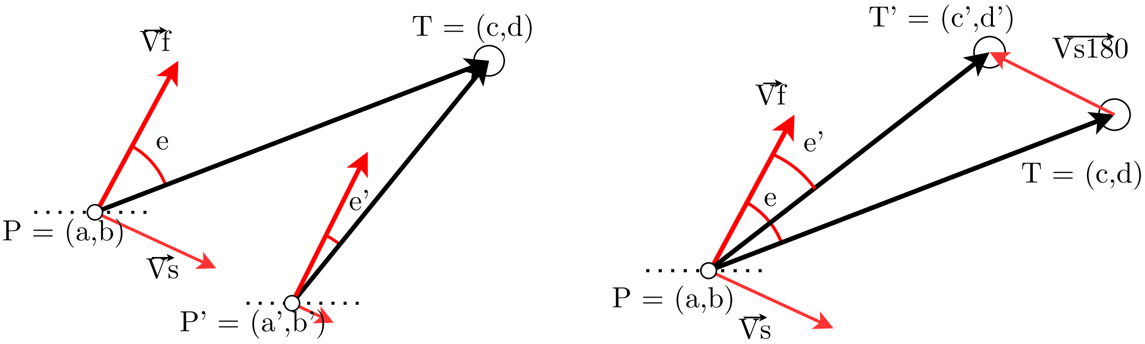
8 Appendix II - Deriving controller equations

The control law for the angular controller is as follows:

$$e = \lambda - \gamma \tag{1}$$

$$e' = e - \delta e \tag{2}$$

For the sideways controller the equations implementing them derive from an insight shown in Fig.2. In order to control the angle e using the sideways we first compute it using Eq.1. To correct this angle the controller would suggest a change in angle (Eq. 2), the same as in the angular controller, represented in $^\circ/s$. In the case of sideways velocity the challenge is converting this to units of mm/s, so the problem becomes the one exposed in Fig.2a: what is $|\overrightarrow{V_s}|$ such that e yields e' in the next step? (assuming the sideways controller is working in parallel to the angular one and independent to its effects). This can be solved by iteration over magnitudes until our desired angle is achieved, but we derived analytical solution to this problem.



(a) Diagram of update of position based on V_s

(b) Equivalent diagram to a) but in which the target moves instead of the pursuer

Figure 2: Deriving equations for the V_s controller